A Finite-State Approach to Shallow Parsing and Grammatical Functions Annotation of German

von
Frank Henrik Müller

Philosophische Dissertation
angenommen von der Neuphilologischen Fakultät
der Universität Tübingen
am 16. Februar 2005

Berlin
2007

Hauptberichterstatter:	Prof. Dr. Erhard W. Hinrichs
Mitberichterstatterin:	PD Dr. Anette Frank
Dekan:	Prof. Dr. Joachim Knape

Danksagung

Ich danke allen, ohne die diese Arbeit nicht möglich gewesen wäre. Da ist zunächst mal mein Betreuer Erhard Hinrichs, der mich mit einem komfortablen Stipendium ausgestattet hat und dann später im SFB 441 eingestellt hat. Bei meiner Arbeit hat er mir zur rechten Zeit Freiheit gelassen und zur rechten Zeit Druck gemacht. Vor allem hat er sich die Zeit genommen, Kapitel für Kapitel meiner Arbeit mit mir durchzugehen, sonst hätte sie nicht so eine gute Struktur. Und was dann noch durch die Lappen gegangen ist, hat die Zweitgutachterin Anette Frank akribisch angemerkt. Tatsächlich hätte ich mit ihren Anregungen genug Themen für eine Habilitation gehabt, aber das steht auf einem anderen Blatt. Mit dabei beim Promotionskolloquium waren dann noch Frank Richter und Irene Rapp als Mitberichterstatter und Klaus-Peter Philippi als Vorsitzender des Promotionsausschusses. Auch ihnen gebührt ein Dank für ihre Mühe.

Und dann sind da ja noch die lieben Kollegen, die mir immer beigestanden haben und manchmal sicher auch ganz schön was ertragen haben. Allen voran ist Tylman Ule zu nennen, bei dessen Hilfe ich am besten gar nicht erst anfange aufzuzählen, da es kein Ende nehmen würde. Nur die mindestens 2500 Cappuccino, die er mir gemacht hat, will ich hier nicht unerwähnt lassen. Fünf Jahre jeden Tag zehn bis zwölf Stunden in einem Büro zu sitzen, prägen sicher. Immer um Rat fragen durfte ich auch Sandra Kübler, was ich auch gerne in Anspruch genommen habe. Herzlich gedankt sei auch Heike Telljohann, wo immer sie grad ist, für die interessanten Diskussionen über die deutsche Syntax. Und nicht vergessen will ich meine beiden Kollegen Holger Wunsch und Piklu Gupta, die dann später noch dazukamen.

Aber angestoßen hat das Ganze natürlich die liebe Familie, ohne die ich ja nicht wäre, was ich bin. Wäre ich nicht in einem so anregenden Umfeld groß geworden, hätte ich sicher nicht studiert und promoviert. Und so bedanke ich mich bei meinen Eltern für die jahrelange moralische und finanzielle Unterstützung und für ihr Verständnis dafür, dass ich selten den langen Weg nach Münster nehmen konnte. Meiner Schwester Stefanie danke ich stellvertrend für alle Steuerzahler, die mein Studium finanziert haben, ohne selbst je studiert zu haben. Und nicht zuletzt danke ich auch meinen Großmüttern Gertrud Müller, Anna Hakenes und Mia Robering für ihre ideelle und finanzielle Unterstützung meines Studiums.

Und so ganz nebenbei habe ich mich im wunderschönen Tübingen auch sehr wohl gefühlt und denke gerne daran zurück. Daran haben natürlich auch die vielen Freundschaften Anteil, die ich dort geknüpft habe. Neben meinen lieben Kollegen wäre hier im Bündel insbesondere meine englische Theatergruppe, die PROVISIONAL PLAYERS, zu nennen. Danke. *Those were the best days of my life.*

Contents

1 Introduction **1**
1.1 Initial Problem 1
1.2 Parsing 2
1.3 Main Issues of this Thesis 3
1.4 Reader's Guide and Overview of the Parser 5

2 A Rule-Based Finite-State Parser for German **9**
2.1 FSAs as a Parsing Model 9
2.2 Parsing German with Hand-crafted Rules 17
2.3 Basic Principles of the Parser 21
2.3.1 Robustness 21
2.3.2 Modularity of Annotation Components 26
2.3.3 Underspecification 31
2.4 The *fsgmatch* Annotation Tool 37

3 Related Work **49**
3.1 General Remarks 49
3.2 Shallow Parsing 50
3.2.1 Finite-State Approaches and Shallow Parsing 50
3.2.2 Braun (1999); Neumann, Braun, and Piskorski (2000); Neumann and Piskorski (2002) 52
3.2.3 Wauschkuhn (1996) 53
3.2.4 Schiehlen (2002, 2003a,b) 54
3.2.5 Kermes and Evert (2002, 2003) 55
3.2.6 Trushkina (2004) 56
3.2.7 Schmid and Schulte im Walde (2000) 57
3.2.8 Becker and Frank (2002) 59
3.2.9 Klatt (2004a,b) 60

3.2.10 Brants (1999) 61
3.2.11 Summary 62
3.3 Grammatical Functions Annotation 62
3.3.1 Finite-State Approaches and GF Annotation 62
3.3.2 Schiehlen (2003a,b) 63
3.3.3 Trushkina (2004) 64
3.3.4 Schmid and Schulte im Walde (2000) 65
3.3.5 Kübler (2004a,b) 66
3.3.6 Summary 67

I Annotating Shallow Structures 71

4 Shallow Annotation Structures 73
4.1 The Nature and Right-of-Existence of Shallow Parsing 73
4.2 Definition of Shallow Structures 93
4.2.1 Chunks 93
4.2.2 Topological Fields 108
4.2.3 Clauses 117

5 Annotating Shallow Structures with FSAs 123
5.1 Outline of the Task Description 123
5.2 Easy-First-Parsing and a Divide-and-Conquer Strategy 125
5.3 Correcting Tags by Using Topological Field Information 133
5.4 Handling Recursion by Iterating FSAs 140
5.5 Annotating Chunks Top-down 147

6 Annotating Complex Chunks 159
6.1 Task Description and Motivation 159
6.2 Appositions and Named Entities 161
6.3 Annotation of Appositional Constructions 163
6.4 Annotation of Named Entities 168
6.5 Annotation of Other Complex Chunks 169

II Annotating Grammatical Functions 173

7 Annotating Deeper Structures – Grammatical Functions 175
7.1 Outline of the Task Description 175

7.2 Definition of Grammatical Functions 179
7.3 Detailed Task Description and Problems 185
7.4 Motivation for Annotation 189
7.5 Methods of Annotation – An Outlook 191

8 Reducing Morphological Ambiguity 195
8.1 Task Description 195
8.2 Reducing Ambiguity on Chunk Level 199
8.3 Reducing Morphology on Clause Level 203
8.4 Further Enriching the Corpus with Morphological Information 205
8.5 Conclusion, Outlook and Comparison with other Approaches . 207

9 Annotating Grammatical Functions (GFs) 211
9.1 Task Description 211
9.2 Converting Lexical Entries into FSTs 222
9.3 Disambiguation by Ranking Rules 230
9.4 A Longest-Match Strategy for the Disambiguation of SC Frames231
9.5 Using Markedness to Disambiguate GF Order 232
9.6 Extending Disambiguation to non-Lexical Patterns 238
9.7 Modeling Ranking by Using OT Constraints 241

III Evaluation and Comparison 257

10 Shallow Annotation 259
10.1 The Evaluation System 259
10.1.1 Task Description 259
10.1.2 The TüBa-D/Z Gold Corpus 261
10.1.3 Conversion into common representation 262
10.1.4 Methods of Evaluation 265
10.2 The Test-Setting for Evaluation 267
10.3 Results 270
10.3.1 Chunks 270
10.3.2 Topological Fields 278
10.3.3 Clauses 290

11 Grammatical Function Annotation 299
11.1 The Evaluation System . . . 299
11.1.1 Task Description . . . 299
11.1.2 GFs in the TüBa-D/Z Gold Corpus . . . 300
11.1.3 Conversion into Common Representation . . . 302
11.1.4 Methods of Evaluation . . . 304
11.2 The Test-Setting for Evaluation . . . 308
11.3 Discussion of Overall Evaluation Results . . . 310
11.4 Impact of Different Components on Annotation . . . 338
11.4.1 Evaluating Morphological Ambiguity Reduction . . . 338
11.4.2 Impact of Morphological Ambiguity Class Reduction . 344
11.4.3 Impact of non-lexical Component . . . 348
11.4.4 Impact of Support Verb Component . . . 355

12 Comparison with Related Work 359
12.1 General Remarks . . . 359
12.2 Shallow Parsing . . . 360
12.3 Grammatical Functions Annotation . . . 370

13 Conclusion 379

14 Bibliography 383

A Abbreviations 395

B The Stuttgart-Tübingen Tagset (STTS) 397

C The TüBa-D/Z Inventory of Syntactic Categories and Grammatical Functions 401

D Lists of Verbs without SC Frame 405
D.1 112 Verb Types without SC Frame . . . 405
D.2 44 Verb Types without Base Form . . . 408

Chapter 1

Introduction

1.1 Initial Problem

During the last two decades, there has been a paradigm shift in linguistics and related sciences. Before this shift, linguists mainly relied on introspective language data for their grammaticality judgments and sometimes supported those judgments with examples taken subjectively from real language data. Since this shift, an increasing number of linguists have taken real language data into consideration which are taken from large electronic corpora of a language. This paradigm shift has not only influenced linguistics in the narrow sense but also areas such lexicography or language teaching methodology.

If appropriate search tools are provided, corpora are already a useful data source even if they do not contain any linguistic annotation, i.e. if they simply contain the words of the language without any further information. However, furnishing a corpus with explicit linguistic information (i.e. linguistic annotation) considerably increases the usefulness of a corpus. In some cases, information as simple as the part of speech suffices for a linguistic study. If, for example, a linguist wants to investigate the conjunction 'da', there is the problem that the conjunction 'da' is a homograph since it may be either a conjunction or an adverb as illustrated in sentence 1. If, however, the part of speech (PoS) is tagged (i.e. annotated) the homography is resolved and the conjunction 'da' can be distinguished from the adverb 'da' by its PoS tag. Still, corpora which are annotated with PoS tags are not fully satisfactory because they simply show the linear order of tokens without any constituent structure or grammatical relations. Many linguistic phenomena cannot be captured just using PoS tags.

(1)

Er	drehte	sich	um,	aber	da	niemand	da	war,	ging	er	weiter.
He	turned	himself	around,	but	since	nobody	there	was,	went	he	on.

'He turned around but since nobody was there he moved on.'

In order to make a corpus suitable for the testing of linguistic theories, it is necessary to annotate it with more specific linguistic phenomena. This annotation may be done manually. However, this is a tedious task, consuming both time and money. For this reason, very large corpora virtually cannot be annotated manually at all. Consequently, automatic syntactic parsing, which is the automatic annotation of syntactic phenomena, is crucial for the annotation of very large corpora. We present an approach to automatic annotation in the thesis at hand.

1.2 Parsing

A parser is a device which decides whether its input adheres to the grammar of a certain language. In order to do so, it checks the correctness of the syntactic structure of the input with respect to the grammar of the language. It may then either simply accept or reject the input, or it may assign the respective syntactic structure to the input, in the case of acceptance. The syntactic structure of the input is the structure of the input with respect to the grammar. For natural languages, this syntactic structure does not necessarily solely subsume the syntactic structure in the linguistic sense because it may also subsume semantic information. In the thesis at hand, however, we mean syntactic parsing in the linguistic sense if we speak of parsing (unless explicitly stated otherwise).

The parser can be viewed as the implementation of an abstract automaton which adheres to a grammar. The parser as such is independent of the grammar, i.e. it can be fed with any grammar. However, the parser is not independent of the expressive power of its grammar, or, viewed from a different perspective, the grammar is limited to the expressive power of the abstract automaton which is the theoretical basis of the parser. Hence, a parser which is fed with a regular grammar can be seen as equivalent to a finite-state automaton. The parser cannot annotate any structures which cannot be described by a regular grammar. In the Chomsky Hierarchy (Chomsky, 1956), the regular grammar is the most restricted and, thus, least powerful type of grammar (type 3 grammar). Further types of grammars are context-free grammars (type 2 grammars), context-sensitive grammars (type 1 grammars) and unrestricted rewrite grammars (type 0 grammars). We have decided to use a regular grammar for our annotation task because its restrictions make it very effective and, thus, capable of annotating large quantities of data in an acceptable time. In this thesis we will investigate the annotation of shallow syntactic structures and grammatical functions with finite-state automata.

1.3 Main Issues of this Thesis

Syntactic parsing can roughly be classified within two paradigms. On the one hand, there are those approaches which create grammars manually and, on the other hand, there are those approaches which automatically learn grammars from manually annotated corpora. In the one case, the linguistic information is integrated into the parser via the explicit encoding by the linguist and, in the other case, the information encoded by the linguist in the corpus comes into the grammar indirectly via automatic extraction. In both cases, the grammar is to some extent domain and text type specific since, in the one case, the grammar is tuned to a certain corpus while it is trained on a certain corpus, in the other case. In both cases, a mechanism has to be found to deal with ambiguous structures because one and the same structure may be assigned more than one annotation. This can be exemplified by sentence 2. Both NPs in the sentence can be either the subject or the object. If the structure is not supposed to be left ambiguous, in such cases, a decision has to be taken. Many learning approaches take this decision on the grounds of probabilities. A mechanism to cope with these cases with a manually constructed grammar is one central issue of the thesis at hand.

(2)	Ihre	Mutter	hat	sie	lange	nicht	gesehen.
	Her	mother	has	she/her	long	not	seen.

‘She has not seen her mother for a long time.
Her mother has not seen her for a long time.’

An important reason for the decision in favour of the manually constructed grammar is to show an alternative to the existing learning approaches which crucially depend on the existence of manually annotated corpora. Since manually annotating a corpus is a time-consuming task and since parsers learned from corpora typically only work well on the domain and text type they were trained on, it is worthwhile to offer an alternative to this procedure. For a person who wants to annotate a corpus, it would then be a question of whether to adapt an existing grammar to the new domain and/or text type or to annotate a corpus of the new domain/text type. This decision might be guided by both theoretical and practical motives.

Non-recursive kernel phrases, so-called *chunks* (cf. Abney, 1991, 1996c), have already been successfully annotated using finite-state automata (FSAs). It is one of the central issues of this thesis to scrutinize whether it is possible to extend this shallow annotation to topological fields and clauses which we posit to adhere to the same syntactic restrictions as chunks in German. In such a way, a very effective annotation approach could be used to build the structure which we deem crucial for the further annotation of grammatical functions (GFs).

After the annotation of chunks, topological fields and clauses with FSAs, these structures can be used as the basis for the annotation of GFs. Chunks can serve as the targets of GF annotation, clauses limit the scope of the attachment of GFs, and the preferences in topological fields can be used for disambiguation. Chunks, topological fields and clauses adhere to syntactic restrictions and can be captured by just using PoS tag information. By contrast, GFs are subject to lexical selection and their realisation is restricted to certain morphological features. Sentence 3 gives an example. The fact that the sentence contains a subject, a direct (accusative) and an indirect (dative) object is caused by the lexical selection of the verb 'geben'. Not all verbs may select those GFs. The GFs are marked by their case and not by their position since the order of constituents is not fixed in German. It will be one central issue of this thesis to investigate whether it is possible to integrate both the lexical selection information and the morphological features into a finite-state approach and, thus, apply the advantages of the finite-state approach to the annotation of GFs.

(3)	Klaus	gibt	den	Brief	seiner	Kollegin.
	Klaus	gives	the	letter	his	workmate.

'Klaus gives the letter to his workmate'

1.4 Reader's Guide and Overview of the Parser

The following section is intended to give the reader an overview of the parser described in the thesis at hand. At the same time, it can be used as a reading guide since we will refer to all relevant chapters in the thesis when introducing the respective part of the parser. The reader may then have a look at the abstracts at the beginning of most chapters. They give an overview of the content of each chapter and refer to the relevant sections for details.

Our parser is a rule-based parser for German which uses finite-state automata (FSAs) as a parsing formalism. All rules are created by hand. For the annotation of grammatical functions (GFs), the fact that certain patterns re-occur is used to semi-automatically yield rules. Nevertheless, the patterns themselves are created by hand. The shallow parser has a rather small grammar consisting of 288 rules. 103 of these are for verb chunks since we place a great deal of emphasis on this structure. 80 rules are for all the other chunks. 78 rules are for topological fields and 27 rules are for clauses. Since the annotation of GFs needs subcategorization information and the order of GFs in German is relatively free, the GF parser has many more rules: Overall, there are 9,017 rules but 31.2% of them are for support verbs (cf. section 9.2). We also have a backup component which does not use any sub-categorization information. This component has just 41 rules and yields very good results (cf. section 11.4.3).

Chapter 2 introduces FSAs as a parsing model and discusses some general issues related to rule-based parsers. It then gives an overview of some general principles applied in the construction of the parser, and it introduces the tool used for the implementation of the parser. Chapter 3 discusses related work. We discuss work which is related because the same or similar structures are annotated, i.e. shallow structures or GFs. In some cases, the same or a similar approach is applied, i.e. FSAs; in some cases, however, approaches which are completely different are applied.

Our parser roughly works in two steps: The first step is the annotation of shallow structures; the second step is the annotation of GFs. The first step is outlined in part I, and the second step is outlined in part II. Part III, which is the last of the three parts of the thesis at hand, deals with the evaluation of the two steps of the parser

The input to the shallow parser is text which is segmented into sentences and tokens. The tokens are part-of-speech tagged with tags corresponding to the STTS part-of-speech (PoS) tagset (cf. Schiller, Teufel, and Thielen, 1995, and appendix B). Tokenization and sentence segmentation are achieved by tools developed for the KaRoPars parser (cf. Ule and Müller, 2004) but not by the author of this thesis. They are not described here. Shallow structures are then annotated solely on the basis of tokens annotated with PoS tags. Chapter 4 first defines those shallow structures by means of general features they share and, then, gives the details of the constituents which we have defined as shallow structures, i.e. chunks, topological fields and clauses.

Chapter 5 is the core chapter describing the method of annotating shallow structures with FSAs. It describes the cascaded FSAs as well as the *Divide-and-Conquer strategy* and the *Easy-First-Parsing strategy* which are used. Furthermore, it describes how we use the topological field information to correct some notorious errors in PoS tags during parsing. The problem of handling recursion with FSAs is also treated in this chapter. A last point is the presentation of our top-down-chunking approach which is in contrast to most other approaches which apply a bottom-up approach. Chapter 6 explains why the typical concept of a chunk is not enough for the annotation of GFs. It then shows how we extend our concept of chunks to cover complex chunks. These are annotated in a different way than typical chunks because lexical information is used to a certain extent.

The shallow annotation structure, i.e. chunks, topological fields and clauses, is the ideal basis for the annotation of GFs, which is described in part II. Apart from the shallow annotation structure, the annotation of GFs relies on morphological information, subcategorization information and on linear ordering principles. Chapter 7 defines GFs and gives the motivation for their annotation. It also presents a detailed task description. Morphological information is crucial for the annotation of GFs since most GFs are associated with case. Chapter 8 describes how we annotate morphological ambiguity classes using the tool DMOR (Schiller, 1995). A complete disambiguation of morphology is not possible on the basis of local information but we show in chapter 8 how we use the shallow structures which we have annotated to considerably reduce morphological ambiguity.

Chapter 9 is the core chapter describing GF annotation on the basis of shallow structures and morphological information and with the help of subcategorization information taken from the lexicon IMSLex (Eckle-Kohler, 1999). We describe how we integrate the subcategorization information from the lexicon into the FSAs. In total, we extract 105 different subcategorization frames for verbs and 7 for adjectives. 12,688 verbs receive on average 2.6 different subcategorization frames and 2,224 adjectives are assigned on average 1.5 different frames. Furthermore, we describe the mechanism which allows us to choose between patterns if more than one could be applied, i.e. the ranking of FSAs; and we explain the principle according to which the FSAs are ranked, i.e. the degree of markedness of a certain pattern. We also show how we extend our approach to cover patterns for which there is no subcategorization information in the lexicon.

Part III of the thesis is about the evaluation of the parser. We devoted such a large section of the thesis to the evaluation of the parser because the quality of a tool is important for prospective users. But there are also other reasons: We measured the impact of different factors like PoS tagging, morphological annotation and subcategorization information on the quality of annotation. These findings can help other researchers when constructing parsers because they show in a general way the contributions of certain information to the performance of the annotation generated by a parser. What is more, is that these findings also have a meaning for the nature of the German language. The contribution of morphological and subcategorization information to the annotation performance can be interpreted in terms of the nature of GFs in German and how decisive linear order, morphology or subcategorization information are for their definition.

Part III is divided into three chapters: Chapter 10 presents the evaluation of shallow annotation and chapter 11 that of GFs. Chapter 12 compares the results of our parser with those of other approaches. The chapters concerning evaluation of our parser both have a section in which the system of evaluation is described in detail. This includes a description of the TüBa-D/Z which we use as a gold standard and a description of how we convert the TüBa-D/Z such that it can be compared to the output of our parser. Furthermore, the two chapters contain a section which describes the exact test setting. The results of the parser are then discussed with respect to the factors discussed in the paragraph above.

Chapter 2

A Rule-Based Finite-State Parser for German

ABSTRACT: Chapter 2 gives an introduction to the nature and architecture of our parser. Section 2.1 presents finite-state automata (FSAs) as a parsing model and illustrates the motivation for using this very model for the parser at hand. Section 2.2 explains the motivation for and the main features of the parsing approach used in the dissertation at hand, namely the annotation of constituent structures and grammatical functions with hand-crafted rules. Section 2.3 introduces and illustrates principles spanning the annotation of all linguistic phenomena (i.e. robustness, modularity and underspecification) and, thus, all components of the parser. Section 2.4 shows how the FSAs are implemented with the help of the tool *fsgmatch* by giving illustrative examples of typical linguistic phenomena.

2.1 FSAs as a Parsing Model

A parser is the implementation of an abstract automaton. In the thesis at hand, we use finite-state automata (FSAs) as the abstract automaton which serves as the basis of our parser because FSAs are very effective formalisms for parsing. The reason for this is that FSAs can be described by regular grammars, which are the most restricted and, thus, least powerful type of grammar in the Chomsky Hierarchy (Chomsky, 1956). Since the power of the formalism is restricted, for example it cannot deal with recursive structures, it is considerably faster than other approaches as Abney (1996c) has shown.

In this thesis we will show that, although the power of FSAs is restricted, it is still possible to use them for the annotation of chunks, topological fields, clauses and grammatical functions (GFs), by building a cascade of FSAs in which succeeding automata can build on structures generated by preceding ones (cf. Abney, 1996b,c).

Formally, an FSA is a 5-tuple defined by:

- $\mathcal{Q}$: a finite set of N states (q_0, q_1, q_2, ..., q_N)
- Σ: a finite set of input symbols, i.e. alphabet
- $q_0 \epsilon \mathcal{Q}$: the initial state
- $F \subseteq \mathcal{Q}$: a set of final states
- δ (q,i): a transition function between states

An FSA can only recognize regular languages. The FSA can either be seen as an acceptor or a transducer. In this first case, it simply decides whether the input string adheres to the definitions given for the set of input strings defined by the regular expression grammar of the parser, i.e. the regular language. In the other case, the FSA acts as a finite-state transducer (FST). This means that it not only accepts a certain string but also generates an output. Formally, in this case, the FSA has to be extended by the output alphabet γ; and the transition function has to be defined as a transition function between the input alphabet and the output alphabet.

For the FSTs of the shallow parser, the PoS tags of the standard German tagset the Stuttgart-Tübingen Tagset (STTS) (cf. Schiller, Teufel, and Thielen, 1995, and appendix B) are part of the input alphabet. Since the shallow parser is a cascade of FSTs, in which following FSTs build on structures generated by preceding ones, the structures generated by preceding FSTs are also part of the input alphabet of the following FSTs. The initial state of the FST is the state in which it has not yet received any input. The transition function defines the sequence in which the input elements are recognized. The finite set of states keeps track of the position the FST is in, i.e. it keeps track of the elements already consumed. If the FST is in a final state, it recognizes the sequence of input elements consumed up to that point as part of the language described by the FSA unless there are more elements to be

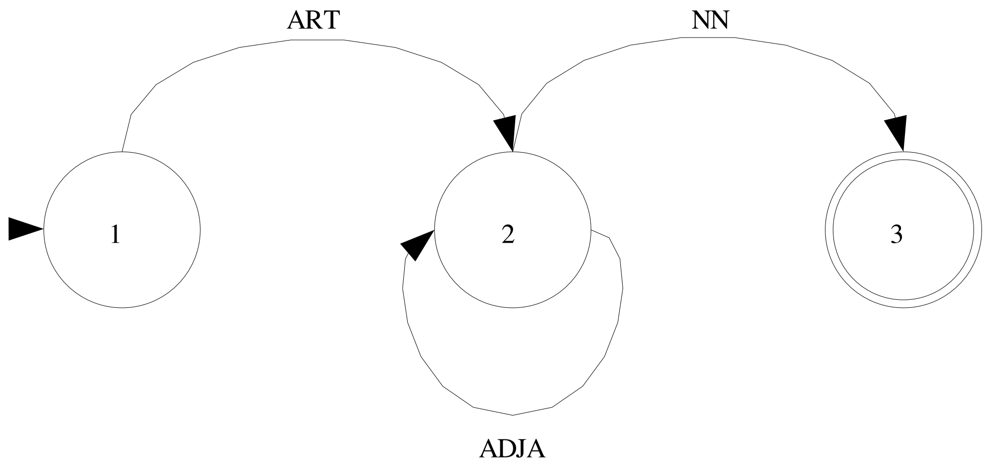


Figure 2.1: FSA which accepts one type of noun chunk (NC)

consumed, i.e. unless one or more further transition from that final state to another one (or itself) is possible.

Figure 2.1 shows an FSA which accepts a string of PoS tags. The example shows a typical idealized example of one type of noun chunk (NC). The initial state (1) begins the FSA. The FSA is then ready to consume an article (ART) and traverse into the state 2. The nomenclature for the states is arbitrary. The states could just as well have any other distinct names. The main characteristic of the states is that they show which following input is allowed at the position of the input string at which the FSA is at the moment. In the case of state 2, these are either the transition consuming an attributive adjective (ADJA) or the transition consuming a noun (NN). Since the transition consuming the adjective leads to the same state it starts from, it can be applied zero or more times. The transition consuming the noun can only be traversed once. After this, state 3 is reached, which is the final state. Since, in this case, there are no further transitions possible, the FSA accepts the sequence at that point as part of the language defined by the FSA. In the case of an FST, the FST would simply emit the intended output string (i.e. NC).

The FSA in figure 2.1 shows a deterministic FSA. In a deterministic FSA, the actions of the automaton are always completely determined with respect to the input string and the initial state, i.e. given an input symbol, there is one and only one transition possible for each state. If this is not the case, the automaton is a non-deterministic FSA. Partee, ter Meulen, and Wall (1993) list four potential cases in which an FSA is no longer deterministic:[1]

- given a state-input element pair, there may be more than one next state
- given a state-input element pair, there may be no next state
- the FSA can read a string of input elements in one move
- the FSA can change state without consuming an input element

Daciuk (1998) gives a further case of non-deterministic FSAs: If an FSA has more than one initial state, it is also not completely determined and, thus, a non-deterministic FSA. Additionally, we point out a further problem with respect to deterministic FSAs: Typically, the task of an FSA is described as recognizing whether an input string is part of the language of the FSA or not. The FSA reads in as many elements as there are in the string, and, if it is in a final state when the string is fully consumed, it accepts the string. However, the task in most parsing tasks is to assign structures to sub-strings, i.e. a whole sentence is read in and the parser has to recognize sub-strings of it as constituents. In this case, it has to be determined at which point the FSA should stop in cases in which the parser has more than one final state.

For the parser described in this thesis, we use deterministic FSTs. This is crucial because only deterministic FSTs guarantee the full efficiency of the finite-state approach. If an FST is not deterministic, it may produce more than one output, i.e. the output of the parser is not predictable. This is, of course, not acceptable for a parser. Thus, in order to guarantee a definite output, a non-deterministic FST would have to be extended by mechanisms which do not comply with the finite-state approach and would, thus, not be as efficient as this approach. We will show in the following how three of the six potential violations of determinism can be dealt with, namely the problem that there may be more than one next state for a given input, the problem that the FSA can change the state without consuming an input element and

[1]Paraphrased from Partee, ter Meulen, and Wall (1993), sec. 17.1.3, who use a different notation and a different nomenclature.

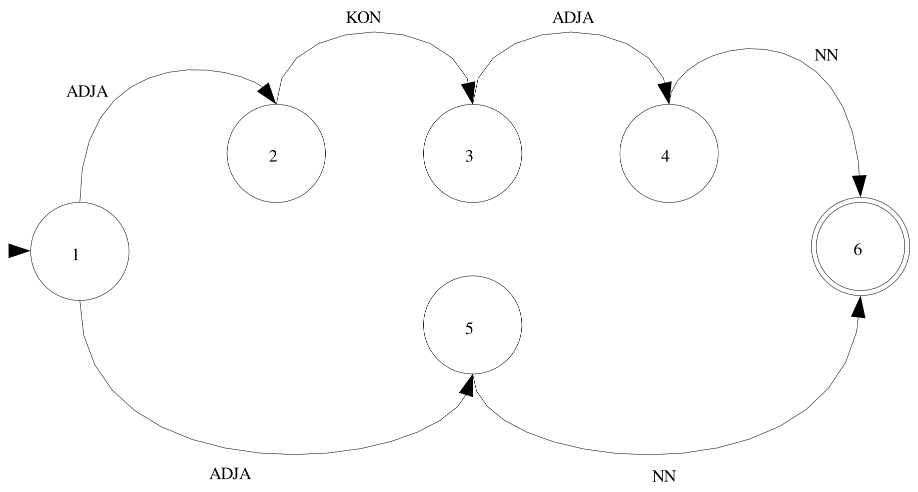


Figure 2.2: Non-deterministic FSA

the problem of more than one final state. The other three potential violations are problems which are not applicable in the context of our parser.

First of all, it is important to notice that every non-deterministic FSA can be converted into a deterministic FSA. This means that for every non-deterministic FSA there exists an equivalent deterministic FSA (cf. Partee, ter Meulen, and Wall, 1993, sec. 17.1.5). We can use the first of the three problems as a example to illustrate how non-deterministic FSAs can be transformed into deterministic ones. Consider figure 2.2: We want to build an FSA which recognizes both an NC with one adjective and an NC with two coordinated adjectives. Since this is an either/or-decision, we could make the decision between the first and the second case immediately after the intial state. However, in this case, the FSA would be non-deterministic since there are two transitions ADJA from the initial state 1 to the states 2 and 5 (with the problems mentioned above). Thus, in order to achieve a deterministic FSA, we have to transform our FSA as shown in figure 2.3.

In the case of the transformation of a non-deterministic FSA into a deterministic one in figure 2.2 and figure 2.3, the result also appears intuitively comprehensible since the number of states has fallen from six to five. However, if a non-deterministic FSA has k states, the number of states for the deterministic one might be 2^k. Still, our example shows that this does not have to be the case. But, since there is no alternative if we want to stay

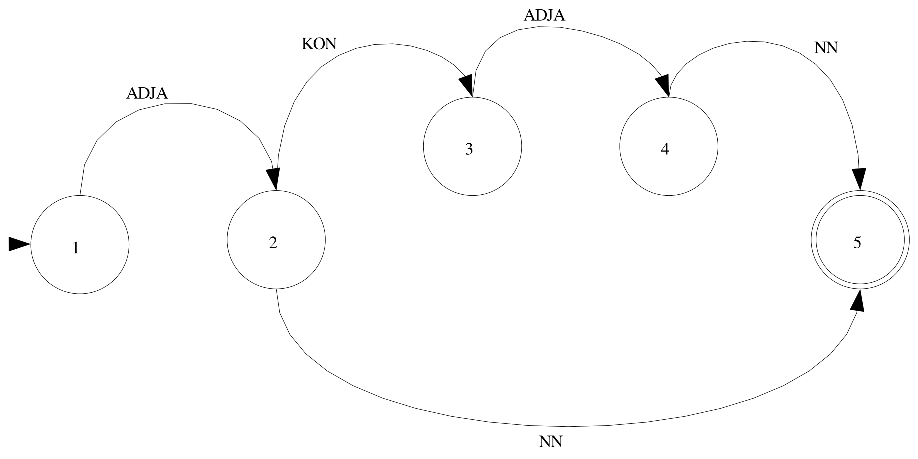


Figure 2.3: Deterministic FSA equivalent to the FSA in 2.2

within the finite-state framework, we have to accept this possibility.

The second and the third problem with deterministic FSAs is connected with the selection of the sub-string to be processed from the stream of data. Consider the example in figure 2.4 which exemplifies the second of our problems (i.e. state change without input consumption): The FSA accepts strings representing an NC with an article, an adjective and a noun; and it accepts strings representing an NC without an article but with an adjective and a noun. To put it simply, the article in the NC is optional, thus allowing NCs with and without an article. Consequently, both the NC 'saure Milch' and the NC 'die saure Milch' are accepted. Since the concept of optionality is crucial for parsing, this is a concept which we would like to be able to model with deterministic FSAs. The optional consumption of an element is represented by the ϵ.

In an ϵ transition, the FSA changes from one state into another without consuming any input. This is a way of expressing optionality since the FSA would now accept the string 'ART ADJA NN' and the string 'ADJA NN'. However, FSAs with ϵ transitions are not deterministic. This becomes obvious if one considers what happens at the point at which the FSA receives an ART as an input. It may either consume the input and proceed to state 2 or it may move to state 2 without consuming the input. In the latter case, this would mean that the FSA would not accept the string of elements 'ART ADJA NN' as an NC. If we, furthermore, accept that the input tape is moved

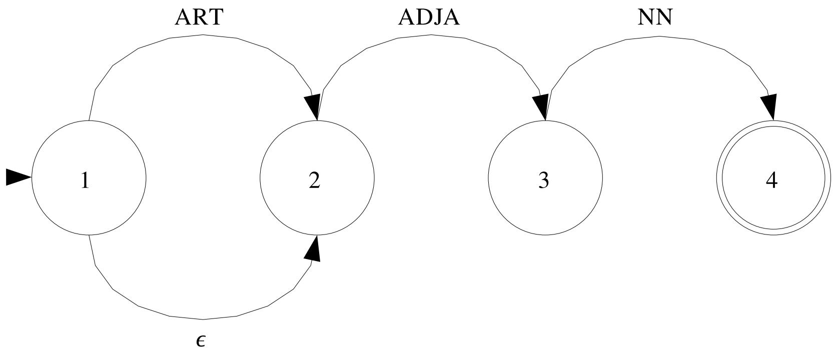


Figure 2.4: FSA with ϵ transition to model optionality

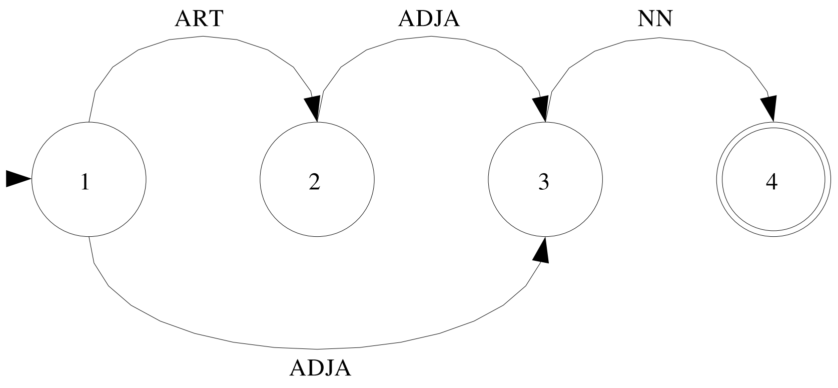


Figure 2.5: Deterministic FSA modeling optionality

by one element if this element is rejected by the FSA (as the ART in our example), then the FSA in figure 2.4 might either accept the input string ‘ART ADJA NN’ as an NC or the string ‘ADJA NN’, which is a sub-string of it.

Thus, the behaviour of the FSA in figure 2.4 is undetermined unless there is some kind of condition like ‘only do ϵ transition if no element can be consumed’. However, such a condition would be outside the finite-state approach. Since any non-deterministic FSA can be transformed into a deterministic one, it is also possible to express optionality with a deterministic FSA. This is illustrated in figure 2.5. If the FSA in figure 2.5 receives the string ‘ART ADJA NN’ as its input, behaviour is completely determined

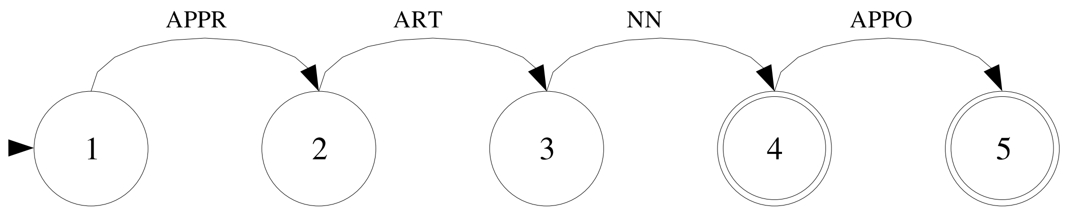


Figure 2.6: FSA with two final states applies *longest match*

since the FSA will change from state 1 to state 2 and finally accept the input. Since the FSA can also change from state 1 to state 3 consuming an ADJA, the string 'ADJA NN' is also accepted. What is very important to point out is that there is no longer a conflict whether to match the longer or the shorter string if the longer string occurs. With the FSA in figure 2.5, the longer of the two strings is matched automatically. If the sequence 'ART ADJA NN' occurs, the element ART will always be consumed at the point at which the sub-string 'ADJA NN' occurs.

The third problem with respect to deterministic FSAs, i.e. the problem that there may be more than one final state, is also connected with the problem of cases in which either a string or a sub-string thereof can be matched. Figure 2.6 shows an FSA which recognizes an idealized prepositional chunk (PC) which either just has a preposition (APPR) or a circumposition consisting of a left part (APPR) and a right part (APPO). Consequently, the APPO is optional just like the article was optional in the previous example. In the FSA this is modelled such that the FSA can be in a final state either in the case that the head noun NN is consumed or in the case that a following (if any) right part of a circumposition occurs.

Still, in cases in which an APPO occurs, the behaviour of the FSA is not determined. It may either stop because it is in a final state (state 4), or it may consume the APPO and move into the second final state (state 5). However, the FSA can be made deterministic by applying the *longest-match* strategy proposed in Abney (1996c). This means that, in cases in which the FSA can reach more than one final state, it proceeds to that final state which consumes most elements. Abney (1996c) argues that this strategy is also in line with the English language and its application is, thus, well motivated. Since we use delimiters in most rules, i.e. we define clear borders of constituents, which cannot be transgressed, the concept of *longest* *match* is not as crucial in our approach as in the one of Abney (1996c). Still, in cases like the one in the example above, it provides a means to enforce deterministic and correct annotation.

2.2 Parsing German with Hand-crafted Rules

We have motivated the use of FSAs for our parsing approach. There remains the issue of why we are using hand-crafted rules for those FSAs since some approaches use grammars automatically acquired from corpora. This seems to be a more straightforward solution, at first sight, because, then, grammars can be learned quickly automatically from language corpora. The case of the automatic annotation of word categories (i.e. PoS tagging) can serve as an example to illustrate the paradigm shift from approaches working with hand-crafted rules to approaches extracting information from linguistically annotated corpora. PoS information was one of the first phenomena to be annotated in linguistic corpora. When PoS tagging was first carried out in the 1960s on the Brown corpus (Kucera and Francis, 1967), it was dominated by approaches using hand-crafted rules in combination with linguistic data sources like Klein and Simmons (1963) and Greene and Rubin (1971).

Later approaches started using statistical information. Marshall (1983) is the first to use a matrix of collocational probabilities which indicates the relative likelihood of co-occurrence of all ordered pairs of tags. However, this approach still required very large computing time and space and, thus, the natural language processing (NLP) community was not completely convinced of the advantages of stochastic approaches. This changed with works like that of DeRose (1988). DeRose (1988) describes an algorithm which is able to assign PoS tags on the basis of a transitional probability matrix – but in linear time. Works such as that of DeRose (1988) have led to a paradigm shift from approaches using hand-crafted rules to stochastic approaches or, more generally, to machine learning approaches. Machine learning approaches offer a very ‘elegant’ solution to the problem of automatically linguistically annotating a corpus because they do not require any heuristics (or at least not as many as approaches working with hand-crafted rules) and can be defined formally as an algorithm.

DeRose (1988) explicitly mentions this fact as one of the advantages of his approach when comparing it to previous approaches: ‘Those features that can be algorithmically defined have been used to the fullest extent. Other add-ons have been minimized.' (p. 35) It is especially what DeRose calls 'add-ons' which is the focus of criticism of most people proposing stochastic approaches. The main reason for this is that these 'add-ons' are typically not reproducible by other scientists and the approach, thus, remains quite opaque to everyone except the developer. For this reason, a hand-crafted grammar would typically be lost after the developer had left the institution. Furthermore, it is said that one big disadvantage of these approaches using hand-crafted rules is that they are tuned to the test set, i.e. if they have to be applied to other types of corpora than the one they have been trained on, then the grammar would have to be reconstructed. This reconstruction could only be carried out by the respective developer since only he or she is capable of understanding the system.

Hence, from the end of the 1980s onwards, almost exclusively machine learning approaches have been used for PoS tagging. Karlsson (1990) is one of the few approaches to use a hand-crafted grammar. The approach is far from being a collection of heuristics. In fact, it uses a constraint-based approach which is formulated concisely. Still, approaches like the one of Karlsson (1990) never became as popular as machine learning approaches. One of the main reasons is that, in science, if there are two solutions to one and the same problem, then the simpler one is preferred to the more complicated one. For the problem of PoS tagging, the machine learning solution does, in fact, appear to be the simpler one. The general principle of the relevant machine learning approach can be phrased in just one scientific paper. The same is true of the general principle of, for example, an approach building on a hand-crafted grammar like that one of Karlsson (1990). However, the grammar as such is far more complex and would have to be laid out in much more detail.

But even if there were a detailed description of the grammar (or even the full explicit grammar itself), the rules and, even more, their interaction would still appear not to be transparent. Proponents of machine learning methods also point to the fact that hand-crafted grammars are tuned to the test set, i.e. the developer of the grammar just considers those phenomena which occur in the test set. If the grammar is then to be applied to another corpus of another text type and domain, then it would be laborious to adapt it to that new target language. The grammar would then have to be expensively reconstructed whereas, for a machine learning approach, the same algorithm would just have to be trained on a PoS-tagged corpus of the same text type and domain such that it could deal with the new target sub-language. Thus, not only the construction but also the adaptation of the PoS tagger to new tasks appears to be much simpler.

There are, however, some shortcomings in the points of view presented above. These are closely connected to the fact that the efforts and problems in arriving at an annotated corpus are typically not taken into account if the advantages and disadvantages of the two approaches are discussed. Very often, the annotated corpus is treated as a unlimited natural resource which simply exists. Since all successful machine learning approaches known today need a corpus with linguistic annotation as a data source for learning grammar rules, a machine learning approach cannot be applied to a text type for which there does not exist an annotated corpus. However, the construction of linguistically annotated corpora is a time-consuming and costly task. Thus, the adaptation of a machine learning approach to a new text type and domain is by no means simple; and the same was true of the development of machine learning approaches in the first place because corpora first had to be annotated before machine learning approaches could be applied.

However, the costs and problems of annotating a corpus are typically not considered in the calculation of the advantages and disadvantages of machine learning approaches; but while for approaches which work with hand-crafted rules, the linguistic knowledge is coded in the grammer, for machine learning approaches, this knowledge is coded in the annotated corpus. Thus, it is justifiable if one includes the efforts in annotating a corpus in the comparison of the costs for the two approaches. If this is done, one can realize that many of the problems of approaches using hand-crafted rules also apply to machine learning approaches. And there is in fact a lot of effort to be made if one wants to annotate a corpus with linguistic information.

First of all, manual annotators have to be trained such that they are capable of consistently annotating a corpus with linguistic information. To a certain extent, it is possible to enhance this process by giving the annotators a list of automatically generated choices. However, this approach also poses problems because humans tend to accept suggested solutions. For the process of manual annotation, one needs at least two annotators because the manually assigned annotation has to be cross-validated and, thus, has to be checked by at least one other annotator. This is done to guarantee high annotation quality in order to avoid that, e.g., one human annotator consistently misannotates certain phenomena or that annotation is inconsistent. Only if the annotators agree to a high degree in their judgements (*inter-annotator agreement*), high annotation consistency is guaranteed. If the annotation is inconsistent, the corpus is not an adequate data source for a machine learning approach to automatic syntactic parsing.

For the manual annotation of corpora, there are annotation guidelines which lay out in which way the linguistic phenomena should be encoded in the corpus. However, these guidelines also leave room for interpretation. Thus, even if there are guidelines, manual annotators spend a lot of time discussing cases of doubt or in discussing cases which did not occur in the text types of the corpus for which the annotation manual was written. If the annotators of the new corpus are not at the same time the annotators of the original corpus, it is highly unlikely that the two corpora are completely consistent. Thus, the fact that a hand-crafted grammar has to be adapted to a new text type (even by a person other than the developer) has to be seen in the context that the same is true of the linguistic annotation of corpora which serve as the basis for machine learning approaches.

Another argument against hand-crafted grammars is that the interaction of the rules cannot be controlled and remains to a large extent opaque. However, as we have shown above, the annotation of linguistic phenomena in a corpus is also to a large extent only accessible to those who have annotated the corpus because the annotation guidelines alone do not allow a full interpretation of the linguistic annotation of the corpus in all cases. Furthermore, the grammar itself which was generated by the machine learning approach is not transparent because the rules coded in it are not linguistic rules but typically collocational probabilities. Furthermore, the output of a parser using a machine learning approach does just allow vague inferences with respect as to which collocational probabilities actually triggered the respective annotation structure.

We have shown above that both approaches, those working with hand-crafted rules and those working with machine learning algorithms, have their advantages and disadvantages. The decision on which one to choose depends on the availability of a corpus or on the willingness to construct a hand-crafted grammar. At the present time, the main focus in the computational linguistics community is on machine learning approaches. Our motivation to use an approach working with hand-crafted rules is that we want to show an alternative to machine learning approaches. We thus present a method for automatically annotating a corpus without the prior manual annotation of a corpus, which is an arduous task.

2.3 Basic Principles of the Parser

2.3.1 Robustness

Typically, with respect to parsing, *robustness* means that the parser outputs partial structures in those cases in which it fails to assign a full syntactic analysis of a sentence. But robustness can also mean that the parser assigns the best possible annotation to those structures which it cannot annotate definitively. The latter strategy is successfully applied in PoS tag annotation which relies on stochastic information. Every token is assigned a PoS tag. In unambiguous cases, the definitive tag is assigned and, in ambiguous cases, the most likely tag is assigned. This strategy is applied because it is assumed that following annotation steps can better deal with uncertain input than with no input at all. However, this strategy also carries some risks because errors which occur at an early stage of annotation can lead to errors in subsequent stages of annotation and, thus, ultimately lead to a complete failure of the parser.

A notorious error in the PoS tagging of German is the distinction between finite and infinitive verbs ending in '-en'. Since, typically, stochastic taggers only take into account the immediate context of a token and the distinction between finite and infinitive verbs in German only becomes apparent in a wider context, this phenomenon causes problems for stochastic taggers. If these problems are not dealt with robustly, the annotation of whole sentences can fail because of the PoS tag error of just one token since structures containing, e.g., two finite verbs or no finite verb, at all, cannot be fit into a correct parse tree (cf. Müller and Ule, 2002). For this reason, it is useful to apply a strategy in which phenomena which cannot be unambiguously annotated are left partially disambiguated and are only completely disambiguated at later stages of annotation in which all relevant information is present. This strategy of partial disambiguation is mostly referred to as *underspecification* in Computational Linguistics and will be referred to by this name in the thesis at hand, as well, although the term is also used in theoretical linguistics (where it comes from) with a similar but not identical meaning. For the case of the distinction between finite and infinitive verbs this means that we simply regard this decision as not final and allow for the correction of those PoS tags in later stages.[2]

[2]The process of correcting wrong tagging decisions in later stages of the parser is described in more detail in section 5.3.

Robustness is essential in language technology applications because not producing a parser output typically means that all subsequent modules do not produce an output either. In those cases, the application as a whole fails (cf., e.g., Butt, King, Niño, and Segond, 1999, Sec. 14.1). An output of partial structures, however, is also an advantage if the parser is not part of a language technology application but is used for the annotation of a corpus for theoretical linguistic purposes. The following two subsections illustrate the advantage of partial output over no output. Although this is not the main issue of the thesis at hand, we show the advantages of a partial analysis over no analysis for language technology application by illustrating it with an example from machine translation (MT). This can be seen as an excursus. Furthermore, the advantage of the strategy for the annotation of corpora for linguistic research is illustrated by showing how a partially annotated corpus can already be used for linguistic queries.

Advantages for Applications

From the syntactic point of view, an MT program ideally has the full syntactic analysis of a sentence including constituent structure and grammatical functions as its input. However, if the full analysis fails, a partial analysis is preferable to the pure string of tokens with PoS tags.

In order to translate a sentence, an MT program can recognize the full syntactic structure of the sentence in the source language and transfer it into the syntactic structure of the target language. Furthermore, transfers on the lexical and the sematic level are needed, which need not be discussed, here. For sentence 4, this means that the constituent ‘die meisten Experten’ has to be recognized as the grammatical function (GF) subject, ‘haben ... unterbewertet’ as the predicate, ‘die deutschen Aktien am neuen Markt’ as the object and ‘in letzter Zeit’ as an adverbial expression. If the sentence is to be translated, for example, into English, in addition to the words in the language, the word order and the constituent order would have to be transferred into English. The glosses and the translation of sentence 4 into English show that there are considerable differences in the linear order of the constituents in the two languages.

(4)	In	letzter	Zeit	haben	die	meisten	Experten	die	deutschen	Aktien
	In	latest	time	have	the	most	experts	the	German	shares

am	neuen	Markt	unterbewertet.
at the	new	market	underestimated.

‘Recently, most experts have underestimated the German shares on the new market.’

If the parser does not work robustly and does not yield any syntactic analysis, the MT program can only translate the sentence from the source language into the target language word by word. This leads to unsatisfactory results as the gloss of sentence 4 shows. If the parser is, however, able to use partial analyses like those described in part I of the thesis at hand, i.e. chunks, topological fields and clauses, then a better translation would be possible even without the explicit annotation of GFs like subject and object. To begin with, clause boundaries are a great help because they help to avoid mixing material from different clauses, which would result in completely unwanted output. The topological field structure can then be used to join the verbal elements in the English sentence which are separate in the German sentence. The fact that the subject is preferably the first GF in the clause can also be used for translation. Since, obviously, the PC cannot be the subject, the first NC would be the candidate.

(5) [$_{\text{CL}}$ [$_{\text{VF}}$ [$_{\text{PC}}$ In letzter Zeit]] [$_{\text{LK}}$ haben] [$_{\text{MF}}$ [$_{\text{NC}}$ die meisten Experten] [$_{\text{NC}}$ die deutschen Aktien] [$_{\text{PC}}$ am neuen Markt]] [$_{\text{VC}}$ unterbewertet]].

Furthermore, the chunk structure facilitates the translation especially of multi-word expressions like ‘in letzter Zeit’ (recently). However, the automatic translation of a sentence becomes the harder the more complex a sentence is and the more it deviates from the expected (i.e. unmarked) constituent order. In those cases, the MT program might yield a clearly unsatisfactory output if a full parse is not provided. However, the decision whether it is better to have an unsatisfactory output or none, at all, is up to the user of the MT program and, thus, not a linguistic one. Still, the example of sentence 4 has shown the advantages of a robust parser for applications like MT programs. These advantages also hold for other applications in which a partial output is to be preferred to no output at all.

Advantages for Linguistic Theory

By providing evidence for certain linguistic phenomena, even unannotated language corpora can serve as an important instrument for the construction of linguistic theories. However, with increasing complexity and decreasing frequency of linguistic phenomena, it becomes more and more useful to query an annotated corpus. In the ideal case, the linguist is provided with a treebank which is fully annotated with syntactic constituent structure and GFs (or even further information). In the ideal case, a query to such a corpus provides the linguist with all intended phenomena and just the intended phenomena. However, the size of fully hand-annotated corpora is restricted and the automatic full annotation of corpora may not be successful in all cases. Instead of using plain text in those cases in which the full annotation fails, it is advantageous if a partial analysis can be used. In the following, we will show how a partial analysis can restrict the results of a query to a corpus such that the query neither over-generates nor under-generates, i.e. such that the query neither yields too many hits which do not contain the intended phenomenon nor does not yield all or at least most of the intended phenomena.

In order to illustrate the advantages of (partial) annotation for corpus queries, we use an example which is also given in Meurers (2005): If we want to query an unannotated corpus for clauses containing more than one modal verb, then we are confronted with the problem in German that these modals can occur in both parts of the sentence bracket and, consequently, be divided by a lot of language material. If we simply generate a query which searches for two modals in one sentence, then the query will over-generate and yield hits like sentence 6, in which the two modals occur in one sentence but not in one clause. But, if we want to exclude such examples by not allowing potential clause boundaries like commas, conjunctions or subordinators in between the two modals, then we do not get examples like the one in sentence 7. By just annotating the topological field structure, which includes the sentence bracket, we are already able to find examples for two modals in one clause not only if these are distant but also if these occur in complex structures like the one in sentence 8. Hence, for the example of two modal verbs in one clause the partial analysis already suffices, and it is, consequently, an advantage to produce a partial analysis, should the full analysis of a sentence fail.

(6) Hertchen Bullert kann nicht verantworten, daß die Kinder ohne
Hertchen Bullert can not justify, that the children without
Ziehharmonikabegleitung singen müssen.
piano accordion accompaniment sing must.

'Hertchen Bullert cannot take responsibility for fact that the children have to sing without the accompaniment of a piano accordion.'

(7)

Für	diesen	Job	muss	man	schon	drei,	vier	Jahre	schweißen	können.
For	this	job	must	one	already	three,	four	years	weld	can.

'For this job, you have to have three or four years experience in welding.'

(8)

Sollen	die,	die	keine	Lust	haben,	etwa	mitkommen	müssen?
Should	those,	who	no	pleasure	have,	really	with come	must?

'Should those who do not feel like it really have to join in?'

In other cases, the partial annotation helps to reduce the number of hits a query yields. If,for example, a linguist wants to scrutinize the modification structure within complex NPs with more than one modifying PP, then a chunked corpus can essentially yield better results than a corpus which is simply PoS tagged. In order to illustrate the advantages of the chunk annotation, we have randomly extracted 2,000 sentences from a corpus of German annotated with our shallow parser and queried them for an NC followed by more than one PC. The query yielded twenty hits. In nine of these cases, the PCs were, in fact, modifiers in a complex NP. We found examples in which the second PC modifies the head noun of the NC in the preceding PC (cf. example 9) and we found examples in which the PC modified the head noun of the whole NP (cf. example 10).

(9)

[$_{NC}$ Achtung]	[$_{PC}$ vor	den	Verpflichtungen]	[$_{PC}$ aus	Verträgen]
respect	for	the	obligations	from	treaties

'respect for the obligations arising from treaties'

(10)

[$_{NC}$ die	Villa]	[$_{PC}$ mit	abfallendem	Garten]	[$_{PC}$ am	Hang]
the	villa	with	dropping away	garden	at the	slope

'the villa on the slope which had a dropping-away garden'

The example shows that a query to a partially annotated corpus can lead to satisfying results despite the over-generation. Robust parsers may, thus not only be used for language technology applications but also for the annotation of language corpora for linguistic purposes. However, it is important to be aware of the limitations of a language corpus which is only partially annotated: Although it is possible to find a lot of examples for a certain linguistic phenomenon, it cannot be ruled out that some phenomena are not found; and it is very likely that the phenomena which are found are not fully representative of the linguistic phenomenon in question. The reason for this is that a query to a partially annotated corpus necessarily has to be simplifying in order not to generate an enormous number of hits which could no longer be processed by the linguist. Instances which do not fit into the simplified query are not found unless they are explicitly searched for. Sentence 11 gives an example for the query for a complex NP containing two PPs: Since there is an adverbial chunk intervening between the head NC and the modifying PCs, this instance would not be found with a query for an NC followed by at least two PCs. One could extend the query such that AVCs are also allowed. However, this would considerably raise the number of false positives (i.e. unwanted hits). Although partial annotation has disadvantages like these, it is still useful because it can be used as a means of proving the existence of a certain phenomenon.

(11) Ich habe [$_{\text{NC}}$ ein Bild] [$_{\text{AVC}}$ hier] [$_{\text{PC}}$ von einem Künstler] [$_{\text{PC}}$ aus Spanien].
I have a painting here from an artist from Spain.

'I have a painting by an artist from Spain, here.'

2.3.2 Modularity of Annotation Components

Similar to the notion of *robustness*, the notion of *modularity* is important from a practical point of view and from the point of view of linguistic theory. As regards the practical aspect, modular arrangement of a parser enhances its maintainability and, consequently, makes it easier to evaluate the parser and to adapt it to different purposes. As regards linguistic theory, it is reasonable to split up the annotation process into different modules because different layers of annotation with different linguistic phenomena should be handled with different methods. These annotation modules are, for example, a tokenizer, a lemmatizer, a PoS-tagging component, a morphological analyzer or a parser. It is, however, also very useful to construct the parser itself in a modular way. From a practical point of view, there are various reasons for splitting up the parsing task into distinct modules. The most important ones are reasons of maintainability, development, evaluation and adaptability, and will be discussed in the following subsections.

Maintainability

A large and complex parser can barely be maintained by one person alone. In order to avoid unwanted interferences between the different collaborators, it is highly advisable to split up the work into distinct tasks to allow for parallel work on those tasks. This strategy, which can best be subsumed under the heading 'one Grammar, many cooks' (cf. Butt, King, Niño, and Segond, 1999, p. 186), allows easy and reliable maintenance of a parser maintained by several linguists. It has, however, to be ensured that the interfaces of the different modules are properly defined because otherwise the different components of the grammar might not correspond as intended. This might occur, for example, if the linguist responsible for the tagger (or the corpus the tagger is trained on) interprets the tagging guidelines in a way other than the linguist responsible for the parsing component.

Development and Evaluation

Evaluation is a prerequisite for the development of a grammar. The advantages of a modular strategy are, therefore, the same for the development and the evaluation process. In order to improve the performance of the parser, modularity makes it possible to exchange modules of the parser to check whether other approaches work better. This is very obvious in the case of, for example, different PoS-tagging approaches because the interface between the PoS-layer and the parsing layer is a very clear-cut one. If the same tagset is used, the PoS tagger can simply be exchanged and the effect on overall parsing results can be investigated. However, it is also possible to construct the parser itself in a modular way. One very common way is to do a shallow parse before the full parse as is proposed in the thesis at hand. Different approaches can then be applied on top of the shallow parsing component. In this way components specific to a certain task can be constructed.

Even given these components, the parser can be constructed in a modular fashion as figure 2.7 shows. First, the text is annotated by the shallow parsing component (KaRoPars). Then morphological information is added and (partially) disambiguated. Subsequently GFs are annotated first by the support verb component, then by the lexical verb component and then by the non-lexical component. This process is described in detail in chapter 9. The modular construction of the parser makes it possible to evaluate the effect of every single component and, thus, to measure their contribution to the

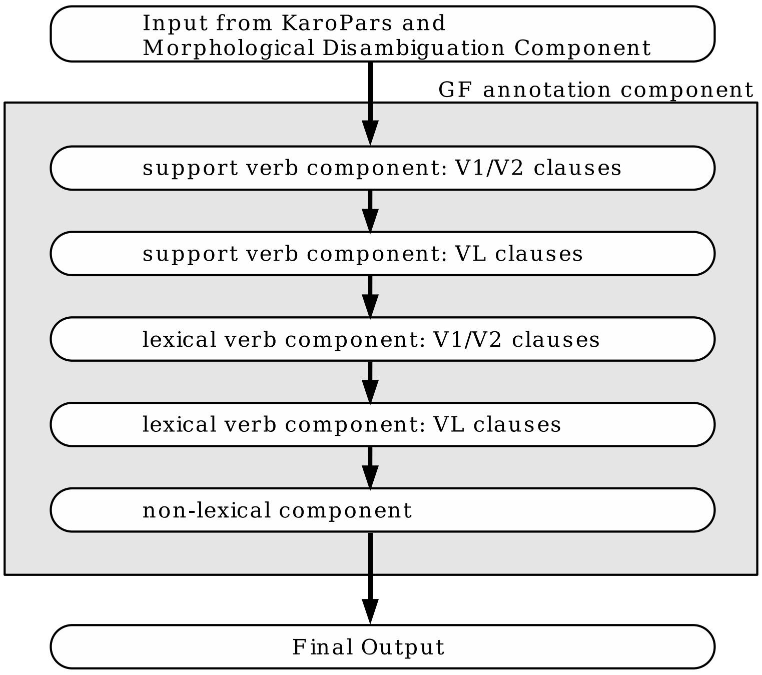

Input from KaroPars and
Morphological Disambiguation Component
GF annotation component
support verb component: V1/V2 clauses
support verb component: VL clauses
lexical verb component: V1/V2 clauses
lexical verb component: VL clauses
non-lexical component
Final Output

Figure 2.7: Modular construction of a parser

overall success of the parser. This is crucial for development as well since it also shows the weaknesses of the parser. Furthermore, a modular structure of the parser can be used to annotate phenomena which are only relevant for certain defined purposes. If, for instances, a certain purpose requires the annotation of Named Entities, a component for this can be easily integrated into the parser.

Adaptability

A syntactic parser might be used for different purposes. It can, for instance, be used for annotating a corpus for linguistic research; it can be used for an MT system; or it can be used for an Information Retrieval (IR) application. All these different purposes require different kinds of annotation. A modular construction of the parser supports 'easy customization' (cf. Neumann and Piskorski, 2002). This means that the parser consists of domain-independent modules which annotate up to a very general level of annotation and of domain-dependent modules which do the specific annotation. The parser consists of a 'core' and of additional components which can be used for various tasks. By adopting a modular approach, it is far easier to adapt the parser to different purposes. It also has to be kept in mind that various text types may differ a great deal. With the approach at hand, it is possible to adapt a parser to specific text types without losing the capability of serving as a general purpose tool.

Different Modules for Diverse Phenomena

Diverse linguistic phenomena require a variety of annotation methods. Consequently, it is useful to split up the annotation tasks into several steps along the lines of the different methods applied. There are various reasons for this. One of them is the efficiency of the parser. This can best be shown with the example of shallow structures vs. GFs. While the annotation of shallow structures is essentially a matter of syntactic restriction, the annotation of GFs is more a matter of lexical selection. Shallow structures, on the one hand, and GFs, on the other, are, therefore, at different levels of linguistic description. To annotate shallow structures, one just needs PoS tag information, to annotate GFs, one also needs sub-categorization information about the respective verb (or other selector) and morphological information about the potential GF targets. As the annotation of the latter clearly needs more linguistic information, which in turn will take more processing time, it is advisable not to use the same 'expensive' method for shallow structures as for GFs. This fact has been pointed out in Abney (1991) as regards the separation of the chunker and the attacher in his parser:

> *By having separate chunker and attacher, we can limit the use of expensive techniques for dealing with attachment ambiguities to the parts of grammar where they are really necessary – i.e., in the attacher.* (Abney, 1991)

The difference between the two phenomena of shallow structures and GFs can be illustrated by the invented example in sentence 12. The order of the tokens in the NPs 'den lieben Sohn' and 'der alte Vater' is fixed. First comes the determiner, then the adjective and then the noun. This order purely depends on the part-of-speech and cannot be violated. The same is true of the position of the finite verb in clauses of this type. It can only come second. In contrast, the order of the subject and the (direct or indirect) object may vary. Either the object or the subject may come first. The existence of the GFs subject or object, in the first place, depends on the verb. Not all verbs subcategorize for both a subject and an object. Furthermore, only the first NP in sentence 12 can be the object and only the second the subject because of the morphological features of those NPs. This illustrates that a lot more information is needed to annotate GFs than shallow structures which is why we separate these tasks in the thesis at hand.

(12) [$_{\text{Obj}}$ Den lieben Sohn] sieht [$_{\text{Subj}}$ der alte Vater].
The dear son sees the old father.

'The old father sees the dear son.'

There are also other reasons for the separation of annotation steps, which deal with different linguistic phenomena and, therefore, with different annotation methods. Some correspond with those already mentioned for the advantages of modularity from a practical point of view: It is much easier to maintain, develop and evaluate a parser in which different methods are implemented in different modules because it is much easier to exchange or improve one annotation method if it is clearly separated from the others because, otherwise, the improvement of one method might lead to unwanted effects in other annotation methods. Chapter 11 is devoted to the evaluation of the GF annotation, and a central issue is the effect of different components of the parser on the overall results. Thus, the effect of the morphological disambiguation component or the impact of the components working with or without subcategorization information can be investigated in detail.

It has, however, to be pointed out that the modular treatment of different linguistic phenomena does not mean that the annotation of the linguistic phenomenon treated by that special module has been fully completed. In fact, subsequent modules may alter the annotation of preceding modules because they may have evidence which was not present at a preceding level. However, the important point is that the different modules do not directly interact with each other. Only this could lead to unwanted interferences.

2.3.3 Underspecification

Underspecification in the context of automatic parsing means that structures which cannot unambiguously be annotated on one level of annotation are left partially ambiguous. They are disambiguated on later levels of annotation if enough information is present to unambiguously decide which kind of structure exists or which category a certain structure has. Thus, although the parser is constructed in a modular way along the lines of different linguistic phenomena – it is possible to alter or extend annotation which has been added on a previous level. Hence, there is a certain permeability of levels. The modules themselves are self-contained but the different levels of annotation are not completely separate from one another. Information from subsequent levels of annotation can 'ooze through' to preceding levels.

The reason for the application of the concept of underspecification is very much connected with a phenomenon which one may call the *missing information dilemma*. Every linguist annotating a corpus is faced with this dilemma: In order to annotate certain local linguistic phenomena, one needs information about the grammatical structure of the whole sentence but, in order to annotate the whole sentence, one needs information about local linguistic phenomena. One example of this dilemma is connected with the distinction between finite and infinitive verbs. For the annotation of clause structure, it is crucial to know whether a verb is finite or not. However, this cannot always be decided on the PoS level by state-of-the-art taggers because it already requires information about the whole clause structure. This is – quite obviously – unknown at the beginning of the annotation process. Thus, the ambiguity cannot reliably be resolved. In our examples 13 and 14, in which we focus on the ambiguity between finite and infinitive forms of the verb ‘herstellen’, even other indicators of the correct PoS tag like the finite verb ‘lassen’ in sentence 13 and the relative pronoun ‘die’ in sentence 14 are ambiguous, because, from the point of view of the bare token, ‘lassen’ could be both finite and infinitive and ‘die’ could be both an article and a relative pronoun.

(13)	Damit	lassen	sich	große	Mengen	von	Dünger	herstellen.
	With this	let	themselves	large	quantities	of	fertilizer	produce.

‘Large quantities of fertilizer can be produced with this.’

(14)	Das	sind	Firmen,	die	große	Mengen	von	Dünger	herstellen.
	Those	are	companies,	which	large	quantities	of	fertilizer	produce.

‘Those are companies which produce large quantities of fertilizer.’

One of the most widely acknowledged solution strategies for the missing information dilemma is the *easy-first parsing* strategy: ‘We make the easy calls first, whittling away at the harder decisions in the process.’ (Abney, 1996c) This strategy means that structures which can reliably be annotated are annotated first, and ambiguity of other structures is thereby reduced (*containment of ambiguity*). A similar strategy implemented for Italian is described by Federici, Montemagni, and Pirrelli (1996):

> *The “minimalist” approach to shallow parsing described in these pages segments a text into units which can be identified with certainty on the basis of the comparatively little amount of linguistic knowledge available at this stage. [...] The chunking process stops at that level of granularity beyond which the analysis gets undecidable, i.e. whenever more than one parsing decision is admissible on the basis of the available knowledge.* (Federici, Montemagni, and Pirrelli, 1996)

The important point here is that parsing stops when decisions become ambiguous – but only on the basis of the knowledge available at this stage. Structures are not ambiguous as such but could be resolved with more knowledge. But, for reasons of effectiveness, they are resolved at a later stage. This part of the strategy is still very much in line with the *easy-first parsing* strategy described in Abney (1996c). However, the strategy is also enriched by the concept of *underspecified chunk categories*:

It is not always the case that the category of a chunk can be identified unambiguously given the available knowledge: this problem can be got around through use of underspecified chunk categories, [...] The chunking process resorts to underspecified categories in case of systematic ambiguity. (Federici, Montemagni, and Pirrelli, 1996)

This concept differs from what is described in Abney (1996c). While the concept in Abney (1996c) is rather one of underspecification of annotation **structure**, the concept in Federici, Montemagni, and Pirrelli (1996) is one of underspecification of both **structure and categories**. If structures are recognized correctly, but the categories cannot definitely be assigned, an ambiguous chunk label is assigned. In case of ambiguous structures like 'un'immagine colorata', where 'colorata' can be interpreted as either an adjective or the participle of a verb, a chunk is introduced which does not resolve the ambiguity but describes it by merging the label to the portmanteau label ADJPART_C (**ADJ** for adjective, **PART** for participle and **C** for chunk).

Our parser makes use of both kinds of underspecification, i.e. it leaves syntactic structures underspecified if they cannot reliably be annotated at that point, and it leaves categories of syntactic constituents underspecified if disambiguation is not possible at that stage of annotation. An example of the former kind of underspecification is that we first annotate shallow structures with our parser before we annotate grammatical functions; an example of the latter kind is that we do not distinguish between the adjectival and the adverbial use of the tokens tagged ADJD in the shallow parser. The examples of the latter kind are closely connected with the concept of *ambiguity classes*.

Ambiguity classes allow us to assign underspecified labels. An ambiguity class is a class of labels which subsumes all the categories which describe a certain linguistic phenomenon at a given point in the annotation process with the knowledge available at that very point. In our parser, ambiguity classes are introduced to come to terms with the missing information dilemma by breaking up the strict separation of the annotation of different linguistic phenomena, while still using the advantages of a modular annotation architecture.

Ambiguity classes have already been introduced for the CLAWS C5 tagset for the *British National Corpus* (BNC) (cf. Garside, 1987; Leech and Smith, 2000) and are recommended by the *Expert Advisory Group on Language Engineering Standards* (EAGLES) for PoS tagsets (Leech and Wilson, 1996).

There are different kinds of ambiguity classes, which can best be defined along the lines of the amount of information they provide. If, for instance, all the possible PoS tags for a token are listed without any preferences for their actual occurrence, this can be called an *unordered ambiguity class*. One example for this kind of ambiguity class is tag VVN-VVD from the BNC Version 1. This tag shows that the PoS tagger was not able to clearly decide between the participle form of a verb (VVN) and the past tense form of a verb (VVD). Thus, instead of making too many incorrect decisions, the parser leaves the decisions partially ambiguous and leaves it open to the user or to further annotation steps to fully disambiguate the token.

If a preference for one PoS tag is given without the probability of the preference, the class can be called a *ranked ambiguity class*. An example of this type of ambiguity class is the tag VVN-VVD/VVD-VVN from the 2nd version of the BNC, where a preference for one of the two tags is given by placing the preferred tag first in the ambiguity class tag. VVN-VVD shows preference for participle and VVD-VVN for past tense (cf. Leech and Smith, 2000). The highest amount of information is offered by a model in which the preferences for different PoS tags are assigned their probabilities. This class can be called a *weighted ambiguity class*.

Theoretically, an ambiguity class can contain as many tags as the tag set minus one (since, otherwise, the phenomenon would be undefined). In practice, however, the number of possible tags is limited. In the BNC, it is limited to two tags. For weighted ambiguity classes it is, for instance, possible to limit the number of tags to all tags with a probability of more than 10% because other tags are most likely useless for further annotation. There is also one further case of ambiguity class, which can be called an *implicit ambiguity class*. In this case, there is no class of different PoS tags, but one tag in itself contains different parts of speech. The reason why there is just one tag for two (or more) clearly distinguishable parts of speech is very often that they are morphologically completely ambiguous and that their ambiguity is not resolvable without a full parse. One example for this implicit ambiguity class is the treatment of 'haben' (*have*) in the standard German PoS tagset, the STTS (Schiller, Teufel, and Thielen, 1995). All instances of 'haben' are tagged VAFIN/VAINF/VAPP (auxiliary verb), although 'haben' may also function as a lexical verb. However, this distinction can only be grasped in a full parse. We adopt this implicit ambiguity class and do not distinguish between lexical and auxiliary use of 'haben' until we annotate GFs. Only at that stage do we check whether 'haben' selects complements and is, thus a lexical verb, or whether it is the auxiliary verb of lexical verbs.

Working with ambiguity classes has many advantages: Primarily, the *easy-first parsing* strategy, which has hitherto mainly been used to annotate the least ambiguous structure first, can now be applied to constituent categories. For instance, the decision between adjectival and adverbial use of chunks headed by tokens tagged ADJD does not have to be taken on a level on which it cannot be decided. Moreover, information which would be lost if just one category label was selected can be retained. Furthermore, although most annotation modules are not perfect – however low their error rate may be – annotation can progress to subsequent levels trying to correct annotation errors of preceding levels. Generally speaking, ambiguity classes provide a flexible means of dealing with information acquired during the annotation process. They allow fine distinctions being made on a lower level of annotation, although those distinctions cannot reliably be resolved when this level is annotated. The ranked or weighted ambiguity classes make it, for instance, possible to assign PoS tags at a point at which some of them cannot unambiguously be assigned.

Two practical examples further illustrate the advantages of this strategy: In German, there are tokens which – from a morphological point of view – appear to be either adverbs or predicative adjectives. A distinction between the two functions cannot be made without taking into account the wider context of the token. This might be one reason for them to be assigned one common PoS tag (namely ADJD) in the STTS. The tag ADJD is, therefore, – according to our definition – an implicit ambiguity class. Thus, there is only a partial disambiguation on the PoS level. However, this suffices already for the chunking task as the difference between adverbs and predicative adjectives is only tackled at later stages of the annotation process.

But, as regards some of the ADJDs, partial disambiguation can already be achieved on the chunking level. If an ADJD clearly occurs within a noun chunk, it is treated like an adverb and becomes the head of an adverb chunk (AVC) because predicative adjectives cannot occur within noun chunks (cf. example 15[3]). Thus, partial disambiguation proceeds, reducing ambiguity while iterating through the different levels of annotation. However, sentence 16 and 17 illustrate that, very often, in order to determine whether the ADJD in question is an adverb or a predicative adjective, one has to consider the governing verb. Consequently, disambiguation can only be achieved when

[3]Only the relevant annotation is given in these examples.

at least clause boundaries are detected and the partially ambiguous chunk in question can be assigned to the verb either as its predicate (in case of the copula ‘sein’; see sentence 16) or as an adverbial (in the case of most lexical verbs; see sentence 17). Until that stage, the chunk headed by an ADJD receives the ambiguity label AJVC (for adverb or predicative adjective chunk).

(15) [$_{NC}$ eine [$_{AJAC}$ [$_{AVC}$ gut/ADJD] schmeckende/ADJA] Suppe]
a good tasting soup

‘a soup which tastes good’

(16) Die Suppe war/VAFIN in der Tat [$_{AJVC}$ [$_{AVC}$ sehr] gut/ADJD].
The soup was indeed very good.

‘The soup was very good, indeed.’

(17) Er kocht/VVFIN in der Tat [$_{AJVC}$ [$_{AVC}$ sehr] gut/ADJD].
He cooks indeed very good.

‘He cooks very well, indeed.’

Another example is the ambiguity between finite and infinitive verbs. But, unlike the tokens tagged ADJD, finite vs. infinitive verbs do not receive an implicit ambiguity class but a weighted ambiguity class. In our case, this means that the most likely tags are assigned with both ranks and probabilities. As some of the tokens in this class are fairly unambiguous, parsing can begin after PoS tagging. The parser simply takes the first-ranked PoS tag and starts to build structure. There is, however, an error rate in the tags concerning the finite/infinitive distinction which is approximately 1.75 times higher than the average error rate (Müller and Ule, 2002).

The main reason for this is that – unlike with other tokens which are morphologically ambiguous – the ambiguity of morphologically ambiguous verbs cannot be resolved within the local context which is usually taken into account by standard PoS taggers. Thus, special attention has been paid to verbs and in those cases in which the modules dealing with the parsing of the verbs fail because the tag of the verb does not fit into any parsing scheme, the parser tries to fit any of the other tags in the ambiguity class into the parsing scheme. The concept of ranked and weighted ambiguity classes thereby helps to ‘repair’ tagging errors on later levels of annotation.[4]

[4]See section 5.3 for details.

2.4 The *fsgmatch* Annotation Tool

We have discussed above the theoretical background of the parser presented in the thesis at hand. In this chapter, we will show some examples of annotation rules such that the actual implementation of our parser, which is presented in subsequent chapters, is illustrated. The annotation tool with which the parser is implemented is the transducer *fsgmatch*[5], which is the main component of the text tokenization tool TTT (Grover, Matheson, and Mikheev, 1999; Grover, Matheson, Mikheev, and Moens, 2000).

The transducer *fsgmatch* fulfills all the requirements which we have outlined as basic principles which should apply to a parser in section 2.3. Since the tool is a transducer, it can be constructed such that it works robustly on input. If no structure can be assigned to the input, the parser simply returns the unchanged input as the output. Thus, subsequent components can still be applied to the data and annotation does not fail completely; and in the case that a backup component can assign structure, it does not fail at all. Furthermore, the transducer allows a modular parser architecture, because it is arranged as a cascade of transducers in which the following transducer receives the output of the preceding transducer as its input. In such a cascade, components can be exchanged without excessive effort during development and the effects being tested. Likewise, other tools can be called between the transducers. In addition, it is easy to leave structure underspecified with the transducer *fsgmatch* because it can add information and also alter or delete it. Decisions which cannot be resolved on an early level may thus be delayed until more information is available.

The general principle of the transducer *fsgmatch* is that it reads in a stream of data and adds (or sometimes alters or deletes) information and then outputs the stream of data with the added information such that it is read in by the next transducer. This process is called piping and the complete architecture is called a pipeline of transducers or a cascade of transducers. In this cascade of transducers, information is added incrementally such that the amount of linguistic information constantly grows. This kind of parsing is also referred to as incremental parsing. Incremental parsing is one of the core concepts of our parser.

The transducer *fsgmatch* uses the data structure of XML to store the added linguistic information in a feature-value structure. It can process an

[5]http://www.ltg.ed.ac.uk/software/ttt/tttdoc.html

```
<t f="mit"><P t="APPR"></P></t>
<t f="der"><P t="ART"></P></t>
<t f="gebotenen"><P t="ADJA"></P></t>
<t f="kommunalen"><P t="ADJA"></P></t>
<t f="Neutralität"><P t="NN"></P></t>
```

Figure 2.8: Initial XML structure before parsing

input stream of raw text as well as existing XML structure. As regards our parser, only the handling of XML data is relevant because the input to our parser already consists of text encoded in XML structure (and annotated, for example, with PoS tags). Typically, the parser would wrap some XML elements into a larger unit thereby gradually creating a tree structure. The new element then contains a feature-value pair which indicates the constituent category. Sometimes, an additional feature-value pair would be added to an already existing XML element.

Figures 2.8–2.11 give an example of the incremental building of markup in the XML structure. Figure 2.8 shows the initial XML structure.[6] The structure consists of a string of tokens ('t'). The token element has a feature 'f', which has the form of the token as its value. The token element contains an element 'P' for PoS tags, which has a feature 't' (for tag) with the respective tag as its value. This structure is the initial input to the transducer *fsgmatch*. Only the value of the feature 't' (i.e. the PoS tag) is used for the annotation of shallow constituent structure. The additional relevant information for grammatical function annotation (e.g., lemma and morphology) is not shown in the examples.

The transducer *fsgmatch* uses a grammar file to assign structure. In the case at hand, the PC is recognized first (figure 2.9).[7] It is marked by wrapping the elements of the PC into a chunk element ('ch'). The category of the chunk is coded as the value of the feature 'c'. As figures 2.10 and 2.11 show, linguistic annotation is added incrementally. Special transducers for each linguistic phenomenon add information building on already existing linguistic information if necessary. In the case at hand, for instance, the search space of the NC and the AJACs is limited by the borders of the PC.

The grammar files used by *fsgmatch* are also coded in XML. *Fsgmatch*

[6]Only the features which are relevant for this example are shown.

[7]We use a top-down approach for chunking, which is described in section 5.5.

```
<ch c="PC">
  <t f="mit"><P t="APPR"></P></t>
  <t f="der"><P t="ART"></P></t>
  <t f="gebotenen"><P t="ADJA"></P></t>
  <t f="kommunalen"><P t="ADJA"></P></t>
  <t f="Neutralität"><P t="NN"></P></t>
</ch>
```

Figure 2.9: Prepositional chunk in XML structure

```
<ch c="PC">
  <t f="mit"><P t="APPR"></P></t>
  <ch c="NC">
    <t f="der"><P t="ART"></P></t>
    <t f="gebotenen"><P t="ADJA"></P></t>
    <t f="kommunalen"><P t="ADJA"></P></t>
    <t f="Neutralität"><P t="NN"></P></t>
  </ch>
</ch>
```

Figure 2.10: NC added to XML structure

```
<ch c="PC">
  <t f="mit"><P t="APPR"></P></t>
  <ch c="NC">
    <t f="der"><P t="ART"></P></t>
    <ch c="AJAC">
      <t f="gebotenen"><P t="ADJA"></P></t>
    </ch>
    <ch c="AJAC">
      <t f="kommunalen"><P t="ADJA"></P></t>
    </ch>
    <t f="Neutralität"><P t="NN"></P></t>
  </ch>
</ch>
```

Figure 2.11: Adjectival chunks (AJAC) added to XML structure

has all the common operators for a regular expression (RE) grammar: Kleene star (none or more elements), Kleene plus (one or more elements) and optionality (element is present or not). Furthermore, disjunction (alternation, i.e. either/or) and grouping (to define the scope of an expression) are possible. Figure 2.12 shows the rule for one type of NC. The element 'RULE' contains the elements to be matched by the RE pattern. The RULE element has a feature 'name' which contains the name of the rule. Rules are called at the end of the rule file (or even outside the rule file in a pipeline) such that the effect of certain rules can be checked quickly for test reasons. An example of this is shown in figure 2.13, in which a disjunction of rules for NCs is called. It is now possible to test with ease the effect of one rule by commenting it out; and in the case that the rules are applied in a sequence, it is possible to change the order of the sequence for test reasons.

In the rule in figure 2.15, all matched structures are represented by subrules, i.e. the actual structures are just referred to and they are at another place in the rule file. The reason for this is that most structures re-occurring frequently and a direct match would thus make the rule file a lot bigger. Furthermore, maintaining the rule file is much easier if changes are only made in one place. Figure 2.15 represents a rule in which the ON occurs in the VF (*Vorfeld*, initial field)[8] and is followed by optional elements.[9] The third structure is an auxiliary verb as the left part of the sentence bracket, then follow optional elements, the OPP with the preposition 'mit', optional elements which cannot be the OA, the OA and, finally, the lexical verb which sub-categorizes for this frame as the right part of the sentence bracket.

The RULE element further contains a feature 'targ_sg' which has the name of an XML element and its feature-value structure as its value. This is the XML element into which the pattern which is matched by the RE is wrapped. In the case in figure 2.12, it is a chunk with the category 'NC'. The REL elements in the RULE element contain the constituents to be matched. The RULE element has a feature which indicates whether these REL elements are to be matched as a sequence or as a disjunction. The default is 'sequence', as applied in the example in figure 2.12. In this case, the feature does not have to be explicitly stated. REL elements have a feature 'type' which indicates the way in which the pattern matching is applied. The default value is that

[8]See section 4 for details of the theory of topological fields.

[9]Typically, there is just one constituent in the VF. There are, however, some few exceptions.

```
<!-- noun chunk introduced by determiner -->

<RULE name="NCdet" targ_sg="ch[c='NC']">
 <REL match="t/P[t='KOKOM']" m_mod="QUEST"/><!-- Vergleichspartikel -->
 <REL type="GROUP" match="DISJ">            <!-- determiner -->
  <REL match="t/P[t~'^(ART|PDAT|PIAT|PIDAT|PPOSAT|PRELAT|PWAT)$']"/>
  <REL match="ch[c='DTC']"/>
 </REL>
 <REL type="REF" match="inNC" m_mod="STAR"/><!-- evthing within NC -->
 <REL type="REF" match="headnoun"/>         <!-- head noun -->
</RULE>
```

Figure 2.12: *Fsgmatch* rule for an NC

```
  <!-- final call of rules for NCs -->

  <RULE name="all" type="DISJ">
      <REL type="REF" match="NCprep"/>
      <REL type="REF" match="NCdet"/>
      <REL type="REF" match="NCadj"/>
      <REL type="REF" match="NCcard"/>
      <REL type="REF" match="NCpure"/>
      <REL type="REF" match="NCpron"/>
      <REL type="REF" match="NCcardhead"/>
  </RULE>
```

Figure 2.13: Rules are called at the end of the rule file

the value of the feature 'match' is matched directly. This is, for instance, a token with the PoS tag 'KOKOM' (for 'Vergleichspartikel') in the first REL element in the rule. The other matching types are those for grouping ('GROUP') and for reference ('REF').

In the rule in figure 2.12, grouping is used to define the scope of a disjunction. In the case at hand, there is the alternative between one of a list of determiners defined by their PoS tags and a (rarely occuring) complex determiner chunk (e.g., 'gar keine', *none at all*). Besides, the feature 'DISJ', RELs of the type 'GROUP' may have two other values for the feature 'match'. These are 'SEQ', which allows us to define the scope of a sequence, and 'DISJF', which takes the first possible match from a list of alternatives. In the case that there is just one possible match, 'DISJF' has the identical effect of 'DISJ'. In fact, 'DISJF' is hardly ever used within rules but it is used when rules are called to allow preference in a set of alternatives.

The third value of the feature 'type' in RELs (i.e. 'REF') is used to indicate that another rule is called by the feature 'match'. It is very useful to code a complex structure in such a subrule as otherwise the main rule[10] would become opaque. It is also useful to use subrules if one structure is used by more than one main rule. Both is the case in the NC rule. There are two subrules to which is referred. One is the subrule for a headnoun (e.g., a noun, a proper noun, capitalized foreign material (FM), a number). The rule 'headnoun' lists a number of alternatives and is called by nearly all rules for NCs. This makes the rules more easily manageable and transparent. The other subrule (i.e. 'inNC'; cf. figure 2.14) contains all the elements which occur in an NC. Since this rule is also quite extensive, the same advantages as for the headnoun rule apply. Figure 2.14 shows that all the elements are listed explicitly. It would have been possible to code the PoS tags in one regular expression. However, this would have been against transparency and maintainability. Furthermore, explicit coding allows for more transparent commenting of elements, which is, again, crucial for maintenance.

It is important to note that the possibility to use subrules does not mean that *fsgmatch* allows recursion and is, thus, more powerful than an RE language. In fact, a rule can never call itself, neither directly nor indirectly via a subrule. In other words: The structure of rules is strictly hierarchical and, thus, a one-way relation. Although, in principle, there can be infinitely many subrules, this does not mean that the language described is not regular

[10]If we talk of 'main rules', we mean rules which create annotation.

```
<!-- subrule for everything within NC -->

<RULE name="inNC" type="DISJ">
  <REL match="t/P[t='PTKNEG']"/>    <!-- negative particle -->
  <REL match="t/P[t='ADV']"/>       <!-- adverb -->
  <REL match="t/P[t='CARD']"/>      <!-- cardinal -->
  <REL match="t/P[t='ADJD']"/>      <!-- modifying adjective -->
  <REL match="t/P[t='ADJA']"/>      <!-- attribut. adjective -->
  <REL match="t/P[t='KON']"/>       <!-- conjunctions -->
  <REL match="t/P[t='$,']"/>        <!-- commas -->
  <REL match="t/P[t='$(']"/>        <!-- hyphens and suchlike -->
  <REL match="t/P[t='KOKOM']"/>     <!-- Vergleichspartikel -->
  <REL match="ch[c='AJACTRUNC']"/>  <!-- attribut. adj. TRUNC -->
  <REL match="ch[c='CARDCOMPLEX']"/><!-- complex card. number -->
</RULE>
```

Figure 2.14: Subrule for all elements which may occur in an NC

because this simply means that the RE may, in principle, be infinitely long. Subrules could, in fact, be spelled out in one rule and are simply a means to make rules appear more transparent.

Another feature of the REL element allows the application of quantifiers. The feature 'm_mod' can have the values 'STAR' for Kleene star, 'PLUS' for Kleene plus, 'QUEST' for optionality and 'TEST' to test context. The 'TEST' feature is not to be confused with what makes context-sensitive languages context-sensitive, i.e. the capability of a context-sensitive rule to be applied recursively to its output. If a context is tested and the site of the application of this context is in one direction only, context testing can, in fact, be modelled within the finite-state paradigm as Karttunen and Beesley (2001) have emphasised. The value 'TEST' of the feature 'm_mod' fulfills this requirement.

The same mechanisms of *fsgmatch* which allow the annotation of shallow structures also allow us to annotate grammatical functions (GFs). Figure 2.15 shows the rule which annotates a sentence with a verb subcategorizing for a subject (ON), an accusative object (OA) and a prepositional object (OPP) with the preposition 'mit'. This would, for instance, be a sentence like sentence 18. As its name shows, the rule has the name 'rule' plus an

```
<RULE name="rule2850" targ="&A-REW; &B-VAL; &C-VAL; &D-VAL;
                            &E-REW; &F-VAL; &G-REW; &H-VAL;">
  <REL var="A" type="REF" match="on_vf_w"/>
  <REL var="B" type="REF" match="optionals_vf" m_mod="STAR"/>
  <REL var="C" type="REF" match="vcl_akt_fin_am_t"/>
  <REL var="D" type="REF" match="optionals_mf" m_mod="STAR"/>
  <REL var="E" type="REF" match="opp_mf_mit_w"/>
  <REL var="F" type="REF" match="allbut_oa" m_mod="STAR"/>
  <REL var="G" type="REF" match="oa_mf_w"/>
  <REL var="H" type="REF" match="vcr_akt_nfin_onoaopp_mit_t"/>
</RULE>
```

Figure 2.15: *Fsgmatch* rule for the annotation of GFs

automatically generated number. Since it is not the intention to wrap the whole sentence into one XML element, the feature 'targ' contains a set of Boolean values which describe whether the structure is just to be matched ('VAL' for value) or also to be wrapped into an XML element ('REW' for rewrite). In the case at hand, the grammatical functions are wrapped into XML elements showing the type of function; and the optional elements in the sentence and the verbs are simply matched.

(18) [$_{\text{ON}}$ Sein Chef] hat [$_{\text{OPP}}$ mit diesen Leuten] noch [$_{\text{OA}}$ nichts]
His boss has with these people still nothing
abgeklärt.
clarified.
'His boss has not clarified anything with these people yet.'

Figures 2.16 and 2.17 show as an example what the subrules in figure 2.15 look like. Figure 2.16 shows the subrule which matches the NC functioning as the OA and assigns the grammatical function category. The @ sign at the beginning of the value of the feature 'targ_sg' shows that a feature-value pair is added to an existing XML element and no XML element itself is added. Thus, the NC receives a feature 'func' which shows its function which is, in the case at hand, OA. The matched structure is a chunk ('ch') which has a category 'c' which begins with NC.[11] Furthermore, the NC has to contain an

[11]Regular expression patterns can be used as values of features in *fsgmatch*. This is signified by the use of a '∼' instead of a '='.

```
<RULE name="oa_mf_w" targ_sg="@[func='OA']">
  <REL match="ch[c~'^NC']/morph[d~'^a']"/>
</RULE>
```

Figure 2.16: Subrule which matches accusative object

element 'morph' which has a category 'd' with a value beginning with 'a' (for accusative).[12]

Figure 2.17 shows the subrule which matches the right part of the sentence bracket containing the lexical verb. There is no feature 'targ_sg' because the structure is simply matched and no annotation is assigned. The feature 'type' has the value 'SEQ' because the three elements in the rule are matched as a sequence. In front of the lexical verb, there are none or more optional elements (which are, again, represented by a subrule). After the lexical verb comes an optional comma. Besides the information that it is the right part of the sentence bracket, the rule contains some more information about the chunk to be matched: The potential chunk has a feature 'mode' which, in the case at hand, has to **contain** the string 'akt'. Whether the chunk has the potential to be in an active mode construction is thereby checked. Only in this construction are all GFs in the subcategorization frame realized. This would not be the case with the passive diathesis, in which the OA would become the ON, and there would, thus, be no OA.

On the basis of chunk-internal information, the values 'akt', 'pass', 'akt-pass' and 'aktpp' have previously been assigned. In the first case, the chunk is definitely part of an active construction; in the second case, it is definitely part of a passive one. In the third case, the chunk may be either in an active or passive construction and the ambiguity is resolved in combination with the left part of the sentence bracket, which contains similar information. In the fourth case, the chunk contains a participle which gives it the potential to be in a passive construction. Again, the left part of the sentence bracket is used to resolve ambiguity. This procedure is an example of the concept of underspecification. In those cases in which decisions can be taken at an early level, they are taken; but in those cases in which this is not possible, the ambiguity is just contained (i.e. ambiguity classes are assigned) and full disambiguation takes place at a later stage.

[12]Morphology is assigned and partially disambiguated on pre-parsing levels described in

```
<RULE name="vcr_akt_nfin_onoaopp_mit_t" type="SEQ">
  <REL type="REF" match="opt_mf_t" m_mod="STAR"/>
  <REL match="ch[c~'^VCR' mode~'akt']/hdverb[s~'-ONOAOPP_mit-']"/>
  <REL type="REF" match="comma_t" m_mod="QUEST"/>
</RULE>
```

Figure 2.17: Subrule which matches right part of sentence bracket with lexical verb subcategorizing for ON, OA and OPP with 'mit'

The chunk in figure 2.17 also contains an element which is central to the annotation of GFs. There is an element 'hdverb', which contains a feature 's' (for subcategorization frame). The value of this feature is a concatenation of all the potential subcategorization frames (SC frames) which the lexical verb that is head of the chunk may realize. In the case at hand, the rule checks whether the string '-ONOAOPP_mit-' is a sub-string of this concatenation and whether, thus, the verb sub-categorizes for an ON, an OA and an OPP with the preposition 'mit'. The SC frames are assigned by a simple lexical lookup at an earlier stage of processing. The features of the element 'hdverb', thus, trigger the application of the rules for the respective SC frames. A rule is only applied to a sentence if the relevant verb subcategorizes for the correct SC frame.

While the rules for the shallow parsing component are manually constructed, those for the GF annotation component are semi-automatically constructed. The reason for this is that, first, there are many more rules than in the shallow parsing component. While there are only 72 rules for chunks, there are 9,017 rules for GFs. Second, a lot of the structures in GF annotation are recurrent. A very obvious example is the fact that all the patterns which include an OPP have to occur with all possible prepositions. In a similar way, one set of subrules for the verbal parts of the sentence bracket has to be generated for every single SC frame. For this reason, the rules are coded in a more abstract way and then automatically converted into XML.

This also gives the opportunity to quickly test alternatives because, during the conversion process, systematic changes can be made with little effort. The process of conversion from the abstract patterns to the XML rules takes no more than one minute. In this way, one can, for example, quickly test

chapter 8

```
ON_VF_W VCL_akt_fin_AM_T OPP_MF_mit_W OA_MF_W VCR_akt_nfin_ONOAOPP_mit_T
```

Figure 2.18: Abstract coding of the XML rule in figure 2.15

whether an optional element after the first (and typically only) element in the VF makes sense, or whether verbs which subcategorize for one certain frame can also subcategorize for another certain frame (e.g., whether all verbs which subcategorize for an OA also subcategorize for a reflexive OA).

Figure 2.18 shows the abstract coding of the pattern which is matched by the rule in figure 2.15. From this pattern, the XML rule in figure 2.15 is generated by a Perl script. In the case at hand, the elements ON_VF_W, OA_MF_W and VCL_akt_fin_AM_T are transformed by patterns in the Perl script which only deal with those very elements. The elements OPP_MF_mit_W and VCR_akt_nfin_ONOAOPP_mit_T are transformed by patterns in the Perl script which treat parts of those elements as variables such that one pattern can be used for many elements, thus facilitating the construction of the parser. In the case of the prepositional object OPP, the variable is the preposition 'mit'. In the case of the right part of the sentence bracket containing the lexical verb, the variable is the SC frame (i.e. ONOAOPP_mit). The last character in each element name simply shows whether the element is assigned a function label ('W')[13] or whether the presence of the element is simply tested ('T').

Optional elements are not explicitly coded in the abstract code. They are inserted automatically by the Perl script. In such a way, the conditions under which a certain element is inserted and the exact composition of the optional elements can be quickly changed for test reasons. The patterns as they are shown in figure 2.18 are semi-automatically constructed. If a verb sub-categorizes for three GFs then the combination of positions is to a certain extent the same for all SC frames with three GFs. The patterns for one SC frame can therefore be converted easily into the patterns for another SC frame.

[13]The abbreviation of this feature was more obvious at the beginning of the construction of the parser, and it was not changed later on.

Chapter 3

Related Work

ABSTRACT: Chapter 3 introduces approaches which are in some way related to the one in the thesis at hand – either because they use a similar method or because they annotate the same linguistic structures. Section 3.1 gives some general remarks about comparing different parsing approaches especially for German. Section 3.2 first introduces the situation for annotating shallow structures, then presents the relevant related approaches and, finally, relates the approaches to each other. Section 3.3 gives a short introduction to approaches annotating GFs and discusses those relevant for the thesis at hand. A conclusion is given which compares the approaches.

3.1 General Remarks

There may basically be two motivations for the construction of a new parser: First, the constructed parser uses an innovative method and, second, the parser yields considerably better results than existing ones. The ideal case is a combination of those two motivations: i.e. it is possible to show that a certain new method is able to provide considerably better results. In order to compare different parsers, it is necessary to describe the methods they use and the linguistic information which they can fall back on. We do this in this thesis in the relevant sections. Furthermore, in order to compare the results of two parsers, those parsers would have to annotate exactly the same linguistic phenomena in exactly the same texts. Otherwise, results are not comparable.

In order to guarantee comparability, *shared tasks* have been established in which different approaches to one and the same linguistic annotation task are tested on one and the same data under the same conditions (except those in which the particular approaches differ). Examples of these *shared-task scenarios* are presented, e.g., in Chinchor and Robinson (1998) for Named Entity recognition, in Tjong Kim Sang and Buchholz (2000) for chunking and in Tjong Kim Sang and Déjean (2001) for clause identification. However, for several reasons, it is not always possible to establish a shared task scenario. In the case of the parser described in this thesis, the main reasons for this have to be seen in the target language (i.e. German) and the linguistic phenomena which are to be annotated.

For the German language, there are, by far, not as many parsers as there are for English. Large-scale shared-task scenarios like the ones for English quoted above do not exist. Consequently, the scope of comparison is already limited by the amount of parsers which exist. Secondly, until now, no common standard of annotation has been established: The notion of *chunks* is interpreted in different ways by different parsing projects, mainly because of the different purposes of annotated structures. Parsing of topological fields has, from a linguistic point of view, only just begun; and there are several definitions of GFs and in some approaches not all types of GFs are annotated. We will, nonetheless, in the following try to compare our parser with as many approaches as possible in order to give the reader an idea of its position in the NLP field.

3.2 Shallow Parsing

3.2.1 Finite-State Approaches and Shallow Parsing

Cascaded FSTs were already used in the late 1950s to annotate shallow structures (Joshi, 1961; Joshi and Hopely, 1996; Joshi, 1999). However, knowledge about this seems to have got lost during following decades. When Steve Abney introduced his concept of chunks as shallow parsing structures, which were to be annotated by cascades of FSTs in Abney (1991) and Abney (1996c), his concept of chunk parsing appeared to be novel to the scientific community. Several reasons can be given for this late breakthrough of FSTs in combination with shallow parsing. One of the reasons might be seen in the shift in theoretical perspective after the 'generative revolution' in the late 1950s which led to the replacement of structuralism by generativism as the mainstream theory in linguistics.

Another reason may be seen in the availability of considerably greater computing power at the beginning of the 1990s as compared to the 1950s. For a long time, the automatic parsing of language was to a large extent carried out to investigate whether it is possible at all to analyze human language with an automaton. Very often, the analyses of sentences would be very detailed but slow. Additionally, there would be a preference for no output rather than incomplete or even partially erroneous output. By the beginning of the 1990s, however, computing power was great enough to allow for computer applications of automatic parsing and, thus, a parser which would be able to efficiently and robustly annotate linguistic structures. Cascaded FSTs together with the concept of shallow parsing provided the formal background for such parsers.

Another important reason for the promotion of shallow parsing was that Steve Abney thoroughly defined chunks as shallow parsing structures using linguistic arguments. In Abney (1991), chunks are not simply given as structures which can be very easily annotated with a minimum amount of information, but both prosodic and psycho-linguistic evidence is given to support the existence of such chunks. With the help of what Abney (1991) calls *major heads*, a general definition of chunks is given which holds for all different types of chunks. Only after shallow parsing has become an accepted approach, Abney (1995) admits that "[w]e can define *chunks* as the parse tree fragments that are left intact after we have unattached problematic elements" not without acknowledging that "[i]t is difficult to define precisely which elements are ≪problematic≫". In subsequent years, many projects have found their own definition of chunks, mainly using the guideline of "unattaching problematic elements". This is precisely the reason why there are so many definitions of chunks.

Joshi (1961) had already used structures which can quite appropriately be termed chunks. These structures are called "clusters" and consist of phrases without explicit attachment of adjuncts. This definition basically corresponds to the "non-recursive core phrases" defined in Abney (1996c). After Abney (1991), Grefenstette (1996) used FSAs to annotate shallow parsing structures. These structures are called "groups" and the approach is referred to as "light parsing". Essentially, what Grefenstette (1996) calls, for example, nominal groups and verbal groups are the same as chunks. A further approach to shallow parsing using FSAs is presented in Aït-Mokhtar and Chanod (1997a). In this approach, "segments" are annotated which are described as “partial construction[s] (e.g., a nominal or a verbal phrase, or a more primitive/partial syntactic chain if necessary)”. Aït-Mokhtar and Chanod (1997a) use a slightly different approach to Abney (1991) in that they allow the later revision of decisions made at earlier levels of the parser.

Below, we will discuss those approaches to shallow parsing of German which come closest to our approach either with respect to the method used or with respect to the structures which are annotated, or both. We do this in order to present the scientific context in which our shallow parser has to be seen.[1]

3.2.2 Braun (1999); Neumann, Braun, and Piskorski (2000); Neumann and Piskorski (2002)

The shallow parsing approach presented in Braun (1999); Neumann, Braun, and Piskorski (2000) and Neumann and Piskorski (2002) is the one which comes closest to ours, since it also uses FSAs and annotates topological fields, clauses and chunks (in this order). Topological fields and clauses are annotated first to make use of topological field information in the annotation of phrases. This is possible because the scope of certain phenomena is restricted to clauses or topological fields. Neumann, Braun, and Piskorski (2000) call this strategy a “Divide-and-Conquer Strategy” after Peh and Ting (1996), who first applied this strategy for NLP. Before, this strategy had mainly been used in programming.

The approach takes tokenized text as its input. Tokens are then annotated with PoS tag ambiguity classes using the tool MORPHIX (Finkler and Neumann, 1988). MORPHIX analyzes the morphological structure of a token. Since the PoS tag is assigned on the basis of morphology only, the potential PoS tags are assigned as an ambiguity class and PoS tag filtering is applied afterwards. This is achieved by using (obviously hand-constructed) contextual filtering rules which are compiled into a single FSA. Some tokens remain ambiguous to a certain extent, i.e. some few tokens remain with more than one reading after PoS tag filtering.

In the first phase of parsing, the topological field structure of a sentence is identified, thereby also allowing the annotation of clauses. In a second phase, “simple, non-recursive [phrases]” are identified (Neumann and Pisko-

[1]A detailed comparison of the evaluation results of our approach and of the other approaches presented here is given in chapter 12.

rski, 2002, p. 25). These can be seen as equivalent to chunks. Furthermore, Named Entities are annotated and the structure of compound nouns is analyzed. The parser uses cascaded FSAs for annotation and, as a consequence, has a modular architecture which allows for the integration of new and/or specialized components. In contrast to our parser, morphological information (agreement check) is used for the annotation of chunks. For this component, "a type description language for constraint-based grammars" is used (Neumann, Backofen, Baur, Becker, and Braun, 1997). It does not become clear, however, to what extent this component is still within the finite-state formalism.

3.2.3 Wauschkuhn (1996)

The approach described in Wauschkuhn (1996) only comes close to our approach as regards the order of parsing and the fact that topological fields are annotated. This approach is the first one which first annotates topological fields and clauses and then goes on to annotate phrases. Wauschkuhn (1996) calls this approach "zweistufiges Analyseverfahren" (*parsing method working in two steps*)[2]. Wauschkuhn (1996) gives robustness as the main motivation for his approach. If it is not possible to assign the whole structure of a clause (including the phrase structure), it is at least possible to assign the topological field and clause structure. This structure is seen as the one that is easier to recognize by Wauschkuhn (1996). That is the reason why it is annotated, first. The advantages described by the concept of a Divide-and-Conquer Strategy are not mentioned in Wauschkuhn (1996).

Although, according to what has been stated above, the approach in Wauschkuhn (1996) can in some ways be seen as similar to the one described in Neumann, Braun, and Piskorski (2000) and in the thesis at hand, it is very different in other respects. It is not a cascade of FSAs which is used by Wauschkuhn (1996) but a chart parser using context-free grammars. This chart parser is not deterministic, i.e. it does not produce one single result but, in cases of ambiguity, outputs all possible results. Unfortunately, there is no qualitative evaluation of these results because no appropriate corpus existed at the time of the construction of the parser; and we are not aware of any subsequent evaluation of it. Thus, the main merit of Wauschkuhn (1996) has to be seen in the fact that it first introduced the parsing of topological

[2]Translation taken from the English abstract in Wauschkuhn (1996).

fields and that it showed that this was possible **before** the annotation of phrases.

3.2.4 Schiehlen (2002, 2003a,b)

The shallow parsing approach presented in Schiehlen (2002, 2003a,b) is also very close to our approach. Schiehlen (2002, 2003a,b) takes tokenized text as input which is STTS-tagged and morphologically enriched. The text is annotated with chunks and clauses by a cascade of FSTs like the one described in Abney (1996c). However, in contrast to Abney (1996c), recursive structures are also dealt with, as this is also the case in our approach. For instance, Abney (1996c) only recognizes simplex clauses but Schiehlen (2002, 2003a,b) also recognizes embedded clauses. The parser is deterministic but a non-deterministic component follows the basic component, which annotates further structures such as coordinated VPs. The annotation of what is called grammatical roles and which subsumes what we call grammatical functions is also non-deterministic. There is no annotation of topological fields whatsoever. In cases of ambiguity, a longest match strategy is applied.

An important difference between our approach and the one in Schiehlen (2002, 2003a,b) is that we do not apply any agreement checking in the annotation of chunks although we do have this information and are using it in the annotation of GFs. However, the cases in which agreement checking would make any difference in chunking are so rare that we decided not to use this information because the effort is disproportionate to the benefit to be gained. Far more errors occur for less trivial reasons. Furthermore, it is not clear to what extent this component based on a constraint-based grammar is within the finite-state formalism (as this was also the case in Neumann, Backofen, Baur, Becker, and Braun (1997) above).

There are basically two approaches to check agreement in chunking. One is to check agreement while chunking is in progress and the other is to apply the agreement check after chunking. If one uses FSAs, the former approach might result in a huge grammar because all potential combinations have to be spelled out. The latter approach, on the other hand, involves a complex reconstruction procedure because, if a non-agreeing structure has already been annotated, it has to be deleted and re-structured. Schiehlen (2002, 2003a,b) tests both approaches.

3.2.5 Kermes and Evert (2002, 2003)

The approach in Kermes and Evert (2002, 2003) also comes close to our approach. Text is tokenized, STTS-tagged and morphologically enriched using IMSLex morphology (cf. Lezius, Dipper, and Fitschen, 2000) before it is annotated with chunks by a RE grammar. As in our approach, centre-embedded structures (e.g., PPs modifying the attributive adjective as in figure 3.1)[3] are annotated and post-head PPs are not attached. Unlike in our approach, post-head genitive NPs are attached. We also have information about case in our parser but do not use it in shallow parsing. We also do not see genitive NP attachment as part of the shallow parser. In contrast to our approach, Kermes and Evert (2002, 2003) use both lexical (including lemma) and morphological information (case, gender, number) in their chunker. We explicitly exclude this information from the shallow parsing process to keep the parser as simple as possible for reasons of speed and efficiency. Morphological information is only used in the GF annotation component of our parser.

Since the parser in Kermes and Evert (2002, 2003) uses an RE grammar, it is equivalent to an FS parser. However, in Evert and Kermes (2003) it is stated that "the CQP query language and especially the Perl code used in post-processing are more expressive, and quite often more convenient than standard finite-state technology". Thus, in contrast to our approach, there are extensions to the FS formalism in Kermes and Evert (2002, 2003), although it is not exactly clear to what extent the FS formalism is extended. Parsing is performed incrementally and bottom-up in Kermes and Evert (2002, 2003). This is in contrast to the parser at hand since our parser is a mixed bottom-up top-down parser which first annotates topological fields (which are not annotated in Kermes and Evert (2002, 2003)), then clauses and then chunks (these are annotated top-down).

Furthermore, candidate structures are generated which may be rejected by later levels of the parser. The parser is thus not deterministic. One example is the treatment of centre-embedded structures. In Kermes and Evert (2002, 2003) candidates for centre-embedded structures are generated and only accepted if, on the last level, there is a clear indicator of centre-embedding, i.e., e.g., a determiner before the centre-embedded structure. Since we use a top-down approach for chunking, this problem does not occur for us although we effectively use the same information to deal with it. We

[3]Only the relevant information is shown in this figure.

$[_{\text{PP}}$ mit	dieser $[_{\text{AP}}$ $[_{\text{PP}}$	aus	der Not]	geborenen]	Führungspitze]
with	this	by	the necessity	born	top management

‘with this management which was just installed because necessity required it’

Figure 3.1: Centre-embedded PP

also only annotate centre-embedding if there is clear indication for it in the form a determiner or preposition as in figure 3.1.

Kermes and Evert (2002, 2003) have a special component for annotating chunks with a ‘specific internal structure’. These chunks subsume certain named entities like dates, measures etc. Furthermore, chunks in brackets and quotation marks are annotated. While Kermes and Evert (2002, 2003) annotate such chunks before the main chunking component is applied, we annotate named entities before and the other types as chunks after the main chunking component.

3.2.6 Trushkina (2004)

The parser described in Trushkina (2004) annotates chunks and topological fields as constituents and GFs as dependencies. It uses the Xerox Incremental Deep Parsing System (XIP) as an annotation tool (cf. Aït-Mokhtar, Chanod, and Roux, 2002). Although this approach is no longer within the FS formalism (cf. Aït-Mokhtar, Chanod, and Roux, 2002, p. 131) in contrast to Aït-Mokhtar and Chanod (1997a,b), it would be possible to implement the grammar as a cascade of FSTs from what is stated in Trushkina (2004), sec. 4.3.2 and 7.3.2: Rules are organized in layers, annotation is deterministic (i.e. already annotated structure is never discarded) and recursion is covered by iteration of rules. Context-testing may be applied. We have emphasized, however, that context-testing can be modelled within the FS formalism as shown by Karttunen and Beesley (2001).

The parser described in Trushkina (2004) is called GRIP (GeRman Incremental Parsing system). As the name suggests, parsing is done in an incremental fashion: The system uses tokenized text as its input. Tokens are then processed by the morphological analyzer developed at the Xerox Research Centre Europe (XRCE). This includes lemmatization and PoS tag assignment according to the XRCE tagset (a slightly modified version of the

STTS). A rule-based module subsequently reduces ambiguities in PoS tags, and another rule-based module reduces ambiguities in morphology. Since the following chunker needs unambiguous input, the ambiguity of both PoS tags and morphological analysis is finally resolved by a PCFG. Morpho-syntactic disambiguation is thus achieved in a hybrid way. The disambiguation of morphology is important for the parser in Trushkina (2004) because the annotation of dependencies representing GFs relies on morphological information.

The chunker mainly relies on PoS tag information and annotates roughly the same structures as the shallow parser in the thesis at hand: Chunks, including centre-embedding and excluding post-modification, topological fields and clauses. It is thus, according to our definition, a shallow parser. Since the shallow parser only relies on PoS tag information and on "limited lexical information" (p. 130), there remains the question of why morphological disambiguation is not tackled **after** the shallow parsing process because the valuable information of shallow parsing structure could then be used. The advantage would be that the checking of agreement information within chunks could make use of the chunk structure; the checking of agreement between potential subject and verb could make use of clause structure; topological field information might also have been used, e.g., the restriction that there can only be one constituent in the VF.

Disambiguating morphology before shallow parsing has the effect that the parser in Trushkina (2004) makes it, in fact, necessary to formulate many rules twice. In order to check the agreement in a potential NC, the linear order of the elements in it has to be formulated for the morphological disambiguation component. Afterwards the same information has to be used in the shallow parser to construct the respective chunk. It would, thus, be more logical to apply agreement checking after shallow parsing. Another desideratum for the shallow parser would be a more detailed evaluation since only the overall figures are given but, very often, figures differ considerably between various categories.

3.2.7 Schmid and Schulte im Walde (2000)

The following approach is very different to ours. Schmid and Schulte im Walde (2000) apply a hybrid technique for the annotation of chunks. First, text is tokenized and PoS-tagged with partially disambiguated tags using DMOR (Schiller, 1995). DMOR analyzes tokens morphologically and also assigns them PoS tags on the basis of this morphological analysis but without any contextual information. Since, on the basis of its morphology, a token may have more than one analysis, a token may have more than one PoS tag. Schmid and Schulte im Walde (2000) therefore have a manually constructed context-free grammar (CFG) to hand, which contains 4,619 rules and which covers 92% of the 15 million token corpus they use. This CFG can be trained on an unlabelled corpus and, thus, becomes a probabilistic context-free grammar (PCFG). Since also the heads of phrases are taken into account, the grammar is a head-lexicalized PCFG (H-L PCFG). The advantage of such a grammar – once it is constructed – is that it can be easily tuned to the text type it is intended for, especially since it can be trained unsupervised, i.e. on unannotated text.

However, the disadvantage is that the grammar does not cover all structures in the corpus (in this case 8%) and is thus not robust. Still, robustness is crucial for a chunker as we have stated in section 2.3.1. To achieve robustness, Schmid and Schulte im Walde (2000) extend the chunk part of the H-L PCFG with semi-automatically generated robustness rules which allow for the annotation of unrestricted text. These robustness rules are first automatically generated to cover all possible chunk structures. Then, some of the rules are generalized using linguistic knowledge. The number of rules is thus reduced to 3,332. Furthermore, for the strategy yielding the best results in the end, the base grammar is extended by some "simple verb-first and verb-second clause rules". In order to annotate chunks, the grammar then generates all possible chunks for a clause and chooses the most probable chunk set, i.e. the most probable combination of chunks in one clause and not just the most probable single chunk.

The parser described in Schmid and Schulte im Walde (2000) solely annotates and evaluates noun chunks. Probably, the original H-L PCFG parser also annotates other structures. The NCs annotated include PCs in which the determiner and the preposition are morphologically combined, e.g., 'zum Beispiel' ('for-the example', *for instance*). It also includes PCs centre-embedded in NCs like the one in figure 3.1. Furthermore, appositions are annotated. Neither post-modifying PPs nor post-modifying NPs are annotated. The parser thus annotates slightly fewer structures than are annotated by our parser.

3.2.8 Becker and Frank (2002)

Like Schmid and Schulte im Walde (2000), Becker and Frank (2002) use PCFGs for annotation. However, while Schmid and Schulte im Walde (2000) simply annotate noun chunks, Becker and Frank (2002) simply annotate topological fields. Furthermore, the PCFG they use is not head-lexicalized but a standard PCFG model. For the training of the PCFG, the NEGRA corpus (Brants, Skut, and Krenn, 1997) is used. In contrast to the corpus used by Schmid and Schulte im Walde (2000), the NEGRA corpus is a corpus which was annotated over a period of some years. It contains (among other information) syntactic constituent structures in the form of syntactic trees. Since this corpus does not contain any topological fields, topological fields have to be inferred "by defining linguistically informed conversion rules". After conversion, only the test section of the corpus is manually corrected. The conversion procedure yields 93.0% labelled precision and 93.7% labelled recall. The topological field structure of the training corpus is, thus, not without errors.

Becker and Frank (2002) do not use a H-L PCFG because lexical information is less important for the annotation of topological fields than for the annotation of, for example, PP attachment. This is the reason why we also do not use any lexical information for the annotation of topological fields. In fact, PoS tag information alone suffices for this task. Becker and Frank (2002) test the effect of four variants of the PCFG on performance. First, categories are parameterised. This means that categories are subdivided into a more fine-grained scheme such that they are virtually contextualized. One example is the category CL which is subdivided into, for example, the subcategories CL-REL and CL-V2 because the former is a verb-last and the latter is a verb-second clause. Both the context and the structure of those two subcategories is different.

Furthermore, Becker and Frank (2002) test how the PCFG can deal with the training material without punctuation. This is done because topological field parsing can typically rely on the strict punctuation rules of German. This is advantageous if the text type to be dealt with is quite formal. However, in more informal contexts, punctuation is not a reliable clue for parsing. This is even more the case in spoken communication where it is totally missing. Hence, Becker and Frank (2002) test how a PCFG would perform without this information.

The third tested strategy is binarization. Topological fields would typically expand to very heterogeneous sequences of PoS tags since constituent order in German is quite free. Consequently, a stochastic parser would run into sparse data problems. By binarizing topological field structure, this problem could be overcome. The fourth strategy used by Becker and Frank (2002) is pruning. Since the training material used for the PCFG is converted automatically, it may contain rare structures which might be the result of conversion errors. Thus, structures occurring just once are deleted from the grammar. This also has the effect that the grammar becomes smaller.

3.2.9 Klatt (2004a,b)

Klatt (2004a,b) annotates chunks (which he calls *minimal phrases*) and topological fields. Like in our approach, the definition of chunks includes centre-embedding and excludes post-head modifiers. Klatt (2004a,b) distinguishes between NP chunks and DP chunks: Basically, NP chunks are non-recursive kernel phrases not headed by a determiner and DP chunks are determiners followed by an NP chunk. Topological fields are also annotated but recursion in topological fields is not covered. Before parsing, text is tokenized and PoS-tagged as described in Klatt (2002). Klatt (2002) uses a rule-based approach which makes use of morpho-syntactic constraints followed by a stochastic approach (Schmid, 1999). The skeletal structure of the PoS tagset used is the STTS but there are some more fine-grained distinctions and also some modifications to the tagset. Topological fields (without recursion) are annotated first and chunks afterwards.

The annotation method used in Klatt (2004a,b) is a deterministic transition network. It is not clear, however, how powerful this transition network is exactly. By contrast to most approaches, input is not strictly processed linearly (i.e. from left to right or vice versa) but the easiest decisions are made first. The mechanism allows a bi-directional search from every token position in the input text. It is also possible to test context, i.e. a certain transition network is only triggered off in a certain context. Furthermore, the parser has special transition networks for certain tokens. These are typically applied before the more general rules. Apart from the extended STTS tags and token information, the parser makes use of corpus-based tests in order to recognize multi-word units. This means that, in certain contexts, a corpus look-up is made to check whether a certain sequence might be a multi-word unit.

3.2.10 Brants (1999)

Brants (1999) presents an approach which uses Markov Models (MMs) to annotate chunks. The MMs are trained on treebanks. Thus, this approach is very different to ours. However, Brants (1999) also uses a cascaded architecture like we do; and he explicitly refers to cascaded FSAs as a source of inspiration for cascaded MMs. MMs are best suited to dealing with a tagging task, i.e. with the task of labelling tokens and not with the task of assigning structure. In order to assign structure, MMs have to be cascaded. Hence, cascaded MMs are an extension to the PoS tagging task. This extension makes MMs capable of assigning structure beyond the level of PoS tags. This is done by making the output of one MM level the input of the following MM level. Thus, each level is represented by its own MM.

The crucial difference between MMs for tagging and MMs for parsing is that, in the latter, a sequence of terminals may be replaced by **one** non-terminal, i.e. a phrasal category. Complex phrasal structures are created by the cascaded architecture. Since it is possible that a structure generated on a lower level does not fit into a structure at a higher level, more than one potential structure is passed to the following level if it is above a defined threshold. The following level can then select the best solution with respect to the structures to be generated at this level. The number of levels in the cascade is fixed just as it would be fixed in an FSA cascade. However, a fact which we also exploit is important to mention: Recursion in language is unrestricted in theory only. In the NEGRA corpus, which is used by Brants (1999), the average number of layers is about 5, and 99.9% of all sentences have 10 or less layers.

In fact, after the application of 7 layers of annotation, both precision and recall do not change significantly any more in the experiments carried out in Brants (1999). The F-score has reached its maximum on this level. In this context, it has however, to be mentioned that the NEGRA treebank has a comparatively shallow annotation structure. A PP, for instance, consists of just one layer. Consequently, the parser which is trained on this treebank can work with comparatively few levels. It would be interesting to find out whether this type of parser would also work on a treebank with a more complex annotation like, e.g., the TüBa-D/Z. In Brants (1999), only chunks are annotated. Centre-embedded structures are included and post-nominal modification like PP-attachment and genitive modification are excluded. Clause structure is not annotated and, since topological fields are not annotated in the NEGRA training corpus, topological fields are not annotated, either.

3.2.11 Summary

With respect to the structures which are annotated, Braun (1999); Neumann, Braun, and Piskorski (2000) and Neumann and Piskorski (2002), and Wauschkuhn (1996) come closest to the approach at hand because both approaches annotate chunks, topological fields and clauses. However, since there is no qualitative evaluation in Wauschkuhn (1996), it is mainly the pioneering work in annotating topological fields first and other structures later which is the major achievement of Wauschkuhn (1996). It is, unfortunately, impossible to say how well this approach performs in real life applications. With respect to the method being used, three approaches come close to ours: First, Braun (1999); Neumann, Braun, and Piskorski (2000) and Neumann and Piskorski (2002), second, Schiehlen (2002, 2003a,b), and, third, Kermes and Evert (2002, 2003). In the first two approaches, it does not become clear, however, to what degree they are still within the finite-state formalism when they apply agreement checking.

Schmid and Schulte im Walde (2000), Becker and Frank (2002), and Brants (1999) show methods outside the finite-state formalism to annotate shallow structures. While Schmid and Schulte im Walde (2000) need an existing grammar in order to train it on an unannotated corpus both Becker and Frank (2002), and Brants (1999) have to rely on an annotated corpus. We have presented those approaches as alternative approaches to ours, and we will show how they qualitatively compare to our approach in chapter 12.

3.3 Grammatical Functions Annotation

3.3.1 Finite-State Approaches and GF Annotation

There are even fewer approaches for the annotation of GFs in German than there are for shallow parsing of German, and there are even fewer which use FSAs. In fact, with respect to the annotation of GFs using FSAs, there are only a few approaches for all languages, in general, although this strategy has already been suggested in Abney (1991). Aït-Mokhtar and Chanod (1997a,b) and Aït-Mokhtar, Chanod, and Roux (2002) are one of the few exceptions. They have constructed an incremental parser. This parser first annotates shallow structures like chunks and clauses and then recognizes GFs on top of this structure. However, the situation is very different for French since, in French, the constituent order is typically subject, verb, object (SVO) whereas, in German, constituent order can vary significantly. Furthermore, Aït-Mokhtar and Chanod (1997b) simply evaluate the annotation of subject and objects. The reason for this can also be seen in the nature of GFs in the French language. In our approach for German, we evaluate the annotation of seven GFs. And, finally, although Aït-Mokhtar, Chanod, and Roux (2002) has to be seen as the successor of Aït-Mokhtar and Chanod (1997a,b), it is no longer within the FS formalism. It "can handle rich and fine-grained lexical and dependency descriptions via feature lists." (Cf. Aït-Mokhtar, Chanod, and Roux, 2002, p.131.)

Another approach using FSAs to annotate GFs is introduced in Oflazer (1999) and Oflazer (2003). In this approach, GFs (among other relations) are annotated as dependencies. However, the task is very different from the one for German since Turkish is an agglutinative language and GFs are more clearly expressed by morphological features than is the case in German. Furthermore, except for subject and object, GFs in Turkish and German cannot be directly compared. Oflazer (1999, 2003) presents an "all-parses approach", i.e., the approach is non-deterministic. If it is possible to assign more than one structure, all the respective structures are assigned. The evaluation in Oflazer (2003) is rather modest. Only 200 sentences are contained in the test set, 30 of which are already within the development set.

3.3.2 Schiehlen (2003a,b)

The approach in Schiehlen (2003a,b) comes quite close to our approach since it also uses cascades of FSTs (cf. section 3.2.4) and it also incrementally annotates shallow structures first and GFs on top of those structures. While in our approach, GFs are encoded as labels of constituents, they are encoded as dependencies in Schiehlen (2003a,b). The essential distinction between Schiehlen (2003a,b) and our approach, however, can be seen in the treatment of ambiguity in GF annotation. Since there is often more than one possible GF structure which can be assigned, ambiguity arises. Schiehlen (2003a,b) tackles this problem by leaving GF structure underspecified. Thus, in the example sentence 'Peter kennt Karl', in which either 'Peter' is the subject and ‘Karl’ is the object or vice versa,[4] both possibilities are assigned.

To disambiguate these underspecified structures, Schiehlen (2003a,b) applies a constraint grammar-based technique which uses context variables. In this technique, each ambiguous sequence receives a specific name, and the respective ambiguous dependencies for this sequence receive context variables indicating specific linguistic context features. Then, constraints are applied to the context variables. This technique, which is applied in a module following the cascade of FSTs, is no longer within the FS formalism. This disambiguation mechanism is the most crucial difference between the parser in the thesis at hand and the parser in Schiehlen (2003a,b). It is one of our main goals to show that it is possible to deal with ambiguities in GF annotation **within** the FS formalism.

3.3.3 Trushkina (2004)

The parser described in Trushkina (2004) has already been discussed in section 3.2.6 as far as the annotation of shallow parsing structures and morpho-syntactic disambiguation is concerned. After this, GFs are annotated as dependencies on top of shallow parsing structures. The annotation of GFs, then, mainly relies on the morphological information which is disambiguated **before** shallow parsing (as described in section 3.2.6), the shallow parsing structure and on linear order. The reason for this is that Trushkina (2004) wants to show that it is possible to annotate GFs mainly without lexical information. A clear disadvantage of this strategy is that – provided that morphological disambiguation succeeds – only those GFs can be annotated which are connected with case. This means that, for example, the important group of prepositional objects cannot be annotated at all. And, is, in fact, not annotated in Trushkina (2004). Reflexives which have no overt case difference, like ‘sich’ which may be dative or accusative cannot be distinguished, either. Furthermore, a lot of predicatives (e.g. adverbs) cannot be annotated simply on the grounds of their morphological features because they simply do not have any.

And, in fact, Trushkina (2004) uses a lexicon for the annotation of GFs. This lexicon contains some 150 high-frequency verbs and is compiled based on training data. This is somewhat in contradiction to the intention to show that GF annotation can do without a lexicon especially if one considers that

[4] The sentence may, hence, mean either ‘Peter knows Karl’ or ‘Karl knows Peter’.

the 150 high-frequency verb **types** may cover a considerable amount of verb **tokens**. In our test section, there are 4,951 lexical verb tokens, which amount to 1,475 verb types. There are no such figures given at all for the test section in Trushkina (2004); but, for our test section, 150 verb types would, thus, be roughly 10% of all verb types.

However, with *Zipf's Law* in mind, one has to consider that highly-ranked words occur far more frequently than low ranked ones. Since Trushkina (2004) does not give frequency figures and since Zipf's Law is an experimental law and not a theoretical one, one cannot exactly estimate whether the high-frequency verb types in Trushkina (2004) comprise about 30%, 40% or even 50% of the verb tokens in the test section. The fact that not just the highest-ranked verb types have been chosen for the verb subcategorization lexicon but certain verb types which cause special trouble for the parser intensify the impression that annotating GFs without a lexicon is a rather hard task.

Dependencies representing GFs are annotated incrementally, i.e. dependencies which are annotated can make use of already annotated dependencies. Consequently, an *easy-first parsing* strategy is applied in which those dependencies which can be annotated most reliably are annotated first. In the case at hand, those dependencies are typically annotated first which occur more often and/or are more crucial for dependency structure, i.e., subjects are annotated before direct objects before indirect objects.

The disambiguation mechanism is constraint-based. First, dependencies are established on the basis of morphological features and shallow parsing structures. Then, dependencies can be eliminated or renamed according to the specified constraints. "Constraints are stated on the context, on the internal structure of the nodes and on features of lexical nodes. Moreover, constraints on previously established relations are imposed." (p. 147) This would mean, for example, that a dependency of the type 'direct object' is assigned between a noun in an NP in the VF if it has the feature 'accusative', if the NP is not introduced by 'als',[5] and if no other such dependency has been assigned yet.

3.3.4 Schmid and Schulte im Walde (2000)

We have already introduced the approach described in Schmid and Schulte im Walde (2000) (cf. section 3.2.7) and stated in which way it differs from ours.

[5]In which case it would more likely be a modifier of the direct object.

Schmid and Schulte im Walde (2000) annotate shallow structures (chunks) as well as GFs. However, this is done simultaneously. In Schmid and Schulte im Walde (2000), text is first annotated with PoS tags which also include morphological (including case) information. Afterwards, structures are disambiguated using a H-L PCFG. It thus makes sense to simultaneously annotate shallow annotation and GFs, at least those which involve case information. Schmid and Schulte im Walde (2000) mark the respective chunks with their case information using the H-L PCFG constructed and trained as described in section 3.2.7. However, there remains the problem that, once the case of an NC is assigned, it is still not clear to which clause it belongs and to which head (mostly verb) it refers; but Schmid and Schulte im Walde (2000) do not report any details of the annotation of clauses in their parser or of their respective dependencies.

3.3.5 Kübler (2004a,b)

The parser described in Kübler (2004a,b) uses a method to annotate GFs which is very different from ours. Still, it is included here because it is one of the few which annotates GFs for German. Kübler (2004a,b) uses a memory-based learning (MBL) approach. Like other learning approaches, MBL also requires training data in order to learn how to annotate unknown language data. However, MBL differs greatly from the learning approaches presented above (e.g. PCFGs) since it is part of the *lazy learning* paradigm: The learning approaches presented above (*eager learning* approaches) process training data and abstract them into rules and/or probabilities, i.e., the actual training data are only indirectly involved in annotation (via the abstraction process).

This is different in MBL. In MBL, training instances are stored in an instance base without abstraction. The assumption in MBL is that instances which are similar belong to the same class. The parser, hence, compares the input to the instances in the instance base and assigns the input the category of the most similar instance. However, since it cannot be assumed that there are instances in the instance base which are totally similar to the input, a similarity metric has to be defined according to which the input data can be compared to the data in the instance base. This similarity metric is, in fact, the only abstraction inherent to MBL; and the success of the parser crucially depends on the definition of this metric. The features of the input and of the respective instance are compared by the similarity metric and can be weighted.

The MBL parser uses PoS-tagged and chunked text as its input. PoS-tags are assigned according to the STTS tagset using the TnT tagger described in Brants (2000). Chunking is achieved by the CASS parser which is a cascaded deterministic finite-state parser (cf. Abney, 1996c). The MBL parser then first tries to find the most similar sentence in the instance base on the basis of word order and PoS tag order in the input sentence. If this step does not succeed, the sentence is handed over to a backing-off module which mainly uses chunk information. This backing-off module also deals with structures which could not be annotated on the basis of word and PoS tag order information.

It is important to mention for the approach in Kübler (2004a,b) that it does **not** annotate GFs on top of chunks but that it uses the chunk information as additional information for the selection of the most similar sentence. In fact, chunk structure and phrase structure may differ considerably. However, the information about, for example, the boundaries of chunks is helpful for the determination of the most appropriate tree to be assigned. If, for instance, a tree structure clashes with a chunk boundary, i.e., the tree to be assigned includes only a part of the chunk, then the parser tries to assign a shorter tree (cf. Kübler, 2004b, p.75).

3.3.6 Summary

There are just a few approaches for the annotation of GFs in German. Only the approach presented in Schiehlen (2003a,b) comes relatively close to ours since it also uses FSTs for the incremental annotation of shallow structures and annotates GFs (as dependencies) on top of those structures. However, in order to resolve ambiguous GF structures, Schiehlen (2003a,b) applies a constraint grammar-based technique which uses context variables and is, thus, outside the FS formalism. Trushkina (2004) also annotates GFs as dependencies on top of shallow structures. However, annotation is mainly restricted to GFs connected with case. A constraint-based formalism is used for disambiguation.

While Schiehlen (2003a,b) and Trushkina (2004) largely rely on handwritten grammars, Schmid and Schulte im Walde (2000) and Kübler (2004a,b) mainly work with training algorithms. Schmid and Schulte im Walde (2000) use a H-L PCFG to annotate chunks including morphological information which show their GF. This type of learning may be called *eager learning*.

It uses language data to extract rules and/or probabilities from them (in this case mainly probabilities because the CFG already existed before). The other type of learning, which is applied in Kübler (2004a,b) as an MBL, is called *lazy learning*. Kübler (2004a,b) stores instances from the training corpus without abstraction in an instance base. These instances are compared to input to the parser via a similarity metric such that the correct structure can be assigned.

All four approaches presented here can be seen as hybrid approaches since none of them is fully restricted to either learning algorithms or hand-written grammars. Schiehlen (2003b) investigates a combination of the FST parser with an n-gram-based parser in order to fully disambiguate non-deterministic output of his FST parser. Trushkina (2004) uses a PCFG-based morpho-syntactic tagger for the disambiguation of morphological features which are the basis of GF annotation in her approach. Schmid and Schulte im Walde (2000) is also a hybrid approach since the H-L PCFG uses a hand-written grammar as its basis; and, finally, Kübler (2004a,b) uses a hand-written chunk grammar (CASS) as one source of information for her MBL parser.

With respect to the GFs which are annotated, there are differences from the four approaches presented here. Schmid and Schulte im Walde (2000) is fully restricted to GFs connected with the four cases in German. Trushkina (2004) also annotates sentential objects as GFs apart from those GFs connected with case.[6] Kübler (2004a,b) annotates all the GFs annotated in the TüBa-D/Z since she uses the TüBa-D/S as training material which uses the same annotation formalism as the TüBa-D/Z. Thus, apart from case-related GFs, prepositional objects, sentential objects, modifiers of phrases, heads of phrases, and appositions are annotated. The same range of GFs is annotated in Schiehlen (2003a,b) although dependencies may have slightly different names and definitions due to the different formalism used.

[6]There are also dependencies annotated within the verb complex, which we do define as GFs.

Part I

Annotating Shallow Structures

Chapter 4

Shallow Annotation Structures

Abstract: Chapter 4 introduces shallow parsing which is the main issue of part I. Section 4.1 defines the concept of shallow parsing and explains why there is a need for shallow parsing, at all. Shallow structures are defined as those structures which can be annotated reliably, efficiently and without the use of large knowledge bases. Furthermore, it is important that further annotation can be built on top of shallow structures. We show that chunks, topological fields and clauses fulfill the requirements for shallow structures and give detailed definitions of them in section 4.2. Furthermore, we explain the general concept of the label set of chunks, topological fields and clauses, and we discuss specific problems like coordination in noun chunks.

4.1 The Nature and Right-of-Existence of Shallow Parsing

An annotation structure can be called 'shallow' in the broadest sense if it only partially disambiguates the syntactic structure of a sentence, i.e. if it does not convey all the information which is contained in a full annotation structure. In contrast to what the word 'shallow' might suggest, a shallow structure does not merely contain constituents which are close to pre-terminal nodes although this might have been the original meaning. A shallow structure simply contains less syntactic information than a full annotation structure. Thus, it lacks, e.g., the information about what kind of function a noun phrase or a prepositional phrase has. Because the annotation does not contain all information, this kind of parsing is sometimes referred to as *partial* *parsing* (cf. Abney, 1996c). This term is a lot more concise than *shallow parsing* since it is neutral about what kind of information is missing in the annotation. However, as the term *shallow parsing* is more established in the literature, we will stick to it in the dissertation at hand.

Since, for linguistic reasons, a fully annotated structure would always be preferable to a partially annotated one, there remains the question of the right of existence of shallow parsing. To a large extent, this right of existence is justified by technical arguments and depends, thus, on the definition of the particular developer. Consequently, in general, there is no clear dividing line between shallow and deep annotation. If shallow annotation is contrasted with full annotation, then it could, in the extreme case, even be defined as reaching one step before full annotation. There are, however, some common features which are shared by most shallow parsing approaches: Since most approaches use shallow parsing as a basis for full parsing, the shallow structures are defined in a way which makes it possible to straightforwardly build full parsing structures on the shallow ones. Furthermore, most shallow parsing approaches subsume those structures which can be annotated reliably at a considerable speed without the use of large data bases (cf. the list in figure 4.1). Shallow parsing, thus, follows the concept of *easy-first parsing* described in Abney (1996c), in which indisputable structures are annotated first thus achieving growing *islands of certainty* which allow the annotation of further structure.

In other words, or defined negatively, a shallow annotation structure subsumes only those structures which can be annotated without problems; or as defined by Abney (1995) for *chunking* in the broader sense (i.e. in the sense of shallow parsing):

> *We can define* `chunks` *as the parse tree fragments that are left intact after we have unattached problematic elements.* (Abney, 1995)

For our system, we use a definition of shallow parsing which includes both a technical and a theoretical perspective, which are interdependent. From a theoretical perspective, we define shallow structures as those structures which are subject to syntactic restrictions rather than subject to lexical selection; from a technical perspective, we define shallow structures as those structures which can be captured simply using the part-of-speech (PoS) in-

→ useful basis for full annotation

→ high precision of annotation

→ no large knowledge bases required

→ computationally inexpensive algorithm

Figure 4.1: General Features of Shallow Annotation Structure

formation[1] of the tokens in a regular expression (RE) grammar, which can be implemented using FSAs. Grammatical phenomena which need lexical information for their annotation are not included in the shallow structures. Thus, the dividing line between shallow and deep structures is the fact whether, in theoretical terms, a phenomenon is subject to lexical selection, or, in technical terms, whether it can be annotated with or without lexical information. Consequently, according to this definition, there is a quite clear dividing line between shallow and deep structures since they belong to different levels of syntactic description. This has already been described in other words by Abney (1991) for chunks:

> *Chunks also represent a grammatical watershed of sorts. The typical chunk consist of a single content word surrounded by a constellation of function words, matching a fixed template. A simple context-free grammar is quite adequate to describe the structure of chunks. By contrast, the relationships* ***between*** *chunks are mediated more by lexical selection than by rigid templates.* (Abney, 1991, author's emphasis)

One of the first and the most prominent of all shallow annotation approaches is called *chunking*. Abney (1996a) gives a definition in his chunk stylebook: 'Roughly speaking, a *chunk* is the non-recursive core of an intraclausal constituent, extending from the beginning of the constituent to its head, but not including post-head dependents.' Furthermore, Abney (1991) also gives psychological evidence for the existence of chunks when quoting Gee and Grosjean (1983) who have made experiments with pause durations in

[1]We use the Stuttgart-Tübingen tagset (STTS) described in Schiller, Teufel, and Thielen (1995). Cf. appendix B.

reading and naive sentence diagramming. These experiments have revealed a structure which they call performance structure. Abney (1991) shows that the constituents in this structure are similar to chunks.

In Abney (1991), it is also explicitly made clear that chunks do not include pre-head-modifiers if this would lead to recursion. The chunk structure in example 19 would, thus, consist of two nominal chunks and a 'pending' preposition. Chunks can, consequently, be described as **non-recursive**, **continuous core** phrases. There is, however, also a wider definition of the term *chunking*, which has already been quoted above from Abney (1995) and which we regard as synonymous with the term *shallow parsing*, namely that chunks can be seen 'as the parse tree fragments that are left intact after we have unattached problematic elements' (Abney, 1995). Sometimes, if the term *chunk* is used, it is this wider definition which is referred to. We will, however, stick to the narrow definition of chunks as core phrases if we speak of chunks in this dissertation.

(19) in [$_{\text{NC}}$ Carl's] [$_{\text{NC}}$ lovely summer cottage]

Shallow structures have first been used for English. They suit the syntactic structure of the English language very well because, in English, syntactic dependency is expressed by distributional proximity, in most cases. This is, however, not always the case in German. On the contrary, syntactic dependency can, in German, sometimes even be expressed by distributional distance. The German verb complex is the most striking example: Unlike in English, where only in inverted clauses (e.g. questions) any considerable structure can intervene between the finite verb and the rest of the verb complex, in German, in any affirmative clause the verb complex may be divided by a considerable number of constituents, since the finite verb comes second and the rest of the verb complex comes at the end of the sentence (cf. sentence 20). This difference in the syntactic structure of English and German has an effect on the definition of shallow parsing structures for German in contrast to those in English.

(20)	Die	Wirtschaftsförderausschüsse	Bremens	[$_{\text{verb-complex}}$ haben]
	The	economy promotion committees	of Bremen	have

gestern	der	Errichtung	der	International University Bremen in
yesterday	the	foundation	of the	International University Bremen in

Grohn $[_{\text{verb-complex}}$ zugestimmt].
Grohn affirmed.

'(The state of) Bremen's committees for the promotion of the economy have affirmed the foundation of the International University Bremen in Grohn, yesterday.'

On the level of chunks, one of the most obvious problems in German is center embedding in noun phrases: center embedding occurs if a constituent which is embedded in a constituent of the same type is surrounded by parts of the embedding constituent. The German examples 21 and 22 illustrate the deficiencies of the chunk definition for German as regards center-embedding: If chunks are defined as non-recursive like in English, then prepositional phrases modifying an adjective phrase which is part of a noun phrase cannot be included in the adjective chunk because this would violate the definition of chunks with respect to recursive structures. Hence, following the definition of chunks, a structure like the one in example 21 would be generated.

(21) mit den $[_{\text{PC}}$ von $[_{\text{NC}}$ den USA]] $[_{\text{NC}}$ $[_{\text{AJAC}}$ vorgeschlagenen] Strategien]
with the of the USA suggested strategies

'with the strategies suggested by the USA'

(22) $[_{\text{PP}}$ mit $[_{\text{NP}}$ den $[_{\text{AP}}$ $[_{\text{PP}}$ von $[_{\text{NP}}$ den USA]] vorgeschlagenen]
with the of the USA suggested
Strategien]]
strategies

'with the strategies suggested by the USA'

This structure is, however, inadequate because we also define shallow structures as the prerequisite for further annotation. There is, however, no straightforward way to generate the finally desired constituent structure, which should roughly correspond to the one given in example 22, from the one in example 21. Thus, the structure in example 21 is not an adequate shallow structure. The definition of shallow structures with respect to the annotation of phrases has, therefore, to be redefined such that center embedding is allowed for German. The definition of nominal chunks should, thus, be that it includes the whole phrase from the determiner to the head word just excluding post-modification structures because on top of this structure, further annotation can be built.

clause type	topological fields						
VL:	KOORD			**C**	MF	**VC**	NF
V1:	KOORD	LV		**LK**	MF	VC	NF
V2:	KOORD/ PARORD	LV	**VF**	**LK**	MF	VC	NF
				left part of bracket		right part of bracket	

V1: verb-first clause	VF: initial field
V2: verb-second clause	MF: middle field
VL: verb-last clause	NF: final field
KOORD: coordinator field	LK: left part of bracket
PARORD: non-coordinator field	C: complementizer field
LV: resumptive construction	VC: verb complex

Figure 4.2: The topological field model

A shallow structure for German can, thus, not be defined along the same lines as for English because it is not possible to use such a structure straightforwardly for further annotation.[2] Consequently, we have to weaken the definition of chunks as **non-recursive**, **continuous core** phrases with respect to recursion and allow recursion in chunks if center-embedding occurs. This is still in line with the part of the chunk definition which defines a chunk as 'extending from the beginning of the constituent to its head, but not including post-head dependents.'

With respect to the process of annotation, shallow structures can be defined more generally as those structures which can be annotated with high precision without resorting to large knowledge bases and/or computationally expensive algorithms as it is laid out in the characteristics listed in figure 4.1. These more general characteristics also fit the definition of chunks but they also allow the definition of other constituents than chunks as shallow structures. Topological fields are such a linguistic phenomenon, which fulfills the requirements for shallow structures. The categorization of German clause

[2]This would even more be the case for languages in which syntactic dependency is even less reflected by distributional proximity or word order, e.g. the Slavic languages.

types and the respective word order phenomena in terms of topological fields has a long tradition in the empirical investigation of German syntax (cf. Herling, 1821; Erdmann, 1886; Drach, 1937; Reis, 1980; Höhle, 1986) and is by now widely accepted as a classification of German clauses and their internal structure. Although topological fields are located on a higher level in the parse tree, they still adhere to very strict syntactic restrictions. Shallow parsing for German does, thus, not only allow the annotation of chunks but also the annotation of the macro structure of the German sentence.

Topological fields describe sections in the German sentence with regard to the distributional properties of the verb (and the complementizer in subordinate clauses). Figure 4.2 shows that there are three different types of clauses in German, which can be defined according to the position of the finite verb in them: verb-last clauses (VL), verb-first clauses (V1) and verb-second clauses (V2). VL clauses comprise all introduced subordinate clauses, V1 clauses mainly comprise imperatives and yes/no questions and V2 clauses mostly comprise affirmative clauses and wh-questions. The topological fields *C-Feld* (C; complementizer field) and *Linke Satz-Klammer* (LK, left part of sentence bracket), on the one hand, and *Verbkomplex* (VC; verb complex), on the other hand, constitute the sentence bracket, relative to which the other fields can be described.[3] In V1 and V2 clauses the field LK contains the finite verb and the field VC all other verbal elements; in VL clauses, the field C contains the complementizer and the field VC all verbal elements.

The section preceding the left part of the sentence bracket is called the *Vorfeld* (VF; initial field; only in V2 clauses), the section included in the sentence bracket is called the *Mittelfeld* (MF; middle field) and the section following the right part of the sentence bracket is called the *Nachfeld* (NF; final field). There may also be a *Koordinationsfeld* (KOORD; coordinator field), which contains a coordinating particle, or an alternative field to the KOORD, the field PARORD, which only occurs in V2 clauses and contains non-coordinating particles like 'denn'. Additionally, there may be a Linksversetzungsfeld (LV; resumptive construction). The terminology is analogous to the one in the TüBa-D/Z (cf. Telljohann, Hinrichs, and Kübler, 2003), which we use as a gold-standard for the evaluation (cf. section 10.1.2 and appendix C). Figures 4.3–4.5 give some (constructed) examples to illustrate the structure of the topological fields.

Figure 4.6 shows a sentence taken from the TüBa-D/Z. It shows an ex-

[3]Obligatory fields are in bold type.

		KOORD	C	MF	VC	NF
1.	Ich glaube, I think		daß that	der Plan verrückt the plan mad	ist. is.	
2.		Und And	wenn if	das einer that anyone	merkt? notices?	
3.	Das ist der Mann, This is the man		der who	den Wagen the car	gestohlen hat. stolen has.	
4.	Er fragt, He asks		ob whether	das in Ordnung that in order	ist is	mit dem Geld. with the money.

1. *I think that the plan is mad.*
2. *And if anyone notices that?*
3. *This is the man who has stolen the car.*
4. *He is asking whether it is okay with the money.*

Figure 4.3: Examples of topological fields in VL clauses

	KOORD	LV	LK	MF	VC	NF
1.	Und And		kann can	er das he that	bezahlen? pay for?	
2.		Den Preis, The prize,	kann can	man den one it	behalten? keep?	
3.			Wollt Want	ihr Ärger you trouble	haben have	mit mir? with me?
4.	Aber But		geh go	mir sofort aus der Sonne! me immediately out of the sun!		

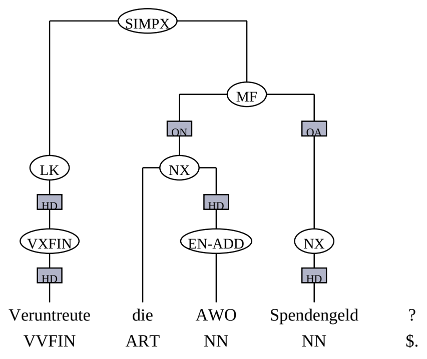
1. *And is he able to pay for it?*
2. *Can we actually keep the prize?*
3. *Do you want trouble with me?*
4. *But get out of the sun, immediately!*

Figure 4.4: Examples of topological fields in V1 clauses

	KOORD	LV	VF	LK	MF	VC	NF
1.		Der Sieger, The winner,	der he	kann can	sich himself	freuen rejoice	heute abend. tonight.
2.	Aber But		kein Mensch no person	konnte could	sie daran her from that	hindern, keep	dorthin zu gehen. there to go.
3.			Ich I	weiß, know			daß es falsch war. that it was wrong.
4.			Jetzt Now	muß must	das Regelwerk the rules	geändert werden. changed be.	
5.			Wer Who	steht stands	denn im Mittelpunkt? actually in the centre?		

1. *The winner can be happy, tonight.*
2. *Nobody could keep her from going there.*
3. *I know that it was wrong.*
4. *Now, the rules have to be changed.*
5. *So who's actually in the focus of attention?*

Figure 4.5: Examples of topological fields in V2 clauses



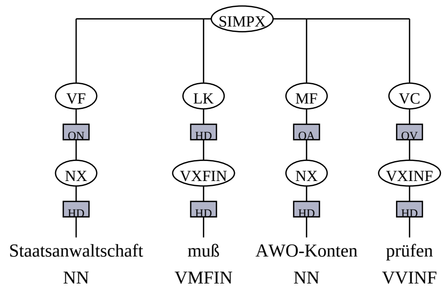
Veruntreute	die	AWO	Spendengeld?
Embezzled	the	AWO	donation money?

‘Did the AWO embezzle donated money?’

Figure 4.6: Example of a V1 clause from TüBa-D/Z

ample of a V1 clause. The finite verb ‘veruntreute’ is the first element in the clause; it is the left part of the sentence bracket. Since there is no right part of the sentence bracket, the MF extends to the end of the clause. Figure 4.7 shows an example of a V2 clause. The VF stretches from the beginning of the clause to the finite verb ‘muß’, which is the left part of the sentence bracket. The MF stretches from the finite verb to the right part of the sentence bracket which consists of the infinitive verb ‘prüfen’. Figure 4.8 gives an example of a VL clause. Since VL clauses are subordinate clauses, they are always part of a matrix clause (here a V2 clause). The VL clause is embedded into the NF of the V2 clause. The complementizer ‘daß’ constitutes the left part of the sentence bracket, which is realized as ‘C’ in VL clauses. The finite verb constitutes the right part of the sentence bracket.

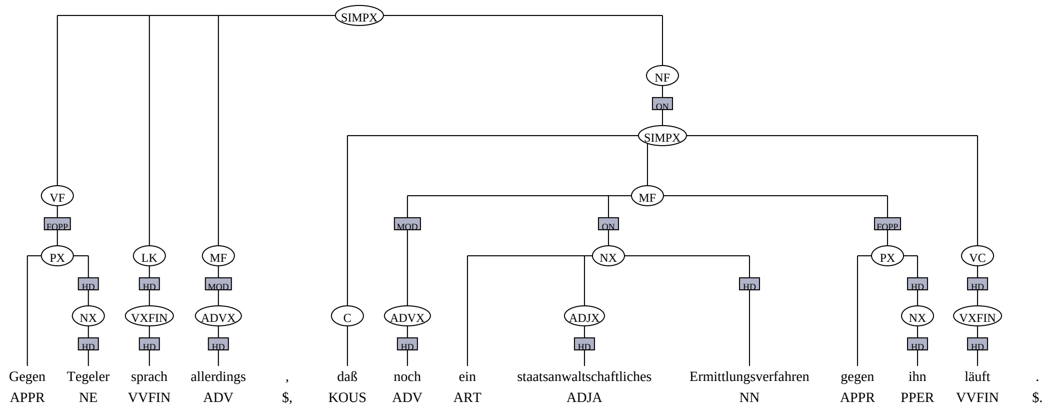
Since the topological field structure, as it is shown in figure 4.2 and illustrated in the sentences in figures 4.6–4.8, is an invariant pattern in German, it is an ideal linguistic phenomenon for shallow parsing. Figures 4.6–4.8 further illustrate that the constellation of the topological fields reveals categories and the boundaries of the clause and, thus, also allows the annotation of clauses. Recursion can also be accounted for by the topological field model because subordinate clauses may be embedded in topological fields and, with this model, it is possible to leave their attachment open until disambiguation in



Staatsanwaltschaft	muß	AWO-Konten	prüfen
Public Prosecutor's Department	must	AWO accounts	inspect

'Public Prosecutor's Department has to view AWO accounts'

Figure 4.7: Example of a V2 clause from TüBa-D/Z



Gegen	Tegeler	sprach	allerdings,	daß	noch	ein
Against	Tegeler	spoke	admittedly	that	also	a

staatsanwaltschaftliches	Ermittlungsverfahren	gegen	ihn	läuft.
Public Prosecutor	preliminary investigations	against	him	runs.

'Admittedly, it speaks against Tegeler, that there are also preliminary investigations of the Public Prosecutor made against him.'

Figure 4.8: Example of a VL clause from TüBa-D/Z

further annotation steps.

The model of topological fields is primarily a distributional model. It does not give any account of the grammatical function structure and it does not reveal the relations between the constituents within the topological fields, either. In fact, the very structure of constituents **within** topological fields is left open in this theory. The model still has some clear advantages from both a theoretical and an annotational perspective: As regards the theoretical perspective it is, for example, important to point out that many constituent-order phenomena can be described relative to topological fields. As regards the annotation of further grammatical phenomena, the constituent order in the different topological fields may be utilized for annotation. This is possible because, although there are, in German, very few syntactic restrictions in the constituent order within the fields, there are, however, a lot of syntactic preferences which may be utilized if connected with other information like morphological features and valency structure.

As regards automatic annotation, one of the main advantages of topological fields is that they are the skeleton of the sentence and that, thus, once topological fields are annotated, clause boundaries and potential points of attachment are known. The annotation of topological fields considerably reduces the scope of ambiguity because the verb is always part of the sentence bracket and the respective grammatical functions are always realized in the corresponding fields. Without the annotation of topological fields, the scope of the grammatical functions of the verbs is much wider, especially in complex sentences, in which, additionally, it is not clear which potential functions belong to which verb. After the annotation of topological fields, the syntactic restrictions and preferences which are valid in them might be utilized for further annotation (together with other linguistic information).

(23) Man könne, wegen der Erfahrung, die Ulrich Eckhardt
One could, because of the experience which Ulrich Eckhardt
mitbringt, nicht auf das Know How des Leiters der Festspiele
brings along not with the know-how of the head of the Festspiele
GmbH im Umgang mit dem Martin-Gropius-Bau verzichten,
Ltd in handling with the Martin-Gropius building dispense
sagt Radunski.
said Radunski.

'Because of the experience which Ulrich Eckhardt brings along, one could not do without the know-how of the head of the Festspiele Ltd in handling with the Martin-Gropius building said Radunski.'

The advantages of annotating topological fields before annotating grammatical functions can be illustrated by figure 4.9, which shows the shallow structure of sentence 23 as annotated by our shallow parser (the category labels of the constituents are described in section 4.2). The syntactic tree for the sentence in figure 4.9 shows that there are is a V2 clause into which another V2 clause is embedded into which a VL clause (REL) is embedded. There are three lexical verbs. As the grammatical functions may be realized on either side of the verbs, it is by no means clear, where the potential grammatical functions of the respective verbs are. After the annotation of the topological fields, however, the scope is reduced: First of all, the annotation structure in figure 4.9 shows that 'könne . . . verzichten' is one verb complex. This is very useful for further processing because most modals and auxiliaries can also function as lexical verbs if they occur as the only verb in a clause.

Furthermore, the topological field and clause structure reveals that the potential grammatical functions of the verb 'sagen' may only be 'Radunski' and the embedded V2 clause. The potential grammatical functions of the verb 'verzichten' may only be found within the embedded V2 clause but not in the VL (REL) clause. The relative clause itself does only qualify as a noun-modifying grammatical function. The potential grammatical functions of the verb in the relative clause are restricted to the MF in that clause. This example shows that the topological field structure reduces the scope of ambiguity within the sentence by confining the search space for potential grammatical functions of verbs. The same effect can be seen in sentence 24 which is shown in figure 4.10. Basically, the sentence is divided into three parts and each of these parts can be treated separately in further annotation. It should be taken into account, however, that the annotation of field and clause structure is a shallow one like the one of the chunks. It is, for instance, left unspecified to which constituent in the matrix clause the relative clause relates. Still, the shallow structure reduces the potential attachment sites and the pre-structuring achieved with the annotation of the fields is a solid base for further annotation.

(24) Das gemeinsame liegt eher in der Rhythmik, die sich von
The common lies rather in the rhythmicity which itself from
dem, was an Funk in den Vereinigten Staaten gespielt wurde,
that what as Funk in the United States played was

durch	größere	Vielschichtigkeit	unterscheidet.
through	higher	complexity	distinguishes.

‘The common ground can rather be found in the rhythmicity which can be distinguished by its higher complexity from the funk which was played in the United States.’

The difference between shallow and deep structures can, furthermore, be illustrated by a quote from Höhle (1982), who writes the following about the constituent order in German:

> *In der Topologie (Wortstellungslehre) des Deutschen kann man grob zwei Bereiche unterscheiden: Gewisse Satzbestandteile sind strengen topologischen Regularitäten unterworfen; dies gilt besonders für verbale Elemente und ‘Konjunktionen’, aber in hohem Maße auch für die Bestandteile von Nominal-, Präpositional- und Adjektiv/Adverbialphrasen. Insofern sind die Fakten, die in diesem Bereich zu beschreiben sind, weitgehend unstrittig. Die Positionsmöglichkeiten verschiedener Elemente – etwa NPs und PPs – innerhalb des Mittelfeldes relativ zueinander scheinen dagegen einigermaßen undurchsichtig.* (Höhle, 1982, p. 75)
>
> *In the topology (science of word order) of German, two main areas can be distinguished: Certain elements of the clause are subject to strict topological regularities; this is especially true of verbal elements and ‘conjunctions’ but it is also largely true of the constituents of nominal, prepositional and adjective/adverbial phrases. Insofar, the facts which are to be described in this area are, to a large extent, indisputable. On the other hand, the possible positions of different elements – e.g. NPs and PPs –* ***within*** *the middle field in relation to each other appear somewhat opaque.* (our translation; our emphasis)

The phenomenon described by Höhle (1982) exactly defines the dividing line between shallow and deep structures. Shallow structures are exactly those structures where the ordering principles are indisputable while deep structures are those structures in which the constituent order appears opaque. Sentences 25–27 further illustrate the difference between those two

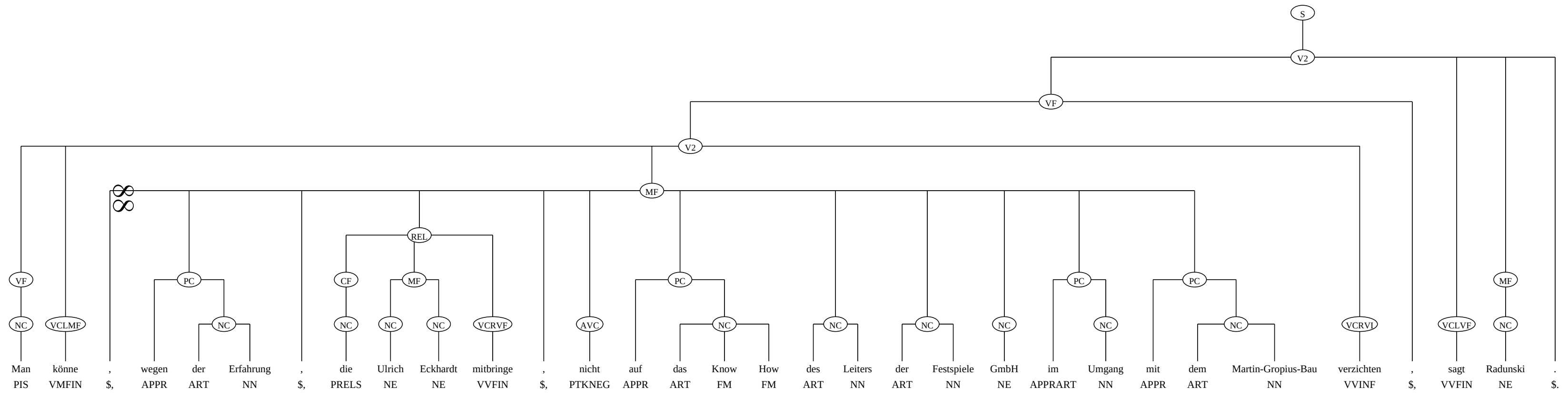


Figure 4.9: Complex sentence from the TüPP-D/Z

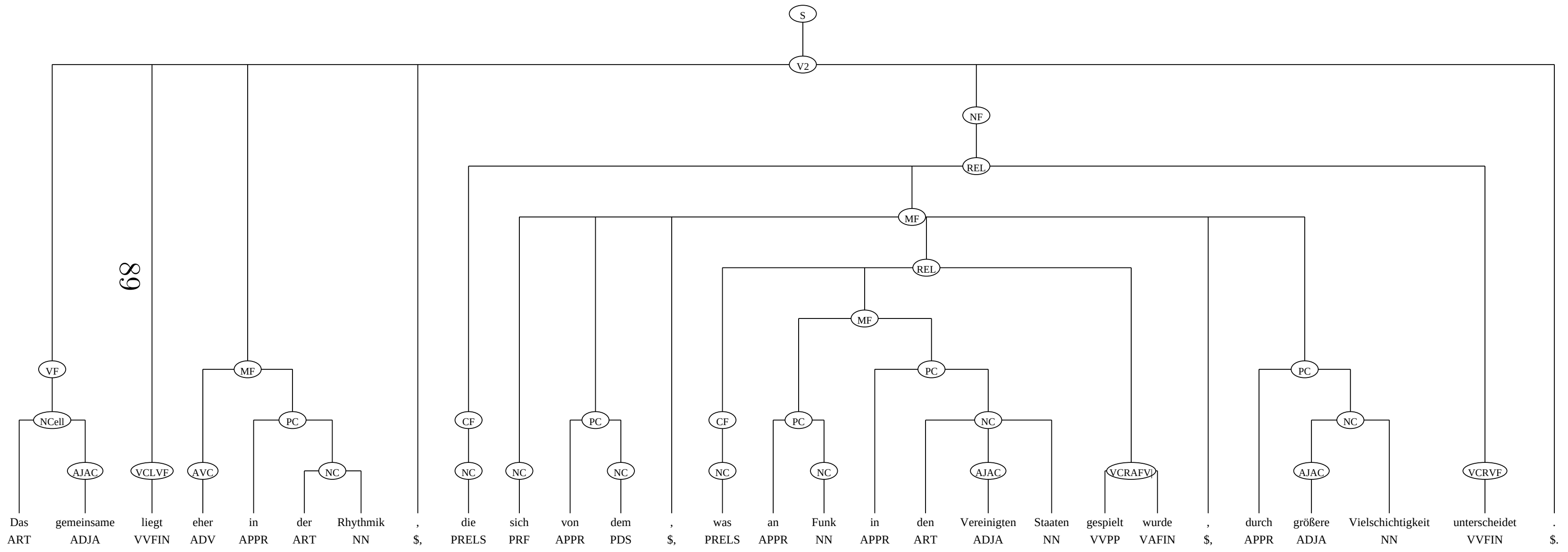


Figure 4.10: Complex sentence from the TüPP-D/Z

phenomenon areas. In all three sentences, there is the order VF, LK, MF, VC. Furthermore, the chunk structure is the same in all three sentences as there would be no other possibility for the order of constituent within the chunks. Sentence 25 is the original sentence from TüBa-D/Z (cf. figure 4.11). It is, however, possible to re-arrange the chunks in the sentence in some ways. Sentences 26 and 27 show two possibilities. While in sentence 25 the nominative object is the first element in the clause, in sentence 26, the accusative object is the first element. In sentence 27, this position is occupied by the dative object. Thus, while the structure of topological fields and chunks is syntactically restricted, the order of grammatical functions can be varied. Furthermore, the occurrence of grammatical functions is subject to lexical selection because different verbs sub-categorize for different grammatical functions.

(25) [$_{\text{VF}}$ [$_{\text{NX}}$ Die Initiatoren]] [$_{\text{LK}}$ hatten] [$_{\text{MF}}$ [$_{\text{PX}}$ vor kurzem] [$_{\text{NX}}$ den
The initiators had a short time ago the
kommunalen Spitzenverbänden] [$_{\text{NX}}$ eine massive Behinderung des
municipal central associations a massive obstruction of the
Volksbegehrens]] [$_{\text{VC}}$ vorgeworfen].
referendum accused of.

'A short time ago, the initiators had accused the municipal central associations of a massive obstruction of the referendum.'

(26) [$_{\text{VF}}$ [$_{\text{NX}}$ Den kommunalen Spitzenverbänden]] [$_{\text{LK}}$ hatten] [$_{\text{MF}}$ [$_{\text{NX}}$ die
The municipal central associations had the
Initiatoren [$_{\text{PX}}$ vor kurzem] [$_{\text{NX}}$ eine massive Behinderung des
initiators a short time ago a massive obstruction of the
Volksbegehrens]] [$_{\text{VC}}$ vorgeworfen].
referendum accused of.

'A short time ago, the initiators had accused the municipal central associations of a massive obstruction of the referendum.'

(27) [$_{\text{VF}}$ [$_{\text{NX}}$ Eine massive Behinderung des Volksbegehrens]] [$_{\text{LK}}$ hatten]
A massive obstruction of the referendum had
[$_{\text{MF}}$ [$_{\text{NX}}$ die Initiatoren [$_{\text{PX}}$ vor kurzem] [$_{\text{NX}}$ den kommunalen
the initiators a short time ago the municipal
Spitzenverbänden]] [$_{\text{VC}}$ vorgeworfen].
central associations accused of.

'A short time ago, the initiators had accused the municipal central associa-

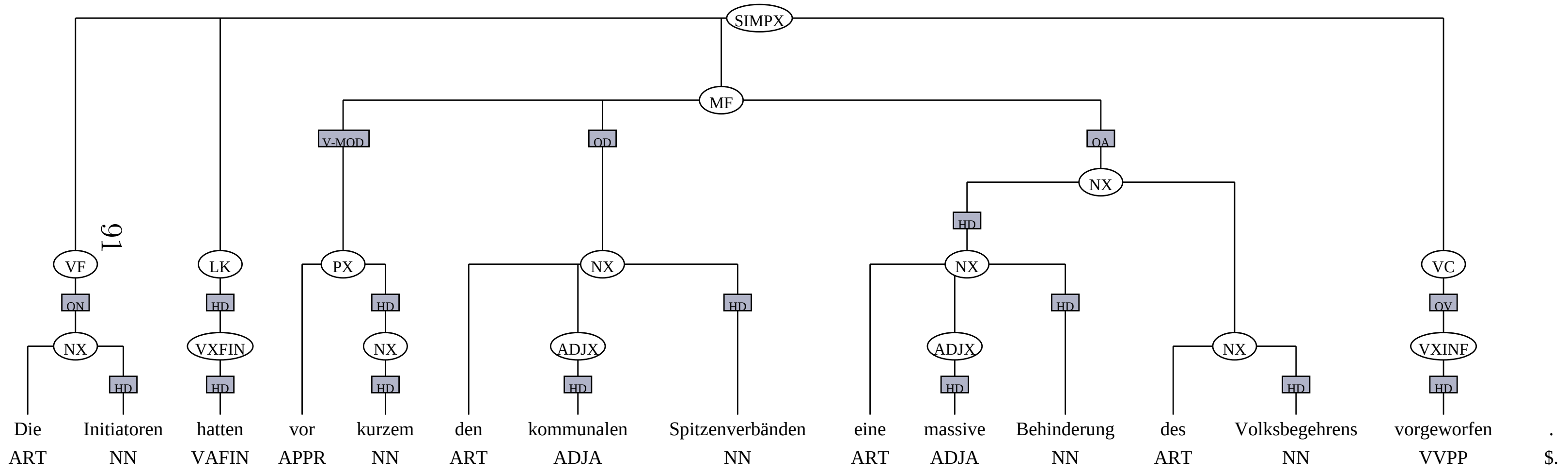


Figure 4.11: Sentence containing three GFs extracted from the TüBa-D/Z

tions of a massive obstruction of the referendum.'

In conclusion, we can state that the three linguistic phenomena chunks, topological fields and clauses can be defined as shallow structures because they comply with the criteria we have laid out for shallow annotation structures in figure 4.1. Consequently, they are an ideal candidate for the annotation by FSAs. The FSAs act as transducers which use grammars with the power of regular expressions (REs) as their transition function. Shallow annotation structures are defined as those structures which can be captured using an RE grammar which takes PoS tags as its initial input and yields annotated structures as its output. Since the annotation system is set up as a cascade of FSAs, the output of the preceding FSAs is input to the following ones and, consequently, some FSAs take already annotated syntactic structure as their input. Recursion in shallow structures like the relative clauses in figure 4.10 can be covered by iterating the FSAs as shall be seen in chapter 5.

A simplified rule for one kind of German noun chunk can look like the one in figure 4.12, where '*' (Kleene Star) means none or more. The rule covers noun chunks beginning with an article followed by zero or more adjectives and ended by the head noun. The rule shows that a noun chunk can be captured just using PoS tag information. This technique has already been extensively used for chunking. The important fact about topological fields is that they can be captured using the very same technique which has been used for chunking. Although topological fields already contain a lot of information about the structure of the sentence, it is possible to annotate them with FSAs just using PoS tags as input symbols. For this reason, they are annotated as a shallow structures before the annotation of grammatical functions.

Figure 4.12 shows how the skeleton of the sentence can be captured by showing a simplified rule for a verb-second (V2) clause, which is the prototypical affirmative clause. The sentence begins with the initial field (VF) which consists one or more chunks ('+', Kleene Plus) or subordinate clauses.[4] The second element in the clause is the LK, which contains the finite verb (hence 'verb-second' clause). Then follows an optional MF ('?' indicates zero or one) with at least one constituent. The right part of the sentence bracket is represented by VC which contains the non-finite verbal elements. There are not always non-finite elements. The NF follows the VC. It contains at

[4]The VF typically contains just one phrase. This may, however, consist of more then one chunk and/or subordinate clause.

```
noun chunk --> article adjective* noun

initial field (VF) --> (chunk|subordinate clause)+
middle field (MF) --> (chunk|subordinate clause)+
final field (NF) --> (chunk|subordinate clause)+
left part of sentence bracket (LK) --> finite verb
right part of sentence bracket (VC) --> non-finite verb

V2 clause --> VF LK MF? VC? NF?
```

Figure 4.12: Regular Expression (RE) Rules used in the FSAs

least one chunk or subordinate clause (NFs very often contain subordinate clauses). All topological field structures of German sentences can be captured by rules like this because they are an invariant pattern of the German language.

4.2 Definition of Shallow Structures

The following sections contain the definitions of the shallow structures, i.e. of chunks, topological fields and clauses. The names of the labels of constituents differ from those of the TüBa-D/Z. We decided in favour of a more fine-grained set of labels because we want to make use of the constituents in further annotation, and the more fine-grained distinctions allow a direct access to features of certain constituents. Thus, while, in TüBa-D/Z, there is just a distinction between simple clauses (SIMPX) and simple relative clauses (R-SIMPX), we distinguish between verb-first and verb-second clauses, on the one hand, and between infinitive clauses, relative clauses and general subordinate clauses within the verb-last clauses, on the other hand.

4.2.1 Chunks

There are five major types of chunks. Verb chunks (VC_)[5], noun chunks (NC), adjective chunks (AJ_C), adverb chunks (AVC) and prepositional chunks (PC). The label set for the chunks is structured according to the

[5]The '_' stands for one letter, if it occurs within a chunk name, and for one or more letters, if it occurs at the beginning or end of a chunk name.

Chunk Label	Definition
AJAC	attributive adjective chunk
AJACTRUNC	AJAC with truncated adjective
AJACC	at least two coordinated AJACs
AJVC	predicative adjective chunk/adverb chunk
AJVCTRUNC	AJVC with truncated adjective/adverb
AJVCC	at least two coordinated AJVCs
AVC	adverb chunk
AVCC	coordinated AVC
NC	noun chunk
NCTRUNC	NC with truncated noun
NCC	at least two coordinated NCs
NCell	elliptical noun chunk (i.e. without head noun)
PC	prepositional chunk
VCTRUNC	verb chunk with truncated verb
VCL_	verb chunk as left part of sentence bracket
VCR_	verb chunk as right part of sentence bracket
VCF_	verb chunk in topicalized (fronted) position
VCE_	verb chunk containing finite verb in *Ersatzinfinitiv* structure

Table 4.1: Overview of the chunk labels

principle of the *logical tagsets* described in Leech (1997), sec. 2.3.2, for PoS tagsets. The principle of logical tagsets, which is also applicable to other linguistic label sets, implies that the label set can be represented "as a hierarchical tree [...] with attributes being inherited from one level of the tree to another." This can be illustrated by the overview of the label sets for chunks in table 4.1: At the general level, there is the distinction between 'AJ' (adjective), 'AV' (adverb), 'N' (noun), 'P' (preposition) and 'V' (verb). Adjectives can further be divided into 'AJA' (attributive) and 'AJV' (adjectival or adverbial function).[6] The following 'C' in all chunks stands for chunk. Then follows either nothing (for unmarked), 'C' for coordinated or TRUNC for truncated. In the case of NCs, there is also the option of 'ell' for elliptical.

[6]Morphologically unmarked adjectives may function as either predicative adjectives or as adverbs in German.

1. **Conciseness** Brief labels are often more convenient to use than verbose, lengthy ones.

2. **Perspicuity** Labels which can easily be interpreted are more user-friendly than labels which cannot. [...]

3. **Analysability** Labels which are decomposable into their logical parts are better (particularly for machine processing, but also for human understanding) than those which are not.

Figure 4.13: Criteria for label sets from Leech (1997), p. 25

In the case of the C7 tagset, which Leech (1997) gives as an example of a hierarchically structured logical tagset, just one letter is assigned per hierarchical level. This is in line with the principle of conciseness in figure 4.13. However, it is not in line with the principle of perspicuity which demands labels to be easily interpretable. In C7, tags for adverbs get the letter 'R', which is not intuitive. In the case of the trade-off between conciseness and perspicuity, we have decided for perspicuity. The third principle of analysability in figure 4.13 makes it easier for humans to understand the category the label stands for, but it is especially useful for automatic parsing. The reason for this is that it is possible to match whole groups of labels by using REs like 'NC*' which would match **all** noun chunks. The advantage of this fact will become even more obvious in the following subsection dealing with verb chunks.

Verb Chunks

Verb chunks play a special role in the chunk structure because they are both chunks and part of the sentence bracket – and as such topological fields. We have refrained from adding a further layer of annotation on top of verb chunks to show that they are topological fields because that layer would not add any further linguistic information. Since the structure of the verb chunks already contains a lot of information about the type and the structure of the clause they occur in, their categorization is very fine-grained. This categorization is used in the annotation of the topological fields, which is the basis for the annotation of clauses. It is furthermore used in the annotation of grammatical functions. In order to maximize the usefulness of the verb

```
[NC
     .NN Dialoge ]
.$, ,
[NC
     .PRELS die ]
[AVC
     .PTKNEG nicht ]
[PC
     .APPR ohne
     [NC
          .NN Weiteres ] ]
[VCRAFVZ
     .PTKZU zu
     .VVINF verstehen
     .VAFIN sind ]
```

Figure 4.14: Verb chunk in relative clause

```
[NC
     .PRELS die ]
[NC
     .PPER er ]
[VCEAF
     .VAFIN hätte ]
[VCRMIVI
     .VVINF sehen
     .VMINF sollen ]
```

Figure 4.15: VCE_ in Ersatzinfinitivkonstruktion in relative clause

chunks in further annotation, special emphasis has been given to the logical label set design of verb chunks. This can be seen in table 4.2.

Verb chunks are categorized on the basis of their syntactic distribution and their internal structure. As regards syntactic distribution, there are four main types of verb chunks: VCL_, VCR_, VCF_ and VCE_. VC$\boxed{\text{L}}$ chunks are chunks which are the **left** part of the sentence bracket and VC$\boxed{\text{R}}$ chunks are chunks which are the **right** part of the sentence bracket. VC$\boxed{\text{F}}$ chunks are chunks which in the basic word order would be the right part of the sentence bracket but which are **front**ed and, thus, topicalized. VC$\boxed{\text{E}}$ chunks occur in **Ersatz**infinitiv constructions in subordinate clauses with verb-last position. They contain the finite verb, which – without the Ersatzinfinitiv occurring – would be part of the right part of the sentence bracket but which is moved to the left of it in such a construction (cf. figure 4.15). VC$\boxed{\text{E}}$ chunks are only recognized if no constituents intervene between the VC$\boxed{\text{E}}$ chunk and the VC$\boxed{\text{R}}$ chunk.

The following letters in the name of the verb chunk correspond to the second and third letters in the verb tag, which denote verb type (V=lexical, A=auxiliary, M=modal) and finiteness (F=finite, I=infinitive, P=perfect participle)[7]. The sequence of the verbs in the chunk type name corresponds to the syntactic dependency, which is the opposite of the sequence of the occurrence of the verbs in the sentence. Thus, in the sequence *Dialoge, die nicht ohne Weiteres zu verstehen sind* (cf. figure 4.14) the verbal complex *zu verstehen sind* is assigned the chunk type VC$\boxed{\text{R}}$ for right part of sentence bracket, $\boxed{\text{AF}}$ for auxiliary finite, $\boxed{\text{VZ}}$ for lexical verb/infinitive with 'zu'. As they are parts of verbs, verbal particles are also included in the class of the verb chunks with the chunk type name VCRPT.

The structure of the verb chunks as it is shown in table 4.2 adheres to the principles of logical label set design. This has great advantages in the following processing stages because it is a lot easier to match groups of verb chunks using an RE grammar. Thus, the analysability of labels – as it is postulated in figure 4.13 – can be used in the annotation of topological fields and, later on, in the annotation of grammatical functions. If we want to match, e.g., the left part of the sentence bracket only if it contains an auxiliary or a modal, we can use the RE 'VCL(A|M).*'. If we want to match the right

[7]An exception is being made in the treatment of the infinitive with 'zu', where 'I' is replaced with 'Z' in the chunk type name (cf. figure 4.14) and with the imperative where a 'B' is assigned.

verb chunk	position in clause	verb category	verb finiteness	...
VC	L=left	V=lexical	F=finite	...
	R=right	A=auxiliary	I=infinitive	...
	F=fronted	M=modal	P=perfect participle	...
	E=Ersatzinfinitiv		Z=infinitive with 'zu'	...
			B=imperative	...

Table 4.2: Logical label set design for verb chunks

```
[PC
     .APPRART im
     [NC
          .NN Interesse ] ]
[NC
     .PIAT aller
     .NN Mitgliedstaaten ]
```

Figure 4.16: Noun chunks

part of the sentence bracket only if it is finite, we can use the RE 'VCR.F.*'. Thus, although the logical design of the label set is not indispensable, it makes writing grammar rules more convenient and transparent and, thus, the grammar more reliable.

Noun Chunks

Noun chunks are the most common chunks. They consist at least of a noun, pronoun or cardinal number as a head word and of optional determiners, adverb chunks or attributive adjective chunks (cf. figures 4.16, 4.22 and 4.23). As the definition of chunks does not allow recursive structures except in cases of center-embedding, post-modifying prepositional or nominal constituents may not be part of a noun chunk (cf. figure 4.16). In case of preposition-article contraction (APPRART), the contraction is annotated in its function as a preposition (cf. figure 4.16).

As pronouns are normally not modified, they are the only element of a noun chunk when they occur (cf. figure 4.28). Adverb chunks are only included in a noun chunk when they occur after clear syntactic indicators

```
[PC
     .APPR um
     [NC
          .CARD sechs ] ]
```

Figure 4.17: Cardinal as the head of a noun chunk

```
[PC
     .APPR aus
     [NC
          [AJAC
               [AVC
                    .ADV bloß ]
               .ADJA wirtschaftlichen ]
          .NN Motiven ] ]
```

Figure 4.18: Modifying adverb chunk in attributive adjective chunk

```
[NCC
     [NC
          .ART die
          [AJAC
               .ADJA großen ]
          .NN Gnus ]
     .KON und
     [NC
          .NN Zebras ] ]
```

Figure 4.19: Two coordinated NCs

```
[NC
     .ART Der
     [AJAC
          .ADJA älteste ]
     .NN Künstler ]
[VCLAF
     .VAFIN ist ]
[NC
     .NN Jahrgang ]
[NC
     .CARD 1929 ]
.$, ,
[NCell
     .ART der
     [AJAC
          .ADJA jüngste ] ]
[NC
     .CARD 1963 ]
```

Figure 4.20: Sentence containing elliptical NC

```
[NCC
     [NCell
          .ART das
          [AJAC
               .ADJA neunzehnte ] ]
     .KON und
     [NC
          .ART das
          [AJAC
               .ADJA zwanzigste ]
          .NN Jahrhundert ] ]
```

Figure 4.21: Elliptical NC coordinated with NC

for the beginning of a noun chunk and before the head word, because, in the other cases, their attachment is ambiguous without taking into account semantic or lexical knowledge. In case adverbs modify an adjective or a cardinal number, they are attached to the AJAC chunk (cf. figures 4.18 and 4.23). Clear indicators of the beginning of a noun chunk are determiners or attributive adjectives, but also prepositions (which are themselves not part of a noun chunk) (cf. figure 4.18). If there is no clear syntactic indication that the adverb belongs to the NC, it is not included. In ambiguous cases, adverbs are, thus, attached to the higher of the potential nodes in the tree.

If noun chunks are coordinated, they are chunked as a coordinated noun chunk NCC. If the coordinated noun chunks are modified by an attributive adjective chunk, the adjective chunk becomes part of the first noun chunk as it is impossible to decide on the chunk level whether the adjective chunk modifies both nouns or the first noun only (cf. figure 4.19). Very often world knowledge or the understanding of the wider textual context would be required to solve the ambiguity. The same applies to articles which might refer to both noun chunks or the first noun chunk only.

The only other possibility to avoid this problem would be to apply the same strategy as with the adverbs and attach the adjective chunk and the determiner to a higher node like in example 28. This would even allow us to add an extra constituent in the case that the adjective and the determiner do in fact modify both nouns (cf. example 29). However, there would still be no way to straightforwardly generate the desired structure from this structure if the adjective and determiner just modify the first noun; and, what is even more important, there would also be a need to annotate coordinated NCs **before** simple NCs because in the inverse case the structure of the simple chunks would already exist at the time of the annotation of coordinated NCs. This would, however, result in a far more complex rule system as certain structures would have to be dealt with both in the rules for the coordinated and the simple NCs.

If simple NCs are treated before coordinated NCs, coordinated NCs can easily be annotated by just grouping two or more coordinated simple chunks. Since the higher attachment of the determiner and the adjective would, thus, not be of advantage for further annotation and even complicate chunking, we annotate coordinated NCs modified by a determiner and/or adjective as it is shown in figure 4.19. This does not always yield perfect results but the results can be accurately defined and, thus, be dealt with in further stages of annotation. We, thus, accept this problem as a problem of the chunking

```
[AJVC
     .ADJD furchtbar ]
[NC
     [AJAC
          .ADJA militaristische ]
     .NN Trick-Aufnahmen ]
```

Figure 4.22: AJAC chunk with modifying ADJD attached outside chunk

approach which cannot take into account semantic or world knowledge.

(28) $[_{\text{NCC}}$ die $[_{\text{AJAC}}$ großen] $[_{\text{NC}}$ Gnus] und $[_{\text{NC}}$ Zebras]]

(29) $[_{\text{NC}}$ die $[_{\text{AJAC}}$ großen] $[_{\text{NCC}}$ $[_{\text{NC}}$ Gnus] und $[_{\text{NC}}$ Zebras]]]

There is one kind of NC which does not have a noun, pronoun or cardinal number as its head word. These NCs are labelled NCell for elliptical noun chunk. An NCell must consist of at least an attributive adjective chunk. In cases in which an NCell is coordinated with the NC containing its head word, those noun chunks are also chunked as a coordinated noun chunk (cf. figure 4.21).

Attributive Adjective Chunks

There are two main types of adjective chunks: attributive adjective chunks and predicative/adverbial adjective chunks. The distinction between them is made on the basis of the PoS tags of the head word of the chunk (i.e. the adjective tags ADJA (attributive) or ADJD (predicative/adverbial)). AJAC chunks are chunks with an attributive adjective (or cardinal number) as their head. The definition of an attributive adjective being that it modifies a noun, the AJAC chunk is always part of a noun chunk. AJAC chunks may contain modifying PCs in center-embedded structures. A modifying adverb (ADV) or ADJD may be included in the AJAC if there is clear indication of its containment in the noun chunk. If, however, there is no such clear indication, the ADV or ADJD is left outside the NC and, consequently, outside the AJAC (cf. figure 4.22).

```
[NC
     .ART die
     [AJAC
          [AVC
               .ADJD ungefähr ]
          .CARD vierzig ]
     [AJAC
          .ADJA jungen ]
     .NN Männer ]
```

Figure 4.23: Cardinal number as head of an attributive adjective chunk

```
[NC
     .ART die
     [AJACC
          [AJAC
               .ADJA große ]
          .$, ,
          [AJAC
               .ADJA weite ] ]
     .NN Welt ]
```

Figure 4.24: AJAC chunks coordinated by comma

```
[NC
     [AJACC
          [AJAC
               .ADJA ausgekochte ]
          .KON und
          [AJAC
               .ADJA geschäftstüchtige ] ]
     .NN Musiker ]
```

Figure 4.25: AJAC chunks coordinated by *und*

```
[NC
     [AJAC
          .ADJA traditionelle ]
     [AJAC
          .ADJA klassische ]
     .NN Musik ]
```

Figure 4.26: Two AJAC chunks

```
[AVC
     .ADV Auch ]
[NC
     .ART die
     .NN Kelten ]
[VCLAF
     .VAFIN waren ]
[AJVC
     .ADJD eitel ]
```

Figure 4.27: AJVC as a predicative adjective chunk

AJAC chunks may be coordinated in two ways: with commas or with a coordinator (cf. figure 4.24 and figure 4.25). In this case they are projected to an AJAC$\boxed{\text{C}}$ chunk. Two immediately successive attributive adjective chunks are not projected to a coordinated chunk, as they are not in a relation of coordination (cf. figure 4.26). In cases in which there is no head noun after the attributive adjective chunk, this chunk – together with other elements like determiners – forms the noun chunk. The head noun of this elliptical noun chunk (NCell) may be inherent or it may precede (cf. figure 4.20) or follow (cf. figure 4.21) the elliptical noun chunk.

Predicative Adjective Chunks/Adverbial Adjective Chunks

The tag ADJD, on the basis of which the chunk type AJVC is recognized, is assigned on the grounds of morpho-syntactic features, but the word to which the tag ADJD is assigned may function either as an adjective (cf. figure 4.27) or as an adverb (cf. figure 4.28). Obviously, in the tagset, no distinction was

```
[AVC
     .ADV Da ]
[VCLVF
     .VVFIN kommt ]
[NC
     .PPER Dir ]
[NC
     .ART das
     [AJAC
          .ADJA normale ]
     .NN Leben ]
[AJVC
     .ADJD richtig
     .ADJD langweilig ]
[VCRPT
     .PTKVZ vor ]
```

Figure 4.28: One AJVC as an adverbial and the other as a predicative adjective

made because it is impossible to draw it without making a full parse. As this is still true with respect to the chunking level, the decision is not made on this level, either; especially as it does not lead to any negative effects on the chunking level. The decision whether the chunk serves as an adverb chunk or a predicative adjective chunk is, thus, delayed until further annotation and left underspecified. The decision is made by the GF annotation process in which predicative adjectives are assigned the label PRED. The label AJVC has, thus, to be read as 'either predicative adjective **or** adverb chunk'.

Since it cannot be detected by syntactic restrictions whether an ADV is a modifier of an ADJD, ADVs as modifiers are not chunked together with the AJVC. However, since not annotating an ADJD which modifies an ADJD together as one chunk would lead to more AJVCs as potential predicatives, we annotate them as one AJVC (cf. figure 4.28). Evaluation will show whether this strategy was correct. In the case of coordination of AJVC chunks, they are projected to an AJVC$\boxed{\text{C}}$ chunk (cf. figure 4.29). Coordination may occur with or without coordinator.

```
[AJVCC
     [AJVC
          .ADJD schnell ]
     .KON und
     [AJVC
          .ADJD frisch ] ]
```

Figure 4.29: Coordinated AJVC chunks

```
[AVCC
     [AVC
          .ADV nachts ]
     .KON oder
     [AVC
          .ADV sonntags ] ]
```

Figure 4.30: Coordinated AVC chunks

Adverb Chunks

In the cases in which the attachment of adverbs is ambiguous, the site of their attachment is not specified. The adverb chunk (AVC) is then not part of the modified chunk. In most cases, adverb chunks consist of a single adverb only. The particle *nicht*, which is tagged PTKNEG, is counted among the adverbs. Adverb chunks cannot contain any other constituents than adverbs. Coordinated adverb chunks are grouped into a chunk labelled AVC$\boxed{\text{C}}$ analogous to the adjective chunks and the noun chunks (cf. figure 4.30).

Prepositional Chunks

Prepositional chunks typically consist of a preposition and a noun chunk. In most cases, the preposition precedes the noun chunk. In some cases, it follows the noun chunk. It may then be called a *post-position* (cf. figure 4.31). In some cases, there is a preceding and a following preposition. This phenomenon may be called a *circumposition* (cf. figure 4.32). Contractions of pronouns and prepositions (e.g. *darauf*, *deswegen* or *hiermit*), which are tagged PROAV, may be the only constituents of a PC. Sometimes, the head

```
[PC
     [NC
          [AJAC
               .CARD drei ]
          NN Wochen ]
     .APPO lang ]
```

Figure 4.31: PC with post-position

```
[PC
     .APPR von
     [NC
          .NN Anfang ]
     .APZR an ]
```

Figure 4.32: PC with circumposition

of a prepositional chunk is a token tagged as an adverb (cf. figure 4.33). In some cases, a prepositional chunk may contain a preposition followed by another one, e.g. *bis* followed by another preposition (cf. figure 4.34).

Truncated Chunks

There is another category of chunks which just contains truncated, i.e. fragmentary, words (_CTRUNC). Those words receive the tag TRUNC in the STTS. They cannot usually be assigned to the appropriate word category, as they lack important morphological information. Using contextual information, it is, however, very often possible to detect the proper word category. Fragmentary words are, thus, assigned a chunk corresponding to the respective word category, which is then treated like any ordinary chunk of its cat-

```
[PC
     .APPR seit
     .ADV gestern ]
```

Figure 4.33: Prepositional chunk with adverb head

```
[PC
     .APPR bis
     .APPRART zum
     [NC
          [AJAC
               .ADJA letzten ]
          .NN Augenblick ] ]
```

Figure 4.34: PC with 'complex' preposition

```
[PC
     .APPR in
     [NCC
          [NC
               .ART einer
               [NCTRUNC
                    .TRUNC Rundfunk- ] ]
          .KON und
          [NC
               .NN Fernsehansprache ] ] ]
```

Figure 4.35: Truncated noun chunk (NCTRUNC)

egory (cf. figure 4.35). Truncated chunks are, thus, not a chunk category on its own, but rather belong – according to their head words – as a subcategory to the respective chunk category. Truncated words, are, thus, treated just like their non-fragmentary counterparts. In cases in which disambiguation of the word category is impossible, truncated words are not chunked at all.

4.2.2 Topological Fields

A distinction can be made between two types of topological fields: On the one hand, there are fields which are part of the sentence bracket. They can typically only contain tokens of a restricted number of Parts-of-Speech, namely complementizers and verbs. The left part of the sentence bracket is realized by the topological fields CF (complementizer field) and VCL_ (finite verb in V1 and V2 clauses); the right part of the sentence bracket is realized by the topological field VCR_ (non-finite verbs in V1 and V2 clauses and all verbs in VL clauses). VCL_ and VCR_ are chunks and topological fields at one and the same time. On the other hand, there are fields which can be described relative to the sentence bracket. They can contain tokens of all other Parts-of-Speech. The constituent order in the fields which are not part of the sentence bracket is more varied than in those fields which are part of the sentence bracket. The fields VF, MF, NF and LV are not part of the sentence bracket. A special case is the KOORDF and the PARORDF. These fields just contain one constituent, i.e. the coordinating particle in KOORDF and the non-coordinating particle in PARORDF). These two fields may occur at the beginning of a clause.

The Complementizer Field (CF)

The CF only occurs in subordinated clauses introduced by a complementizer. It is always the left part of the sentence bracket. CFs usually contain just one token (i.e. the complementizer). These complementizers may be relative pronouns (PRELS, PRELAT), interrogative pronouns and adverbial interrogative pronouns in indirect questions (PWAT, PWS, PWAV) or the subordinating conjunctions KOUI and KOUS. In the cases in which these tokens are attributive, the whole noun chunk belongs to the CF (cf. sentences 30 and 31).[8] In some cases, a complementizer may consist of a complex token like *so dass/daß* or *als ob* (cf. sentence 32). The CF can also be occupied by an expression like the one in sentence 33, which also introduces a subordinate clause.

(30)

Sie	stammen	aus	Ländern,	[$_{\text{CF}}$ deren/PRELAT	Regierungen]	
They	come	from	countries	whose	governments	
keinerlei	Respekt	für	die	Menschenrechte	haben.	
no (at all)	respect	for	the	human rights	have.	

‘They come from countries whose governments have no respect for human rights, at all.’

(31)

Niemand	weiß,	[$_{\text{CF}}$ welches/PWAT	Datum]	das	Dokument	trägt.	
Nobody	knows	which	date	the	document	has.	

‘Nobody knows which date the document has.’

[8]The example sentences in this section and the following sections just contain the linguistic markup relevant for the respective sections.

(32) Die meisten Besucher kennt man, [$_{\text{CF}}$ so daß] die Sicherheitsprozedur
The most visitors knows one so that the safety procedure
entfällt.
not apply.

'You know most of the visitors so that the safety procedures can be omitted.'

(33) [$_{\text{CF}}$ Je mehr Dinge] man zu erledigen hat, desto mehr Zeit hat man.
The more things one to complete has, the more time has one.

'The more things you have to take care of the more time you have.'

The Left Part of the Sentence Bracket (VCL_)

While the CF is the left part of the sentence bracket in introduced subordinate clauses (VL clauses), the VCL_ is the left part of the sentence bracket in V1 and V2 clauses which are typically main clauses (cf. sentences 34 and 35) and sometimes non-introduced subordinate clauses (cf. sentences 36 and 37). The VCL_ always just contains one finite verb of the categories lexical verb, auxiliary verb or modal verb.

(34) Ein Almbauer aus Bayrischzell [$_{\text{VCLVF}}$ verhindert] den Skisport
An alp farmer from Bayrischzell obstructs the ski sport
am Wendelstein.
at the Wendelstein.

'An alp farmer from Bayrischzell obstructs ski sports at the Wendelstein.'

(35) Jetzt [$_{\text{VCLMF}}$ wollen] die Sozis einen Antrag [$_{\text{VCRVF}}$ einbringen].
Now want the socialists a petition file.

'Now, the socialist want to file a petition.'

(36) Kowaljow hatte angekündigt, er [$_{\text{VCLAF}}$ werde] den Vorwürfen [$_{\text{VCRVF}}$
Kowaljow had announced, he would the allegations
nachgehen].
pursue.

'Kowaljow had announced that he would pursue the allegations.'

(37) [$_{\text{VCLVF}}$ Stimmt] der Präsident auch zu, fehlt immer noch das Ja
Agrees the president even to, misses still the Yes
des Parlaments.
of the parliament.

'Even if the president agrees to it, there is still the 'Yes' of the parliament missing.'

The Right Part of the Sentence Bracket (VCR_)

VCR_ is defined as being the right part of the sentence bracket. VCR_ is obligatory in all introduced subordinate clauses (i.e. verb-last clauses) and may occur in all kinds of other clauses provided that they contain a complex predicate (i.e. a predicate consisting of two verbs or a verb and a verbal particle). VCR_ may contain one or more tokens. In introduced subordinate clauses, VCR_ contains all the verbal elements and the CF constitutes the left part of the sentence bracket (cf. sentences 38 and 39); in main clauses VCR_ contains all the verbal elements except for the finite verb (which is contained in the VCL_) (cf. sentences 40 and 41).

(38) Ein Antrag, [$_{\text{CF}}$ dem/PRELS] weder die rot-grüne Koalition noch
A motion, which neither the red-green coalition nor
die PDS [$_{\text{VCRMFVI}}$ zustimmen mochten].
the PDS accept like.

'A motion, which neither the reg-green coalition nor the PDS liked to accept.'

(39) Klar, [$_{\text{CF}}$ daß/KOUS] dieser Antrag keine Mehrheit [$_{\text{VCRVF}}$ fand].
Evident, that this motion no majority found.

'It is evident, that this motion did not get the majority.'

(40) Für eine Feier auf öffentlichen Plätzen [$_{\text{VCLAF}}$ hätte] eine
For a ceremony in public places had an
eindeutige Einladung [$_{\text{VCRMIVI}}$ vorliegen müssen].
unequivocal invitation exist must.

'For a ceremony in public places, there would have had to be an unequivocal invitation.'

(41) Etwaige Sicherheitsbedenken [$_{\text{VCLVF}}$ wies] er entschieden [$_{\text{VCRPT}}$
Possible safety considerations turned he roundly
zurück].
away.

'Any safety considerations were roundly rejected by him.'

The Vorfeld (VF)

The VF is defined as the topological field enclosed by the beginning of the sentence on the left-hand side and the VCL_ on the right-hand side. A VF may contain all kinds of constituents except the ones contained in the sentence bracket (i.e. verbal elements and complementizers). An exception is the fronted and thus topicalized right part of the sentence bracket which is labelled VCF_ and enclosed in the VF (cf. sentence 42). As a VF may contain subordinate clauses, a clausal frame as a part of such a subordinate clause may be contained in the VF (cf. sentence 43). Typically, a VF just contains one constituent (which may, however, be very complex; see sentence 44). However, some adverbs (e.g. *freilich*; see sentence 45) may occur along other constituents.

(42) [$_{\text{VF}}$ [$_{\text{VCFVI}}$ Abnehmen]] [$_{\text{VCLVF}}$ kann] ihnen das keiner.
Take from can them that nobody.

'Nobody can take(believe) that from them.'

(43) [$_{\text{VF}}$ [$_{\text{SUB}}$ Daß ihr Vorhaben auf Widerstand stoßen würde]], [$_{\text{VCLAF}}$
That their plan with opposition meet would,
war] den Transplanteuren in Hannover bewußt.
was the transplanters in Hannover clear.

'The transplanters in Hannover were aware of the fact that their plan would meet with opposition.'

(44) [$_{\text{VF}}$ Die Lage der noch etwa 15.000 verbliebenen Einwohner
The situation of the still about 15,000 remaining inhabitants
Grosnys, die seit Wochen in den Kellern der belagerten Stadt
of Grosny, who for weeks in the cellars of the besieged city
ausharren], [$_{\text{VCLAF}}$ ist] katastrophal.
hold out, is disastrous.

'The situation is disastrous for the approximately 15,000 remaining inhabitants of Grosny who have been holding out in the cellars of the besieged city for weeks.'

(45) [$_{\text{VF}}$ [$_{\text{PC}}$ Unter dieser Bedingung] [$_{\text{AVC}}$ freilich]] [$_{\text{VCLVF}}$ wäre] man
Under these conditions however would have one
mit der Vereidigung auf öffentliche Plätze gegangen.
with the swearing-in into public places gone.

'Under these conditions, however, they would have gone into public places with the swearing-in ceremony.'

The Mittelfeld (MF)

The MF is defined as the topological field which is enclosed by the left part of the sentence bracket (i.e. VCL_ or CF) and the right part of the sentence bracket (i.e. VCR_). In cases in which no constituent appears between the left part of the sentence bracket and the right part of the sentence bracket, no MF is annotated (cf. sentence 46). If there is no right part of the sentence bracket, the MF ends at the end of the sentence or at the beginning of a new main clause (cf. sentences 47 and 48), or at the beginning of the Nachfeld (NF) (cf. sentence 49). Sentence 49 also shows that an MF may begin after a comma in non-introduced non-finite clauses and that the MF of the matrix clause ends where this clause begins. In cases like this one, automatic annotation very much relies on punctuation.

(46) Mehrere weitere Menschen [$_{\text{VCLAF}}$ wurden] [$_{\text{VCRVF}}$ verletzt].
Several other people were injured.

'Several other people were injured.'

(47) Der Mann [$_{\text{VCLVF}}$ verletzte] [$_{\text{MF}}$ sich dabei zum Glück nur
The man injured himself in doing so luckily only
leicht].
slightly.

'In doing so, the man luckily only got slightly injured.'

(48) Wir [$_{\text{VCLVF}}$ verkaufen] [$_{\text{MF}}$ ihnen keinen Reis], und dann kriegen wir
We sell them no rice, and then get we
keine Bananen.
no bananas.

'We don't sell them any rice, and then we don't get any bananas.'

(49) Lenin-Räuber [$_{\text{VCLVF}}$ versuchten] [$_{\text{MF}}$ vergeblich], [$_{\text{NF}}$ [$_{\text{INF}}$ [$_{\text{MF}}$ einen
Lenin robbers tried in vain a
im Wald vergrabenen Lenin] [$_{\text{VCRVZ}}$ zu klauen]]].
in the woods buried Lenin to steal.

'Lenin robbers tried in vain to steal a (statue of) Lenin which was buried in the woods.'

The Nachfeld (NF)

The NF is defined as the topological field after the right part of the sentence bracket. In the case that there is not right part of the sentence bracket, the NF is defined as the field containing certain constituents like subordinate clauses occurring at the end of the sentence. This is in line with the definition of NFs in Telljohann, Hinrichs, and Kübler (2003). An NF may contain constituents of various categories. It may, however, not contain all kinds of grammatical functions. As this is of less importance in shallow annotation, it will not be discussed here. The most typical constituent of an NF is a subordinate clause (cf. sentences 49 and 50); another typical but less frequent constituent is a phrase introduced by a *Vergleichspartikel* (cf. sentence 51). Other constituents include prepositional phrases like the one in sentence 52 or even conjuncts like the one in 53. These cases are not typical cases of NF constituents but rather cases in which the author wanted to evoke some dramatic effect. This is even more the case with constituents which can be seen as a kind of addendum or afterthought (cf. sentence 54).

(50) Plötzlich merkte ich, [$_{\text{NF}}$ was für ein ungeheurer Druck auf mir
Suddenly realized I, what an enormous pressure on me
lastete].
laid.

'Suddenly, I realized what an enormous pressure laid on me.'

(51) In Deutschland wurden 4,6 % mehr für Tabakwaren ausgegeben [$_{\text{NF}}$
In Germany were 4.6 % more for tobacco goods spent
als im Vorjahr].
than in the previous year.

'In Germany, 4.6% more money was spent on tobacco than in the previous year.'

(52) Die Dresdner Semperoper ist vollbesetzt $[_{\mathrm{NF}}$ bis in den vierten Rang].
The Dresden Semperoper is fully taken up to the fourth tier.
‘The Dresden Semperoper is fully taken, up to the fourth tier.’

(53) Die Konfrontation solle in den Museen stattfinden – $[_{\mathrm{NF}}$ oder auf
The confrontation should in the museums take place – or in
der Straße].
the streets.
‘The confrontation should take place in the museums – or in the streets.’

(54) Er will beim Management umgerechnet 350 Mark
He wants of the management converted 350 Deutschmarks
rausholen, $[_{\mathrm{NF}}$ das Doppelte des Monatslohns].
get out, twice the monthly wage.
‘He wants to get, converted into Deutschmarks, 350 Deutschmarks from the management, which is twice the monthly wage.’

The Linksversetzung (LV)

The topological field LV is used to annotate resumptive constructions, in which a constituent is dislocated and moved to the left in front of the VF. This constituent is then resumed in its original place, which is the VF (cf. sentences 55 and 56). However, in the special case of the *nominativus pendens*, the referring pronoun may be situated in another place (cf. sentence 57). Due to the limitations of a shallow annotation, these cases can, however, not be recognized. LVs are more likely to occur in spoken language. However, a construction as shown in sentence 55, in which it is a clause which is fronted, is not infrequent in written language. It occurs 130 times in the 15,260 sentences of the TüBa-D/Z.

(55) $[_{\mathrm{LV}}$ Wenn die Leute schon Skifahren müssen], $[_{\mathrm{VF}}$ dann] sollen
If the people actually skiing must, then should
sie es tun, wenn genug Schnee da ist.
they it do, when enough snow there is.
‘If people actually have to go skiing, then they should do it when there is enough snow.’

(56) $[_{\text{LV}}$ “Infos”], $[_{\text{VF}}$ das] sind vor allen Dingen lokale Nachrichten.
“Infos”, that are before all things local news.

‘ “Infos”, that’s most of all local news.’

(57) $[_{\text{LV}}$ Ein frühes Tor], $[_{\text{VF}}$ jeder Trainer] würde sich wohl
An early goal, every coach would himself probably
darüber freuen.
about that rejoice.

‘An early goal, that’s what every coach would probably be happy about.’

The Coordination Fields (KOORDF and PARORDF)

Höhle (1986) states that the KOORDF is not a field in between sentences but a field introducing a sentence. He argues that sentences containing a KOORDF can be uttered without a preceding sentence which can be interpreted as its first conjunct. We share this view because it is supported by empirical data. Sentence 58, for example, has no first conjunct. Furthermore, there are examples like sentence 59, which very often must be interpreted as referring to more than one preceding sentence. However, there are also sentences like 60 and 61, in which there are doubtlessly two conjuncts. Sentence 61 shows the alternative field to the KOORDF, the PARORDF.

The difference between PARORDF and KOORDF is that, if two clauses occur, the conjunction in KOORDF coordinates the two clauses both syntactically and semantically while the conjunction in PARORDF only coordinates the clauses syntactically. Semantically, there exists a relation of subordination because the second clause in sentence 61 gives the reason for the proposition made in the first clause. This can be tested by reverting the order of the two conjuncts. With KOORDF which contains conjunctions like ‘und’ or ‘oder’ the meaning of the sentence does not change. With PARORDF, however, which contains conjunctions like ‘denn’ or ‘weil’ (this subsume just the ‘weil’ which is used in V2 clauses) the meaning changes.

(58) Welche Verleger sind mit welchen Konzepten in der Stadt
Which publishers are with which concepts in the town
ansässig? Was machen das Literaturkontor und die
resident? What do the literature branch office and the
Literaturzeitschriften? $[_{\text{KOORDF}}$ Und] $[_{\text{VF}}$ auch die Bücher selbst]
literature journals? And also the books themselves

sollen gelobt oder verrissen werden.
should praised or panned be.

'Which publishers are resident in town and with which concepts? What's the plan of the literature branch office and the literature journals? And also the books themselves should be praised or panned.'

(59) [$_{\text{KOORDF}}$ Doch] [$_{\text{VF}}$ daraus] wird nun nichts.
But from this becomes now nothing.

'But it will all come to nothing, now.'

(60) Sie liegen auf der anderen Seite des Erdballs [$_{\text{KOORDF}}$ und] [$_{\text{VF}}$
They are on the other side of the earth and
ihr Streit] erscheint weit entfernt.
their conflict seems far away.

'They are on the other side of the earth and their conflict seems far away.'

(61) Das sei ihm verziehen, [$_{\text{PARORDF}}$ denn] [$_{\text{VF}}$ um ihn] brennt die
That be him forgiven for around him burns the
Luft beim Sendestart.
air at the beginning of the program.

'He should be forgiven for that because the air burns around him when he begins his program.'

4.2.3 Clauses

Clauses are annotated according to the scheme of V1, V2 and VL clauses. There is no further subdivision in the categories of V1 and V2 clauses since they are typically maximal clauses, i.e. they do not have any function within a matrix clause (cf. figure 4.36 and 4.37). In cases in which they do have a function in superordinate clauses, this function has to be resolved on later annotation level because the shallow annotation level works with syntactic restrictions alone. With these, it is not possible to determine the function of a V1 or V2 clause. VL clauses, which are typically subordinate clauses, are subdivided into three different categories: general subordinate clauses (SUB, cf. figure 4.37), infinitive clauses (INF, cf. figure 4.37) and relative clauses (REL, cf. figure 4.38). The category REL subsumes all relative clauses introduced by relative pronouns. Clauses introduced by adverbial relative pronouns (i.e. PWAV) are annotated as SUB because, from the perspective of

```
(V1
    [VCLMF
        .VMFIN   Kann ]
    {MF
        [NC
            .ART     die
            .NN      Talkshow ]
        [PC
            .APPRART im
            [NC
                .NN      Privatleben ] ]
        [NC
            [AJAC
                .ADJA    konstruktive ]
            .NN      Gespräche ] }
    [VCRVI
        .VVINF   initiieren ]
    .$.      ? )
```

Figure 4.36: V1 clause

a shallow annotation, it is not clear in which cases a PWAV is in fact a relative pronoun. The category INF subsumes all non-finite clauses containing an infinitive (not the ones containing a participle). It includes clauses introduced by a complementizer and those which are non-introduced. The category SUB subsumes all other introduced subordinate clauses, which are mainly adverbial clauses.

The annotation of the clauses is shallow in the sense that the attachment of subordinate clauses is left unresolved just like the attachment of prepositional chunks is left unresolved in the chunk annotation. This means, for example, that the reference noun of a relative clause is not given. Thus, the treatment of recursion in clauses is analogous to the treatment of recursion in chunks. Clauses cannot contain clauses of the same type unless they are center-embedded. The reason for this treatment is the same as with the chunks. If recursion in center-embedded clauses is not treated, structures are generated which are not adequate shallow annotation structures.

```
(V2
    {VF
        (SUB
            {CF
                [NC
                    .PWS     Wer ] }
            {MF
                [AVC
                    .ADV     aber ]
                [PC
                    .APPR    von
                    [NC
                        .NN      Propaganda ] ] }
            [VCRAFVP
                .VVPP    verdorben
                .VAFIN   ist ] )
        .$,      , }
    [VCLAF
        .VAFIN   hat ]
    {MF
        [AVC
            .ADV     wohl ]
        [NC
            .NN      Grund ] }
    .$,      ,
    {NF
        (INF
            {MF
                [NC
                    .PRF     sich ] }
            [VCRVZ
                .PTKZU   zu
                .VVINF   fürchten ] ) }
    .$.      . )
```

Figure 4.37: V2 clause with VL clause SUB embedded into its VF and VL clause INF embedded into its NF

```
(V2
    {VF
        [NC
            .PPER    Es ] }
    [VCLVF
        .VVFIN   handelt ]
    {MF
        [NC
            .PRF     sich ]
        [AVC
            .ADV     eben ]
        [PC
            .APPR    um
            [NC
                .ART     einen
                [AJAC
                    .ADJA    doofen ]
                .NN      Trick ] ] }
    .$,       ,
    {NF
        (REL
            {CF
                [NC
                    .PRELS   der ] }
            {MF
                [AVC
                    .ADV     dennoch ] }
            [VCRVF
                .VVFIN   verfängt ] ) }
    .$.       . )
```

Figure 4.38: VL clause REL embedded into NF of V2 clause

Chapter 5

Annotating Shallow Structures with FSAs

Abstract: After the motivation and definition of shallow structures in chapter 4, chapter 5 describes the methods and strategies for the annotation of shallow structures. Section 5.1 gives an outline of the task description, i.e. it describes the input to the shallow parsing component and the desired output. It also discusses some problems of annotation. Section 5.2 explains how we use topological fields to apply an *Easy-First-Parsing* and a *Divide-and-Conquer* strategy for shallow parsing. Section 5.3 shows how we use these strategies to correct PoS tags which have a particularly negative effect on parsing accuracy in German. Section 5.4 discusses how we deal with the problem that FSA cannot handle recursion. Section 5.5 motivates why we prefer a *top-down* strategy for the annotation of chunks to a *bottom-up* strategy.

5.1 Outline of the Task Description

The shallow parser described in this chapter is part of the annotation system KaRoPars described in Ule and Müller (2004). The shallow parser takes text as its input which is already segmented into sentences and tokens and which is already Part-of-Speech (PoS) tagged according to the STTS tagset defined in Schiller, Teufel, and Thielen (1995). The parser is implemented as a cascade of FSAs using the tool *fsgmatch*, which is the main component of the *Text Tokenisation Tool* (Grover, Matheson, Mikheev, and Moens, 2000).

Beim	nächsten	Mal	wird	es	schwieriger	sein	,	ein	Verbot	mit	der	Zahl	von	Polizisten	zu	begründen	.
APPRART	ADJA	NN	VAFIN	PPER	ADJD	VAINF	$,	ART	NN	APPR	ART	NN	APPR	NN	PTKZU	VVINF	$.

Figure 5.1: Input to shallow parser

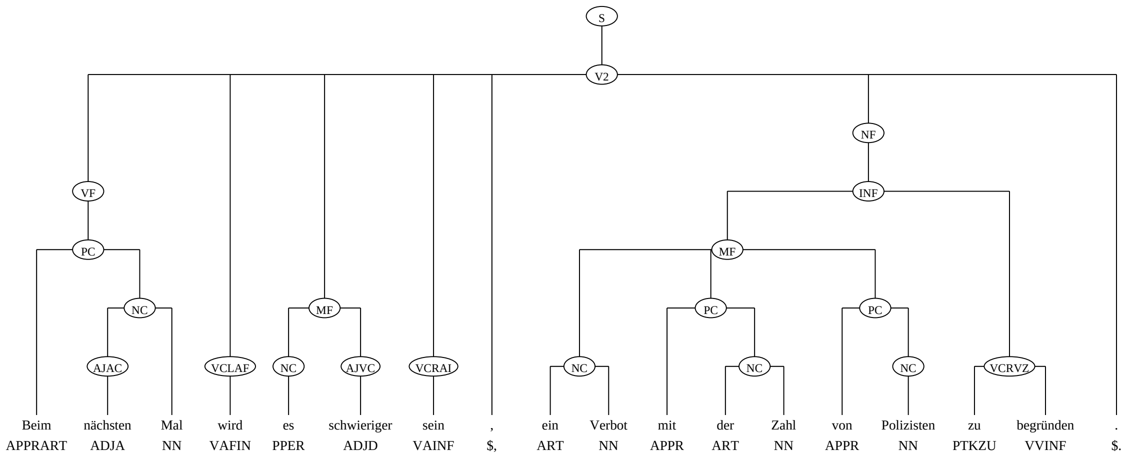


Figure 5.2: Output of shallow parser

The parser takes PoS tags as its input (cf. figure 5.1; words are shown for illustration) and produces chunks, topological fields and base clauses as its output (cf. figure 5.2). Only in the case of subordinators (KOUS and KOUI) and conjunctions (KON), the STTS tagset had to be refined because those classes of PoS tags are relatively heterogeneous. The shallow parser has a rather small grammar consisting of 288 rules. 103 of these are for verb chunks since we put a lot of emphasis on this structure. 80 rules are for all the other chunks. 78 rules are for topological fields and 27 rules are for clauses.

It is, thus, the main issue of shallow parsing to find the rules which allow the generation of the structure in figure 5.2 from the input string in figure 5.1. There is, however, also a related, more general question, namely the question of dealing with ambiguous structures. Since we are just taking into account PoS tag information at the point of shallow parsing, some structures are ambiguous. It is, consequently, the question how to deal with those structures in order to achieve optimal annotation and minimize problems for further steps of annotation. One way of achieving this is the shallow parsing structure itself since it leaves certain ambiguities open. Modifying adverbs and PPs are not attached by the shallow parser.

There are, however, other phenomena which cannot be left open since that would prevent any kind of meaningful annotation. One such phenomenon is the scope of coordination. The difference between the coordination of two phrases and of two sentences can sometimes not be resolved locally as can be seen in sentence 62. The string ‘die Spieler und die Zuschauer’ could be interpreted as a coordinated chunk because it could, in fact, be one in a different context. In order to realize that the ‘und’ in sentence 62 does not connect the two chunks but the two clauses, one would have to look at the whole clause structure. For the annotation, a solution has to be found for problems like these because, if a structure like in sentence 62 is incorrectly annotated as coordinated chunks, then the correct annotation of two clauses is impossible.

(62) Der Trainer motiviert die Spieler und die Zuschauer sind begeistert.
The coach motivates the players and the spectators are delighted.
‘The coach motivates the players and the spectators are delighted.’

5.2 Easy-First-Parsing and a Divide-and-Conquer Strategy

The shallow parser consists of different specialized transducers, i.e. each transducer deals with one linguistic phenomenon or one limited group of linguistic phenomena. These transducers are arranged as a cascade: the output of preceding transducers is the input to the following ones. Consequently, apart from the rules themselves, the arrangement of the respective transducers in the cascade is crucial because this very order reflects the order of the annotation of the different linguistic phenomena. Two types of parsing are typically distinguished with respect to the order of the annotation of different linguistic phenomena: *top-down parsing* and *bottom-up parsing*. In top-down parsing, constituents which are closest to the root node are annotated first and then successively each layer of constituents immediately dominated by the previously annotated one. In bottom-up parsing, preterminal constituents are annotated first and then successively each layer of constituents immediately dominating the previously annotated one.

Figure 5.3 shows that we do not follow a strict bottom-up or top-down annotation strategy but rather a mixed strategy. At first, topological fields are annotated, which are neither pre-terminal nor immediately dominated by the root node. Then, clauses are annotated. They are only immediately dominated by the root node in the case of main clauses.[1] The last group of linguistic phenomena which are annotated are chunks. They are pre-terminal in some cases. Chunks are annotated both bottom-up and top-down. The reason why we are using this mixed order of annotation is that we are following an *easy-first parsing* strategy and a *divide-and-conquer* strategy for the annotation of shallow structures (and also deeper structures later on).

Easy-first parsing is a term coined by Abney (1996c) for a strategy in which "we make the easy calls first, whittling away at the harder decisions in the process". In Abney (1996c), this easy-first parsing strategy is applied bottom-up, i.e. first chunks and then simplex clauses are annotated. However, pre-terminal constituents like chunks are not necessarily the 'easiest' linguistic phenomena to be parsed and they are not necessarily those structures which contribute most to the following parsing process. This is the case for English, in which the chunk structure is subject to syntactic restrictions and reveals, to a certain extent, the clause structure. In German, however, the topological field structure is also subject to syntactic restrictions just like the chunk structure in English and German as figure 5.4 shows.[2] Since the topological field structure is an invariant pattern of the German language and since the pattern is very simple and its main component, the sentence bracket, just contains tokens with some few PoS tags, it is certainly the linguistic phenomenon which can be parsed most easily.

The sentence bracket, which divides the topological fields into VF, MF and NF typically just contains all verbal PoS tags and the subordinator in CFs in VL clauses.[3] In V1 and V2 clauses the left part of the sentence bracket contains just one verb, i.e. the finite verb, and, if there are any other verbal elements, they are contained in the right part of the sentence bracket in a strictly defined linear order. In VL clauses, all verbs are contained in the right part of the sentence bracket in a strictly defined order and the left part of the sentence bracket consist of the subordinator. Provided that PoS

[1]Since sentence segmentation has been done previously, there is always a node 'sentence' which dominates all other nodes including clause nodes (cf. figure 5.2).

[2]Figure 5.4 shows the same syntactic patterns of topological fields as figure 4.2. Our shallow annotation does, however, sometimes have other category labels.

[3]Only in few cases, the CF is complex. In most of those cases, attributive pronouns occur like in the sentence in figure 5.5. However, only 121 of 4437 CFs are complex in the TüBa-D/Z (i.e. approx. 0.03%).

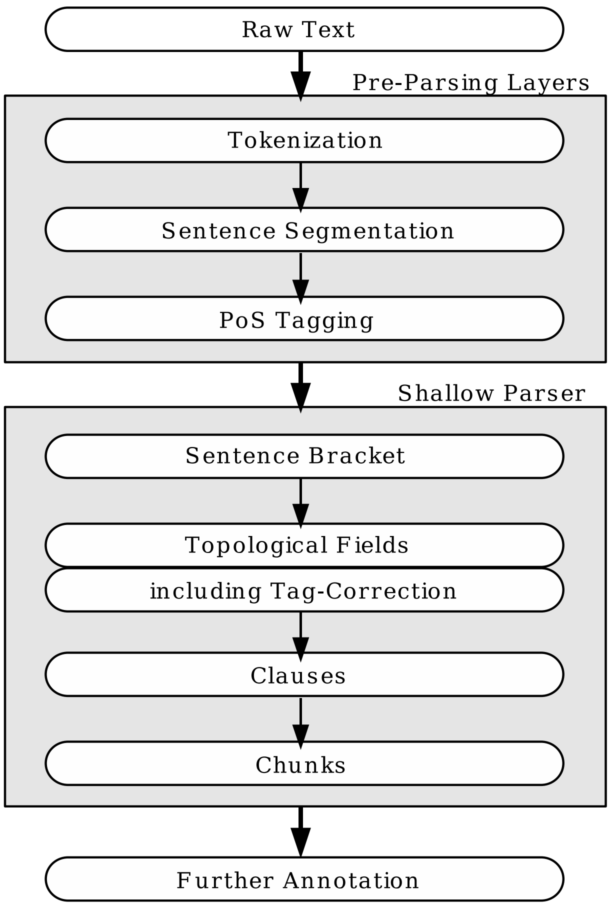


Figure 5.3: Pre-parsing and shallow parsing

clause type	topological fields						
VL:	KOORD	LV		**CF**	MF	**VCR_**	NF
V1:	KOORD	LV		**VCL_**	MF	VCR_	NF
V2:	KOORD/ PARORD	LV	**VF**	**VCL_**	MF	VCR_	NF
				left part of bracket		right part of bracket	

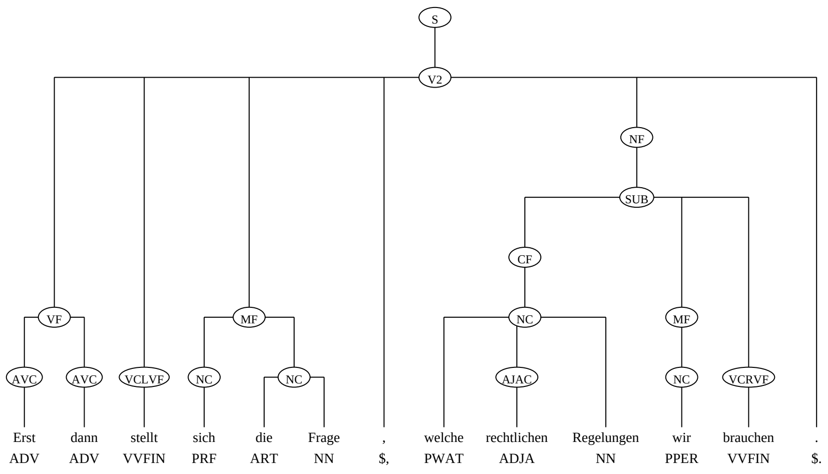
V1: verb-first clause	VF: initial field
V2: verb-second clause	MF: middle field
VL: verb-last clause	NF: final field
KOORDF: coordinator field	CF: complementizer field
PARORDF: non-coordinator field	VCL: left part of verb complex
LV: resumptive construction	VCR: right part of verb complex

Figure 5.4: Topological Fields

tag annotation is correct, only the annotation of the finite verb is ambiguous if it is the only element of the sentence bracket. In this case, it can be either the left part of the sentence bracket in V1 or V2 clauses or the right part of the sentence bracket in VL clauses. In this case, the punctuation and other immediate context information gives useful cues for the disambiguation. These facts show that the sentence bracket is the structure which can be annotated most reliably and which is, thus, annotated first following the easy-first parsing strategy. The topological fields VF, MF and NF can then be annotated afterwards. Since the topological field structure reveals the clause structure and type, clauses are annotated after topological fields. Chunks are annotated last. The direction of annotation is shown in figure 5.3 and exemplified in figure 5.6, which shows the annotation of the topological fields and clauses in sentence 63.[4]

(63)	Und	dann	hat	er	gesagt,	daß	er	mit	dem	Rad	schneller	ankommt
	And	then	has	he	said	that	he	with	the	bike	sooner	arrives

[4]N.B.: Chunks are annotated **after** the annotation of clauses and, thus, not dealt with in figure 5.6.



Erst dann stellt sich die Frage, welche rechtlichen
Not until then poses itself the question, which legal
Regelungen wir brauchen.
regulation we need.

‘Not until then does the question come up which legal regulation we need.’

Figure 5.5: Complex CF

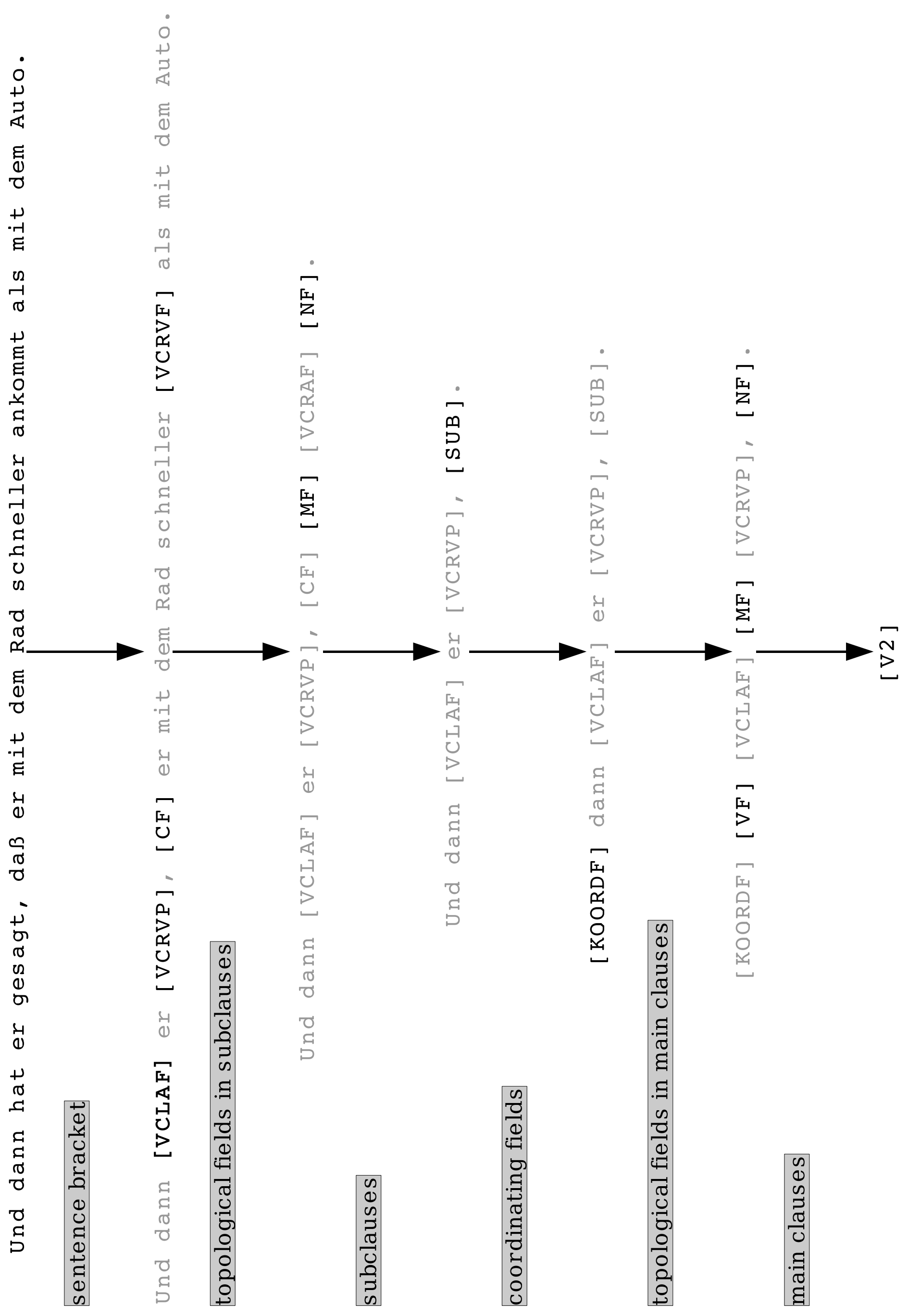

Und dann hat er gesagt, daß er mit dem Rad schneller ankommt als mit dem Auto.
sentence bracket
Und dann [VCLAF] er [VCRVP], [CF] er mit dem Rad schneller [VCRVF] als mit dem Auto.
topological fields in subclauses
Und dann [VCLAF] er [VCRVP], [CF] [MF] [VCRAF] [NF].
subclauses
Und dann [VCLAF] er [VCRVP], [SUB].
coordinating fields
[KOORDF] dann [VCLAF] er [VCRVP], [SUB].
topological fields in main clauses
[KOORDF] [VF] [VCLAF] [MF] [VCRVP], [NF].
main clauses
[V2]

Figure 5.6: Illustration of the succession of the different annotation levels

als	mit	dem	Auto.
than	with	the	car.

'And then he said that he will be faster by bike than by car.'

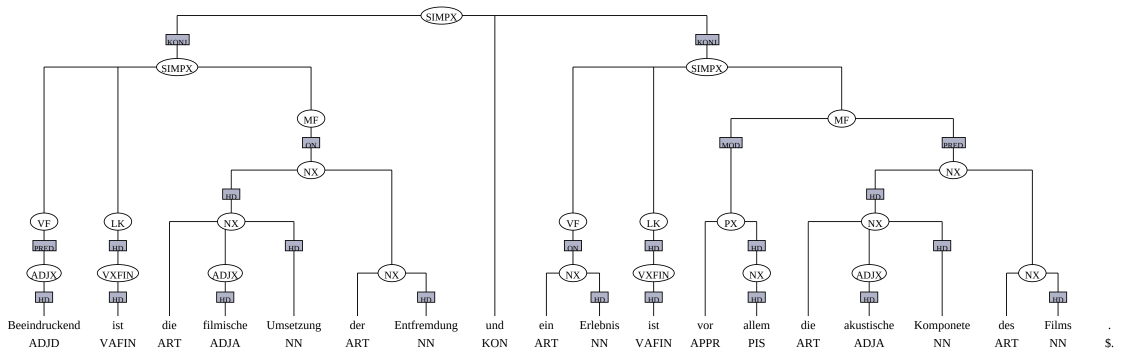
Figure 5.6 shows how the topological field and clause structure is annotated successively.[5] At first, the sentence bracket is annotated. This can be achieved for subordinate clauses and main clauses in one step since no element of the sentence bracket can be contained in any other element of the sentence bracket. The next step is the annotation of the topological field structure in the subordinate clause. This must be kept separate from the recognition of topological fields in the main clause because topological fields in main clauses may contain subordinate clauses. In the following step, the topological field structure of the subordinate clause is recognized as the subordinate clause SUB. After this, coordinating fields are annotated. Then, the topological field structure of the main clause is annotated, which includes the subordinate clause SUB into its NF. After this, the main clause can be annotated as a V2 clause.

The annotation of topological fields as the first linguistic phenomenon in the annotation cascade does not only follow an easy-first parsing strategy but also a divide-and-conquer strategy. In a divide-and-conquer strategy as described in Peh and Ting (1996), a complex clause is not annotated in one step but subordinate clauses are annotated first. The annotation task is, thus, divided into smaller portions which can then be tackled individually. This strategy has the advantage that attachment and coordination ambiguities are reduced because many attachments and coordinations cannot reach beyond the clause level or, in German, beyond the level of one topological field. The advantage of this strategy is higher accuracy of the parser and also higher speed since the scope of ambiguity the parser has to deal with has been reduced.

The sentence in figures 5.7 (TüBa-D/Z) and 5.8 (shallow annotation) shows the advantage of the divide-and-conquer strategy. There are two coordinated sentences. However, without the knowledge of the clause boundaries, the string 'der Entfremdung und ein Erlebnis' could just as well be two coordinated chunks.[6] Only after the annotation of topological fields and clauses,

[5]The structure added at each level is highlighted.

[6]N.B.: This reading could, in this special case, be ruled out by morphological constraints because, typically, only chunks with the same morphological features can be coordinated. We have, however, for reasons of efficiency decided not to use morphological



Beeindruckend ist die filmische Umsetzung der Entfremdung und
Impressing is the cinematic realization of the alienation and
ein Erlebnis ist vor allem die akustische Komponente des
an experience is before all the acoustic component of the
Films.
film.

'The cinematic realization of the alienation is impressing and, most of all, the acoustic component of the film is a great experience.'

Figure 5.7: Two coordinated V2 clauses from TüBa-D/Z

it is clear that there are two coordinated clauses and not two coordinated chunks. Thus, the annotation of topological fields as the skeleton of the clause before chunking prevents the wrong annotation of coordinated chunks which would completely obstruct the correct annotation of the whole sentence.

Figure 5.8, which shows the shallow annotation, furthermore illustrates the effect of the divide-and-conquer strategy: Apart from the fact that the coordination ambiguity of some chunks is resolved, the attachment ambiguity of grammatical functions is now confined since the grammatical functions of, e.g., one verb are now restricted to one clause. That way, the search space for the parser has been considerably reduced, which makes the parser quicker and the rules for the annotation less complicated and more easily manageable and, thus, more reliable.

---

features in shallow annotation.

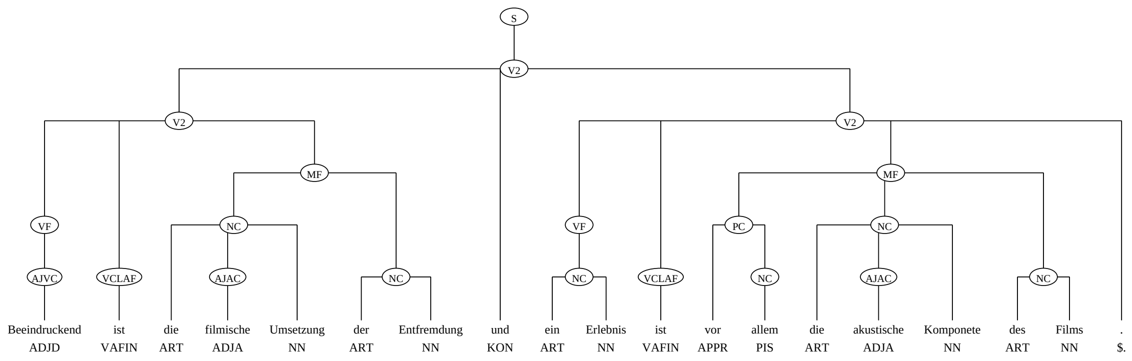


Figure 5.8: Two coordinated V2 clauses from TüPP-D/Z

5.3 Correcting Tags by Using Topological Field Information

There is another strategy (described in Müller and Ule (2002)) which uses topological field and clause structure to resolve local ambiguities which would have lead to wrong parsing results in the whole clause, otherwise. Since standard taggers typically just take into account the local context of a token, some tokens may be assigned the wrong PoS tag, especially if their syntactic dependency is not expressed by linear proximity but by linear distance, i.e. the tokens belonging together syntactically are not close together in the linear order of elements. In German, this is especially the case in the sentence bracket: The verb complex can be split up such that the finite verb is close to the beginning of the clause and the non-finite parts of the verb complex (i.e. the infinitive or participle) are at the end of the clause. This in contrast to the noun phrase, in which the syntactic dependency of determiner, adjective and noun is expressed by linear proximity. Consequently, more errors occur in the PoS tagging of elements if their syntactic dependency is expressed by linear distance.

Since the form of the infinitive and the 1st and 3rd person plural form of the finite verb are homonyms, a tagger which has to rely on local context would have problems with the distinction between the infinitive verb in sentence 64 and the finite verb in sentence 65 because the token which reveals the function of the verb ‘benennen’ is in the left part of the sentence bracket which is twelve tokens apart from the verb ‘benennen’ in the right part of the sentence bracket in sentence 64 and by fourteen in sentence 65. We have shown in Müller and Ule (2002) that it is in fact the case that the tag VVFIN has a higher error rate. The tag VVFIN (finite lexical verb) has an error rate of 9.95% and the tag ADJA (attributive adjective) which is also a high-frequent open class PoS tag,[7] has an error rate of 2.40%.

(64) Bis Mitte Juli [$_{\text{VCLMF}}$ will] [$_{\text{MF}}$ der Verwaltungsausschuß für das
Until middle July wants the administrative board for the
Oldenburgische Staatstheater dem Kulturministerium in Hannover
Oldenburgian state theatre the ministry of culture in Hannover
seinen Wunschkandidaten] [$_{\text{VCRV}\mathbf{I}}$ benennen].
its favourite candidate denominate.

'The administrative board of the Oldenburg State theatre wants to disclose the ministry of culture in Hannover its favourite candidate until the middle of July.'

(65) Der Ministerpräsident will, [$_{\text{CF}}$ daß] [$_{\text{MF}}$ die Mitglieder des
The prime minster wants that the members of the
Verwaltungsausschusses für das Oldenburgische Staatstheater dem
administrative board for the Oldenburgian state theatre the
Kulturministerium in Hannover ihren Wunschkandidaten] [$_{\text{VCRV}\mathbf{F}}$
ministry of culture in Hannover its favourite candidate
benennen].
denominate.

'The minister wants the members of the administrative board for the Oldenburgian state theatre to disclose their favourite candidate to the ministry of culture in Hannover.'

Figure 5.9, which is taken from the TüBa-D/Z, shows the shallow structure of sentence 64. The sentence is a V2 clause with the finite verb in the left part of the sentence bracket (VCLMF) and the infinitive verb in the right part (VCRVI). Since the verb 'benennen' could be both an infinitive or a finite verb as regards its form, a structure like this is very vulnerable to tagging mistakes. We use the annotation of the topological field structure to deal with this problem. After the elements of the sentence bracket have been annotated, we try to fit the sequence of PoS tags and constituents of

[7]The tag ADJA accounts for 5.7% of all tokens and the tag VVFIN accounts for 4.2% of all tokens.

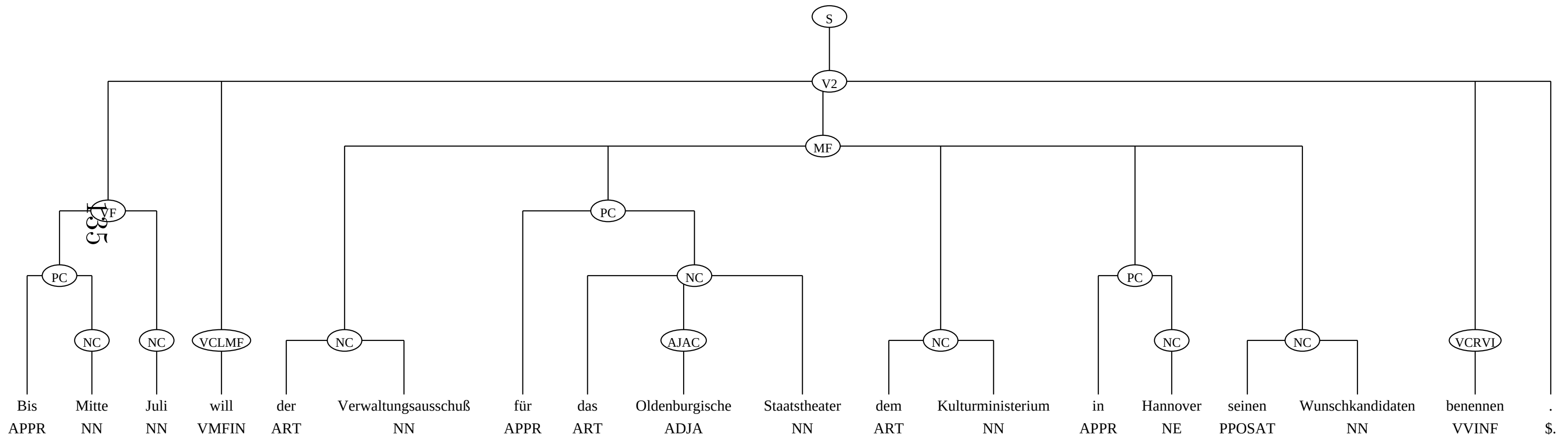


Figure 5.9: V2 clause with ambiguous verb in right sentence bracket

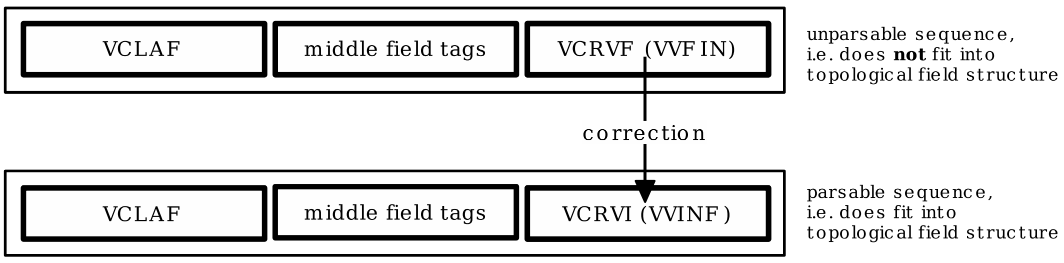


Figure 5.10: Correction of an unparsable sequence

the sentence bracket into a parsable sequence to annotate topological fields like the MF. If no such sequence can be found, i.e. if there is no **parsable** sequence, the parser tries to match the sequence with a constituent of the sentence bracket corresponding to another tag from the PoS tag ambiguity class of the relevant token (i.e. the class of potential PoS tags for the token). If it is then possible to fit the sequence into a parsable structure, the PoS tag of the respective token is changed and also the constituent label of the sentence bracket.

The process can be exemplified by the sentence in figure 5.9. If the tagger had assigned the token 'benennen' the wrong tag VVFIN, the component annotating topological fields could not have assigned the sentence any topological field structure because it would not have fit into any parsable sequence. The topological field MF can only be assigned in a V2 clause if the left part of the sentence bracket contains the finite verb and the right part of the sentence bracket the non-finite parts of the verb complex. If 'benennen' had been assigned the tag VVFIN, this would not be the case. The tag correction component would then change the tag from VVFIN into VVINF and the label of the right part of the sentence bracket from VCRVF to VCRVI (cf. figure 5.10). That way the parts of the sentence bracket, which were annotated just taking into account local context and PoS tags, can be corrected by taking into account topological field structure.

The problem of ambiguous tokens in the sentence bracket becomes even more obvious in the case of ambiguity between subordinators and other tags. In those cases, the wrong annotation of a token would completely obstruct further annotation of the clause in question and, in most cases, also of the embedding clause. An example of this phenomenon is the token 'als', which can be either a *Vergleichspartikel* (particle of comparison; KOKOM) or a subordinator (KOUS) and the token ‘seit’ which can be either a preposition (APPR) or a subordinator (KOUS). Sentences 66 and 67 give examples for the token ‘seit’: In sentence 66 the token ‘seit’ functions as a preposition (cf. figure 5.11 for the complete shallow structure), in sentence 67 it functions as subordinator (cf. figure 5.12). Again, there is no way to locally disambiguate between the two readings; but since we annotate topological fields before chunks, we can, again, apply the strategy to use the topological field structure to disambiguate structures which are locally ambiguous but which may be resolved by using overall clause structure.

(66)

[$_{VF}$ [$_{PC}$ Seit/APPR	dem	Beginn]	des	Krieges	am	24.	März]
Since	the	beginning	of the	war	on	24th	March

[$_{VCLAF}$ hat]	[$_{MF}$ die	Nato	nicht	so	viele	Fehltreffer]	[$_{VCRVP}$
has	the	NATO	not	so	many	target misses	

gelandet].	
touched	down.

‘Never since the beginning of the war on March 24th has the NATO missed their targets so often.’

(67)

[$_{SUB}$ [$_{CF}$ Seit/KOUS]	[$_{MF}$ Nato-Militärs	am	24.	März	die
Since	NATO soldiery	on	24th	March	the

internationalen	Bemühungen	zur	Lösung	des	Kosovo-Konflikts]
international	efforts	for the	solution	of the	Kosovo conflict

[$_{VCRAFVP}$ übernommen	haben]],	sind	zivile	Politikformen	in	den
taken over	have,	are	civilian	politics forms	in	the

Hintergrund	gerückt.
background	moved.

‘Since NATO soldiery have taken over the efforts for the solution to the Kosovo conflict on March 24th, civilian forms of politics have been moved to the background.’

As regards the distinction between the preposition reading and the subordinator reading of ‘seit’, the principle of the mechanism is the same as with the distinction between the finite and infinitive reading of the verb. First, the sentence bracket is annotated. Then, we try to fit the structure into a topological field structure. For sentence 66 this would mean that, in the case that ‘seit’ had been annotated as a KOUS, there would be an unparsable sequence, i.e. ‘CF VCLAF VCRVP’. Since the sequence ‘VCLAF VCRVP’

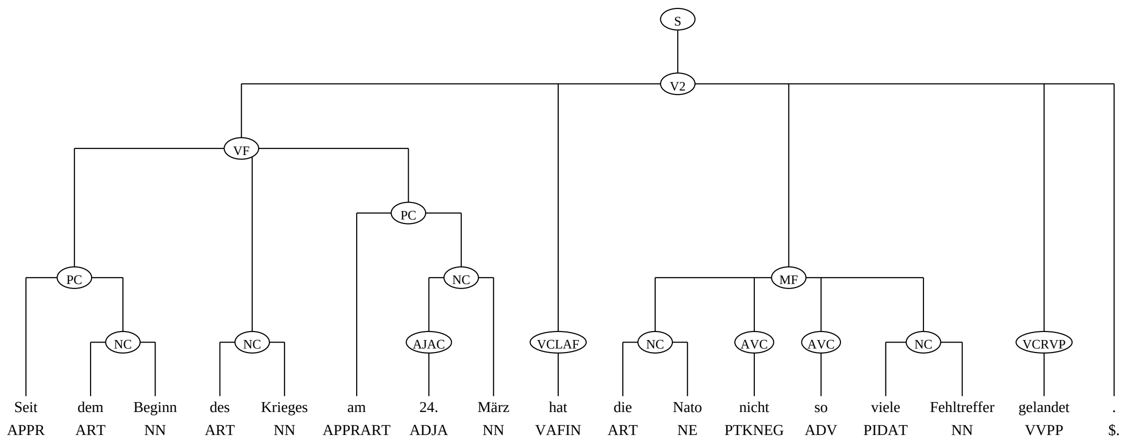


Figure 5.11: The token ‘seit’ as a preposition

would, however, be a parsable sequence, the tag of ‘seit’ would be be changed into APPR because this tag fits into a parsable sequence. The same also applies the other way round as can be shown with sentence 67 (cf. figure 5.12): If the token ‘seit’ is not correctly PoS-tagged as KOUS, then there is just the right part of the sentence bracket, which cannot occur up alone if it is finite. Thus, if there is a token which can potentially be a subordinator like ‘als’, ‘bis’, ‘da’ or ‘seit’, its PoS tag is changed to KOUS and it is annotated as the CF, which is the left part of the sentence bracket. After this, the subordinate clause can be annotated, which would otherwise have been impossible.

The strategy with which the tagging errors are detected and corrected in our parser is similar to the ones described in Hirakawa, Ono, and Yoshimura (2000) and Rösner (2000). Hirakawa et al. (2000) describe a system which uses already existing linguistic knowledge from an MT system in order to learn PoS disambiguation rules from an automatically PoS-annotated corpus. The tagger is seen as a PoS candidate generator, which produces a ranked ambiguity class. After tagging, the text is parsed. For those cases where the top-ranked PoS candidate does not fit into any of the parsable PoS sequences, another candidate is chosen which fits into the *Most Preferable Parsable POS sequence*. Learning from those instances where a PoS tag is changed, the system is able to learn *POS Adjusting Rules*, which are used to improve the tagger.

Unlike Hirakawa et al. (2000), Rösner (2000) does not work with ranked

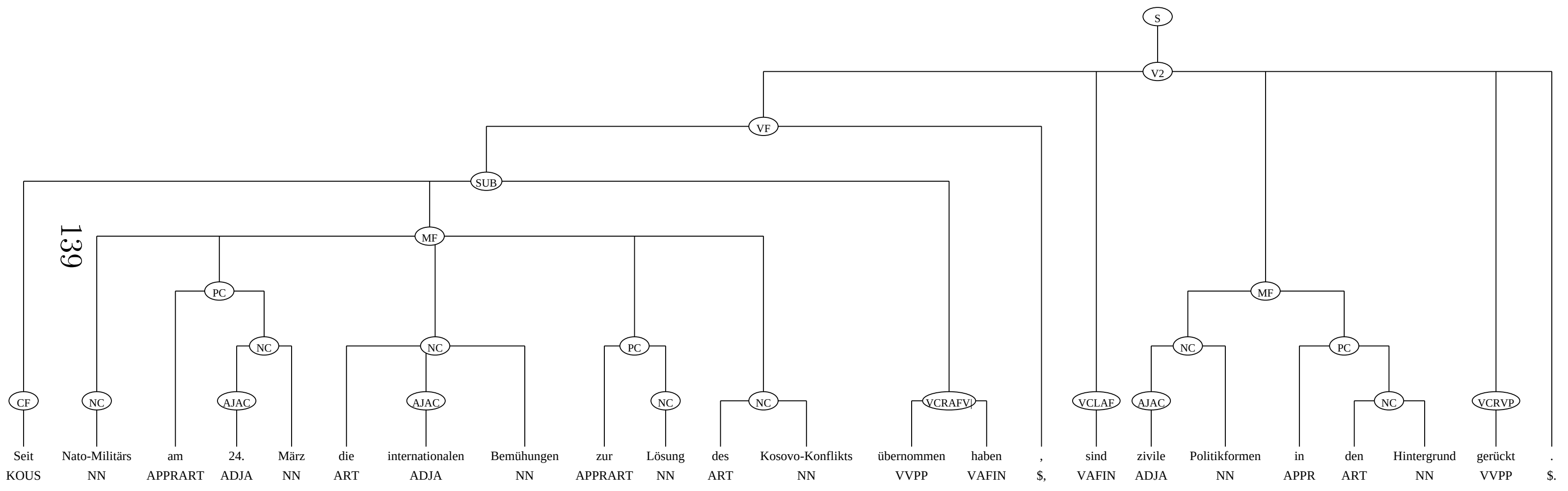


Figure 5.12: The token 'seit' as a subordinator

ambiguity classes. Tags which are morphologically ambiguous to the PoS tagger [8] are assigned to an unordered ambiguity class; tags which are not recognized by the PoS tagger are simply marked as unknown. However, the parser can deal with both situations and either elects a tag from the ambiguity class which fits into a parsable sequence or inserts the proper tag in those cases where Morphix has not assigned any tag. Thus, both systems use the linguistic knowledge of a parser to improve tagging accuracy. The difference between them is that the system of Hirakawa et al. (2000) is not integrated into the parser but merely a system to improve a tagger, although the resulting rules may well be integrated into a parser. The system described in Rösner (2000), on the other hand, does not use the concept of ranked ambiguity classes, which means that all PoS tags are marked as equally likely.

5.4 Handling Recursion by Iterating FSAs

We define recursion as the capability of a rule to be applied to itself.[9] One example is the sample definition of an NP as a string of a determiner, an adjective and a noun followed by a PP which itself is defined as a preposition followed by an NP (cf. 5.13). In principle, this rule can be applied infinitely because it can apply itself again and again. This is one of the reasons why recursion is not handled in shallow parsing (excluding the few exceptions discussed in section 4). In shallow parsing, only the rule without the postmodifying PP is applied. Recursive structures are, thus, left partially ambiguous in shallow structures; their attachment is left unspecified. This can be seen in figure 5.14 which shows the deep structure and in figure 5.15 which shows the shallow structure of a postmodifying PP: The phrase 'die geplante Vertragsverlängerung mit Doll' is annotated as one phrase (NX) in figure 5.14; the NX in it is marked as the head, which leaves the PX as a modifier. Modification is not marked in figure 5.15, where both the NC and the PC are directly attached to the VF.

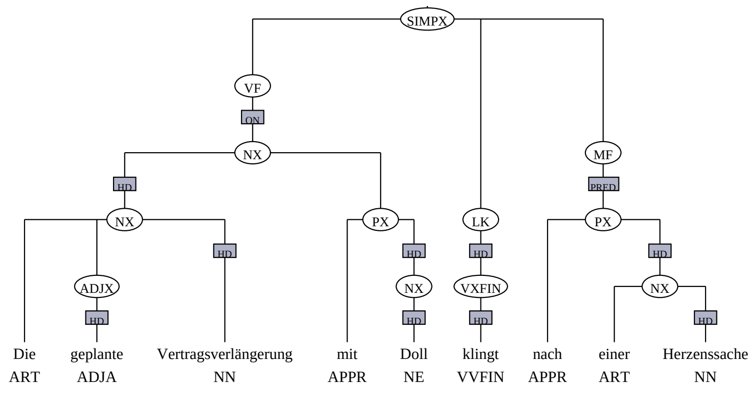
There are, however, cases in which we have decided to cover recursion because, otherwise, the shallow structures are no longer a useful basis for further annotation. Center-embedding in NCs is one of those cases. Without

[8] The PoS tagger Morphix (Finkler and Neumann, 1988) is used.

[9] This means that we do not define chunks which are contained in other chunks as recursive structure as long as the chunks are not chunks of the **same** category.

```
NP --> article adjective noun PP
PP --> preposition NP
```

Figure 5.13: Recursion in the rule for NPs



Die	geplante	Vertragsverlängerung	mit	Doll	klingt	nach	einer
The	planned	contract extension	with	Doll	sounds	like	a
Herzenssache							
heart's desire							

'The planned contract extension with Doll sounds like a heart's desire'

Figure 5.14: Post-modifying PP in TüBa-D/Z, i.e. deep structure

the covering of center-embedding in NCs there is no straightforward way to generate the deep structure from the shallow structure (cf. section 4.2.1) which was one of our requirements for shallow structures. However, the same applies to center-embedded subordinate clauses.[10] Subordinate clauses following the constituent they refer to may be left unattached like PCs are left unattached to the NC they refer to (cf. sentence 68 shown in figure 5.16). However, not annotating center-embedded subordinate clauses would render impossible the correct annotation of the whole clause. This can be seen in sentence 69 shown in figure 5.17: If just the embedded subordinate clause (SUB) 'als wir viel Geld hatten' was annotated and the embedding clause was not annotated, only the first five tokens of the sentence could be fit into

[10]N.B.: All kinds of embedding solely apply to **subordinate** clauses because the very definition of subordinate clauses is that they are embedded into another clause.

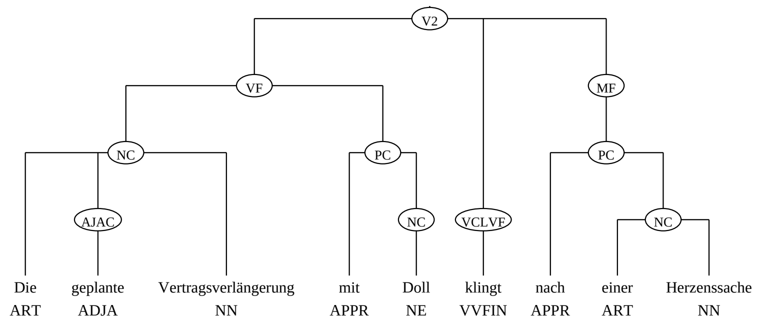


Figure 5.15: Post-modifying PP in TüPP-D/Z, i.e. shallow structure

a meaningful structure and the rest of the constituents of the sentence would be left ‘stranded’.

(68) Hintergrund ist die Überbelegung der Berliner
Background is the overcrowding of the Berlinian
Justizvollzugsanstalten, die zum Teil auch daher
correctional facilities, which to a certain extent also from this
rührt, daß Haftstrafen angetreten werden, um Geldstrafen
comes that imprisonments accepted are to money sentences
abzusitzen.
serve.

‘The background of this is the overcrowding of the Berlin prisons which is caused, to a certain extent, by the fact that people accept prison sentences to do their fines.’

(69) Das Defizit liegt eher darin, daß der Kultursektor auch in
The shortcoming lies rather in that that the culture sector also in
Zeiten, als wir viel Geld hatten, im Bundesvergleich zu
times, when we a lot of money had in federal comparison too
niedrig finanziert worden ist.
low financed been has.

‘The shortcoming rather lies in the fact that – even in days when we did have a lot of money – the cultural sector has been financed too little compared to the national level.’

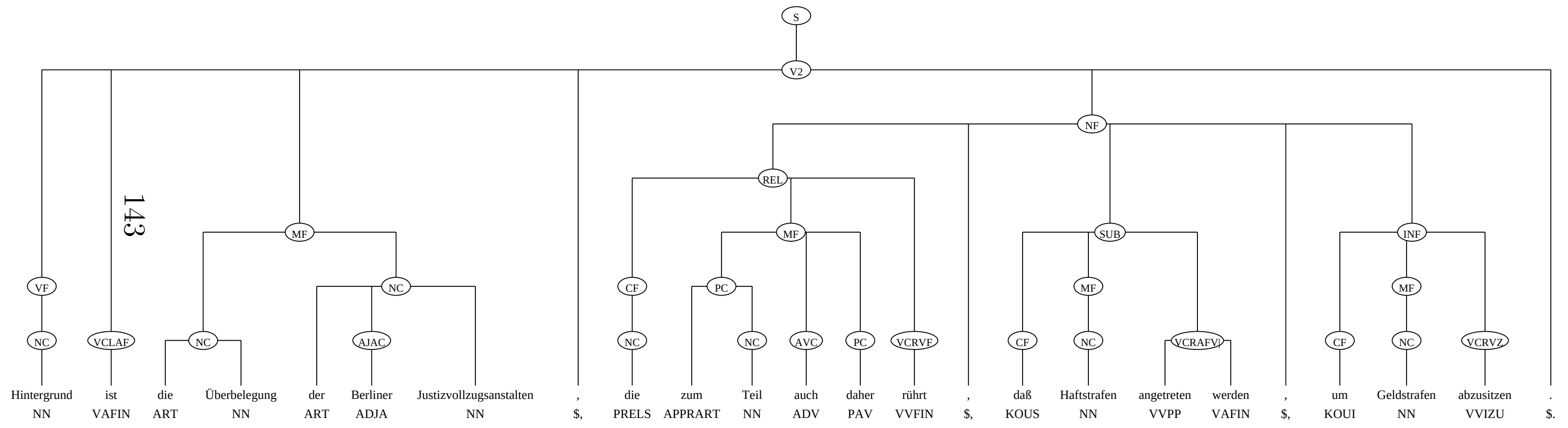


Figure 5.16: Subordinate Clauses occurring successively

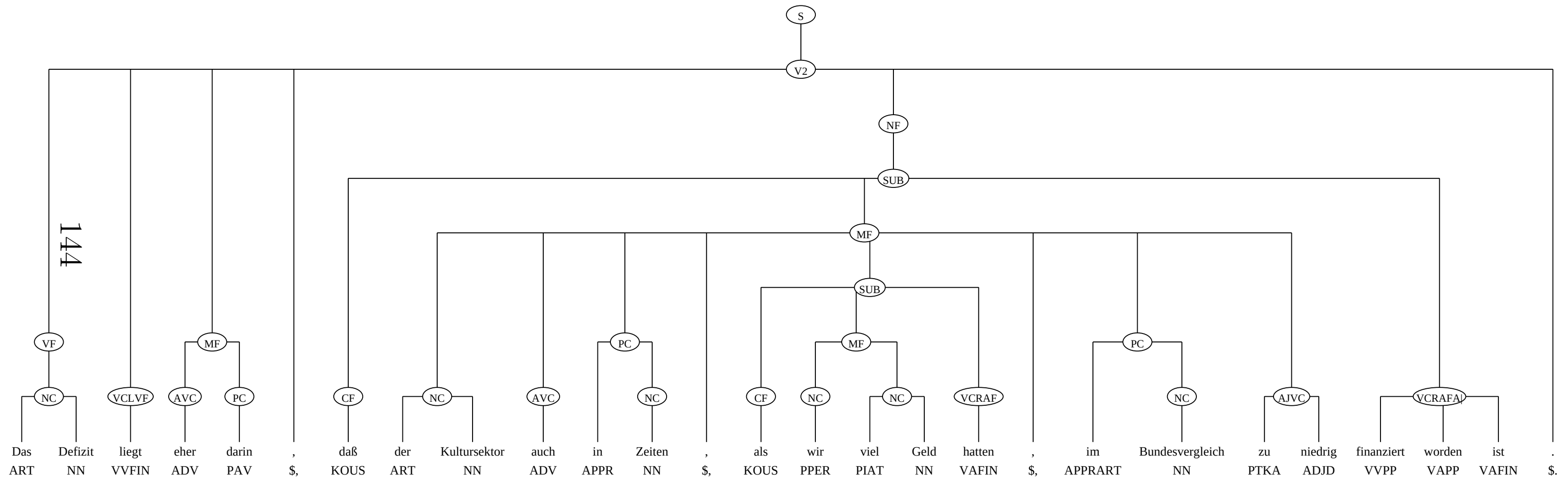


Figure 5.17: Subordinate Clause center-embedded into subordinate clause

Having motivated the need for the annotation of recursive structures in center-embedded clauses, there remains the problem that FSAs cannot deal with recursion. It is, however, possible to iterate the FSAs. Thus, in the cascade of FSAs, the ones which annotate subordinate clauses and their respective topological fields can be called more than once to cover recursion. With respect to the question of how often the FSAs have to be iterated, we have decided to choose two passes because more than one center-embedding hardly ever occurs. The reason for this might be cognitive factors which limit the processing of such structures as Schefe (1975) has pointed out. The TüBa-D/Z does not contain any subordinate clauses which are center-embedded into subordinate clauses which are themselves center-embedded into subordinate clauses (i.e. subordinate clauses in the MF of a subordinate clause which is in the MF of another subordinate clause). Such structures are, however, possible as the constructed sentence 70 shows although, in this case, the subordinate clauses are of different categories and the phrase 'der, wenn er den Mund aufmacht, lügt' seems solidified; this might make processing easier. Should the evaluation of the parser show considerable problems with double center-embeddings, the subordinate clause component could be applied three times.

(70)	Wenn	Du	Deinen	Freund,	der,	wenn	er	den	Mund	aufmacht,	lügt,
	If	you	your	friend	who	when	he	the	mouth	opens	lies
	mitbringst,	geh	ich.								
	take along	go	I.								

'If you bring along your friend who lies when he opens his mouth I'll leave.'

Figure 5.18 shows the way the annotation of center-embedded subordinate clauses works for a slightly simplified version of sentence 69: First, the elements of the sentence bracket are annotated. Since these are annotated mainly on the basis of their inner structure or their local context,[11] they can be be annotated for all clauses before further annotation takes place, even in cases of center-embedding. The following step assigns topological fields in subordinate clauses. Only the MF in the clause 'als wir viel Geld hatten' can be annotated in this step because the sentence bracket of that very clause blocks the annotation of the MF it is embedded in. In the following step, the topological field structure of the clause is matched as a subordinate clause

[11]With the exception of the cases concerning the tag correction component, in which global clause structure is taken into account.

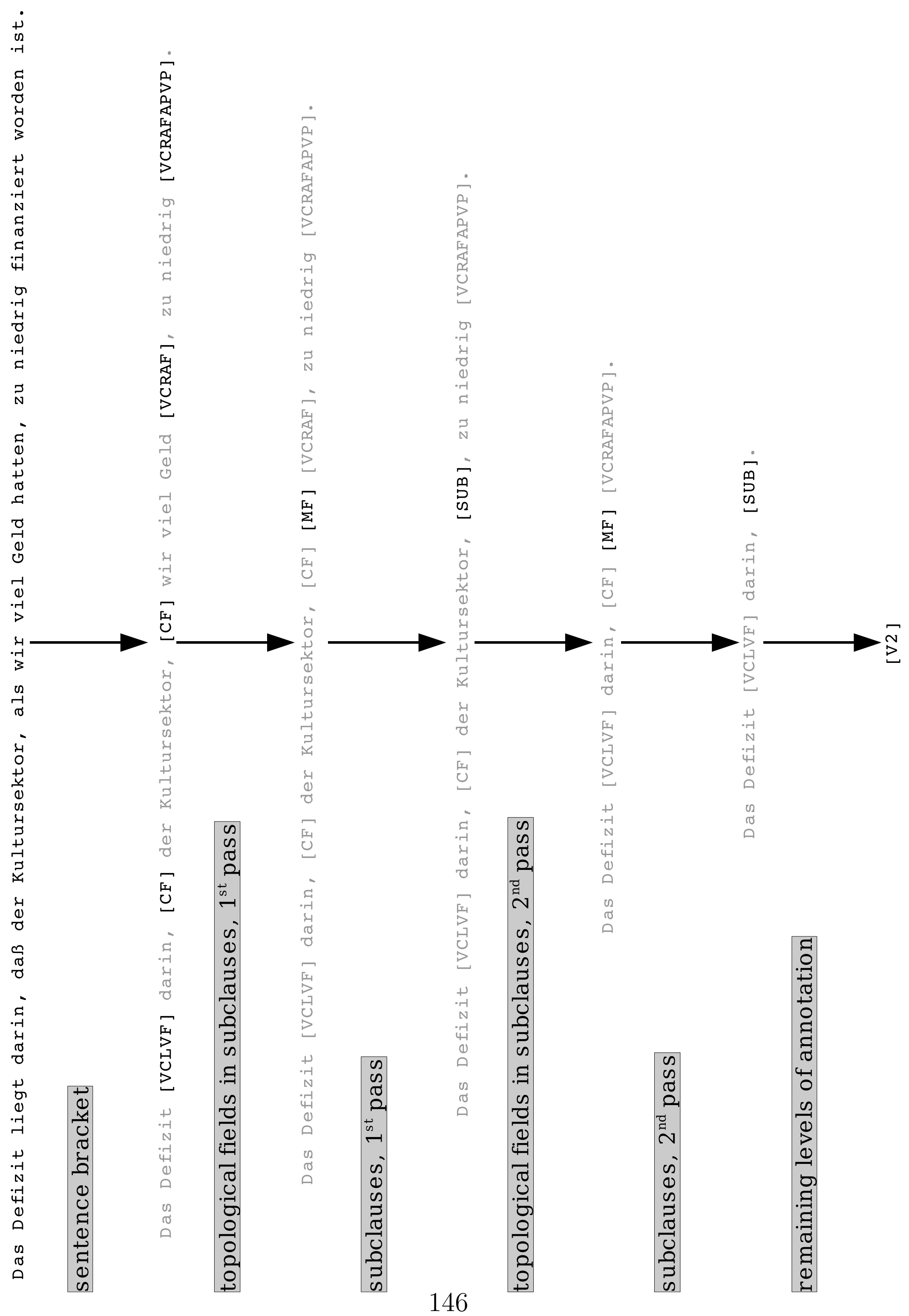

Das Defizit liegt darin, daß der Kultursektor, als wir viel Geld hatten, zu niedrig finanziert worden ist.
sentence bracket
Das Defizit [VCLVF] darin, [CF] der Kultursektor, [CF] wir viel Geld [VCRAF], zu niedrig [VCRAFAPVP].
topological fields in subclauses, 1st pass
Das Defizit [VCLVF] darin, [CF] der Kultursektor, [CF] [MF] [VCRAF], zu niedrig [VCRAFAPVP].
subclauses, 1st pass
Das Defizit [VCLVF] darin, [CF] der Kultursektor, [SUB], zu niedrig [VCRAFAPVP].
topological fields in subclauses, 2nd pass
Das Defizit [VCLVF] darin, [CF] [MF] [VCRAFAPVP].
subclauses, 2nd pass
Das Defizit [VCLVF] darin, [SUB].
remaining levels of annotation
[V2]

Figure 5.18: Illustration of the annotation of center-embedded subordinate clauses

SUB. After this step, the FSAs for the annotation of subordinate clauses are applied a second time. The MF of the clause 'daß der Kultursektor zu niedrig finanziert worden ist' can now be annotated since a subordinate clause SUB is allowed in the MF of a clause and the annotation is, thus, no longer blocked. This makes the annotation of the embedding subordinate clause SUB possible and the whole center-embedded structure is annotated.

5.5 Annotating Chunks Top-down

Chunks[12] are annotated after the annotation of topological fields and clauses.[13] The annotation is achieved top-down. The reason for this is that the structure of chunks adheres to a principle sometimes compared to that of the sentence bracket and, therefore, called the *nominal bracket* (*Nominalklammer*; (cf. Bußmann, 2002). While, for instance, in the VL clause, the CF introduces the clause and the verb complex ends it, in the NC, the determiner introduces the NC and the noun ends it. This phenomenon can, in principle, also be found in English. It does, however, only apply to chunks since it does not include post-modifying structures like PP-attachment.

We can, thus, define an NC as reaching from the determiner to the respective head noun. Provided that PoS tagging is correct, an NC can consequently be annotated by matching the string from a determiner to the nearest following noun in linear order. A PC can be defined accordingly as reaching from the preposition to the head noun of the dominated NC. There are, however, cases in which there is no determiner; but we can define an NC more general as the string which reaches from the marker of the beginning of an NC to the head noun. A similar approach is used by Aït-Mokhtar and Chanod (1997a) for French. A marker could be a determiner but also an attributive adjective or cardinal. In the latter cases, the NC can be matched as the string reaching from the leftmost attributive adjective or cardinal (which can also serve as an attributive adjective) to the nearest following noun in linear order. In the case that none of these pattern applies because the noun is not modified, the pure noun can be matched as an NC.

This strategy makes use of the fact that the linear order of constituents

[12]Cf. table 4.1, repeated here for convience, for an overview of the chunks.

[13]Verb chunks are an exception: They are annotated as the first constituents in the shallow parsing component because they make up the sentence bracket and can, consequently, be seen as both chunks and topological fields.

Chunk Label	Definition
AJAC	attributive adjective chunk
AJACTRUNC	AJAC with truncated adjective
AJACC	at least two coordinated AJACs
AJVC	predicative adjective chunk/adverb chunk
AJVCTRUNC	AJVC with truncated adjective/adverb
AJVCC	at least two coordinated AJVCs
AVC	adverb chunk
AVCC	coordinated AVC
NC	noun chunk
NCTRUNC	NC with truncated noun
NCC	at least two coordinated NCs
NCell	elliptical noun chunk (i.e. without head noun)
PC	prepositional chunk
VCTRUNC	verb chunk with truncated verb
VCL_	verb chunk as left part of sentence bracket
VCR_	verb chunk as right part of sentence bracket
VCF_	verb chunk in topicalized (fronted) position
VCE_	verb chunk containing finite verb in *Ersatzinfinitiv* structure

Table 5.1: Overview of the chunk labels

PC			
preposition	NC		
	determiner	adjective	noun

Figure 5.19: Linear order in PC and NC

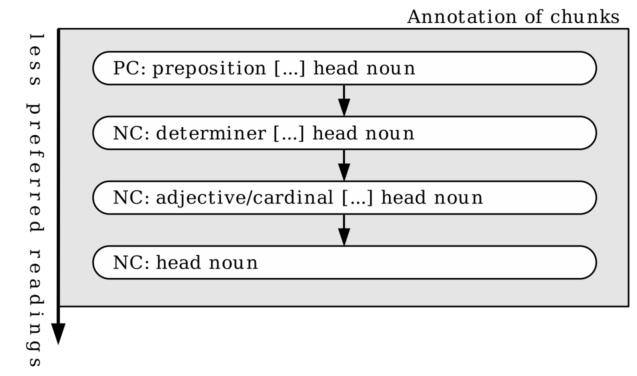
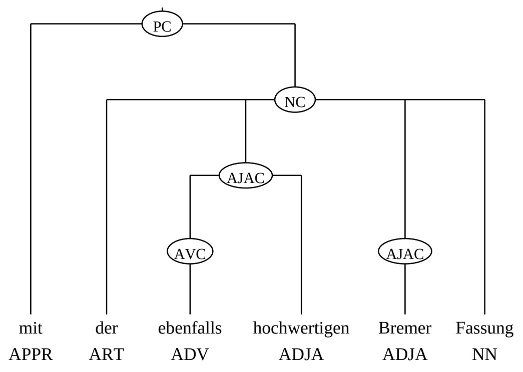

Annotation of chunks
less preferred readings
PC: preposition [...] head noun
NC: determiner [...] head noun
NC: adjective/cardinal [...] head noun
NC: head noun

Figure 5.20: A longest-match strategy for the annotation of chunks

within NCs is in a strictly defined order (by contrast to the linear order of chunks within the topological fields). The three constituents determiner, adjective/cardinal and noun are always in this linear order. If we also take into account the preposition, which is not part of the NC but shows its beginning, we have three markers which can be used for the detection of the beginning of an NC. Since, if one of the markers occurs, it has to be matched, we apply a *longest-match* strategy for the disambiguation of ambiguous structures, i.e. the reading in which the NC stretches from the determiner to the head noun is preferred to the reading in which it stretches from the attributive adjective to the head noun (and so forth as illustrated in figure 5.20). If two markers occur, the leftmost of the two is chosen as the beginning of the chunk.

For the annotation of chunks, we have decided in favour of this top-down approach which applies a longest-match strategy in conjunction with chunk boundary markers because it has some advantages over a bottom-up approach: Our approach is capable of matching even complex structures like the one in figure 5.21[14] in one step. The inner structure of the chunk can then

[14]The gloss shows a wider context for better understanding.



seit	dem	Vergleich	mit	der	ebenfalls	hochwertigen	Bremer	Fassung
since	the	comparison	with	the	also	high-class	Bremian	version

‘since the comparison with the version in Bremen, which is also high-class’

Figure 5.21: Complex AJAC in NC

be annotated afterwards. In the case of figure 5.21, a bottom-up approach would not have been able to straightforwardly generate the desired output since, at the point of the annotation of the AJAC ‘ebenfalls hochwertigen’, the adverb ‘ebenfalls’ could not have been attached because the attachment of adverbs is ambiguous. Only at the point of the annotation of the NC would the attachment of the adverb be unambiguous. There would then be a need for a special step to attach the adverb inside the AJAC. Problems like these are avoided using a top-down approach.

Another advantage is the straightforwardness of the approach. This becomes apparent in figure 5.20: For the annotation of PCs and NCs we basically need four simple rules: One to match PCs, one to match NCs introduced by a determiner, one for NCs introduced by an adjective/cardinal and one for unmodified NCs. The constituents between the chunk boundary marker and the head noun can be matched by an ANY expression which includes all constituents which may appear inside an NC. The sequence of the constituents within the NC is not defined in the rule. In a bottom-up approach, it would in principle be possible to match all NCs with one rule. However, the rule would be very complex since it would have to work with a large number of optional constituents at various positions. In order to match, e.g., adverbs like ‘ebenfalls’ in figure 5.21, one would have to insert an optional AVC at every potential point of attachment, e.g. after the preposition, after the determiner or after the adjective.

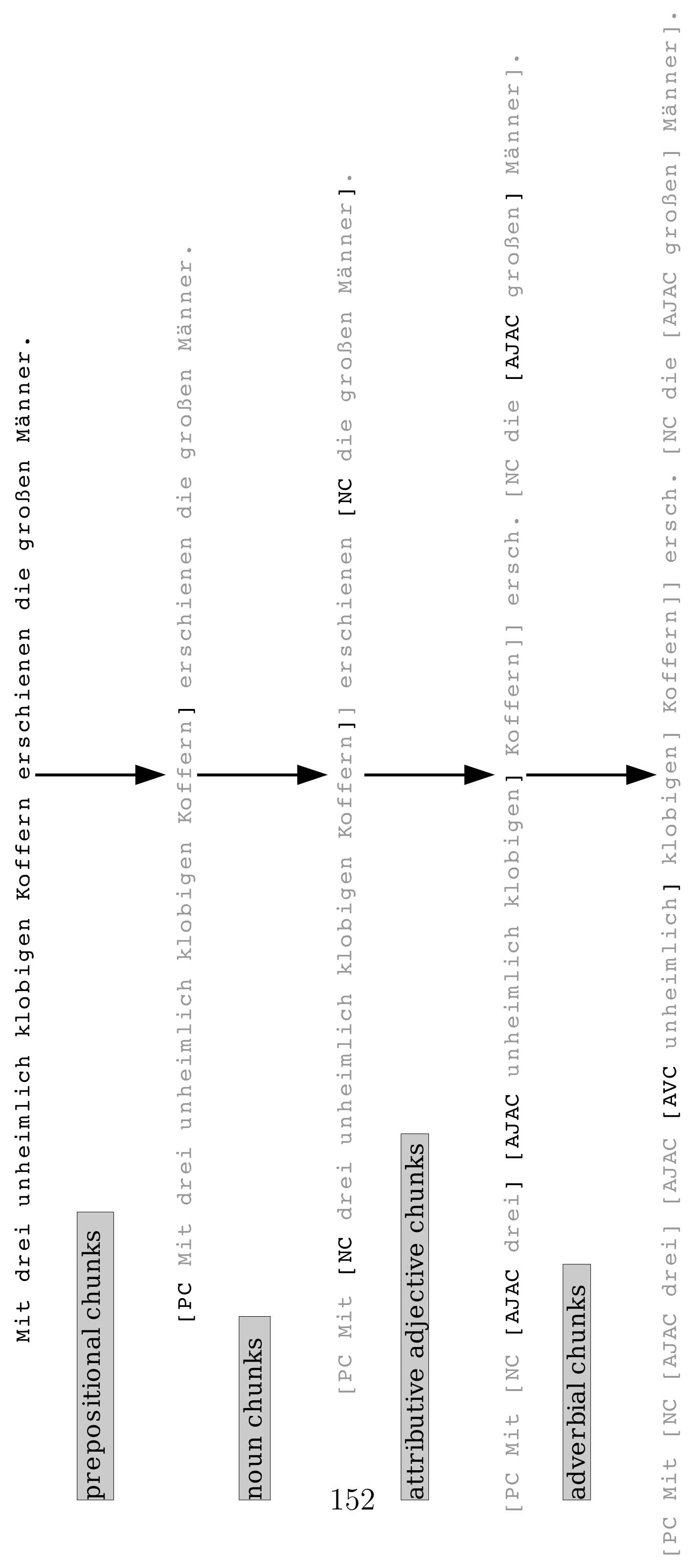
What is more is that a sequence of none or more AJACs could not be matched by a Kleene ‘+’ because an AVC could occur **between** two AJACs. If, in figure 5.21, the second AJAC was modified by an adverb (which cannot be attached in a bottom-up approach), the matching rule would explicitly have to state that, in an NC, there could be a sequence of two AJACs with an interfering AVC, or otherwise leave the sequence of AJACs and AVC within the NC unspecified. In the latter case, however, there would be the need to introduce markers to delimit the NC, but, in this case, there would be the need for more than one rule. In fact, this strategy would then be the same as ours, and our strategy works best top-down.

The inner structure of the chunks is annotated afterwards as it is illustrated in figure 5.22, which shows the annotation steps for chunks for sentence 71: First, the PC is annotated. After this the NCs are annotated, both inside PCs and outside. Then, the attributive adjectives chunks are recognized inside the NCs. Adverbs which appear in front of attributive adjectives become part of the attributive adjective chunk like ‘unheimlich’ and are annotated as adverb chunks in a last step. PCs and NCs are the only chunks in which considerable complexity can occur.

(71) Mit drei unheimlich klobigen Koffern erschienen die großen
With three scaringly bulky trunks appeared the tall
Männer.
men.

‘The tall men appeared with three scaringly bulky trunks.’

Since we simply use PoS tag information for the disambiguation, adverbs are only attached in cases in which their attachment is unambiguous like ‘unheimlich’ in figure 5.22, in which the adverb appears between a marker of the beginning of a chunk (here: the preposition ‘mit’) and a marker of its end (here: the noun ‘Koffern’). Otherwise, adverb chunks are attached to the topological field in which they appear. Adverb chunks typically consist of a single adverb unless they are modified by particle like ‘(all)zu’ (PTKA). If an adverb is modified by another adverb, attachment is high because there is



```
Mit drei unheimlich klobigen Koffern erschienen die großen Männer.
[PC Mit drei unheimlich klobigen Koffern] erschienen die großen Männer.
[PC Mit [NC drei unheimlich klobigen Koffern]] erschienen [NC die großen Männer].
[PC Mit [NC [AJAC drei] [AJAC unheimlich klobigen] Koffern]] ersch. [NC die [AJAC großen] Männer].
[PC Mit [NC [AJAC drei] [AJAC [AVC unheimlich] klobigen] Koffern]] ersch. [NC die [AJAC großen] Männer].
```

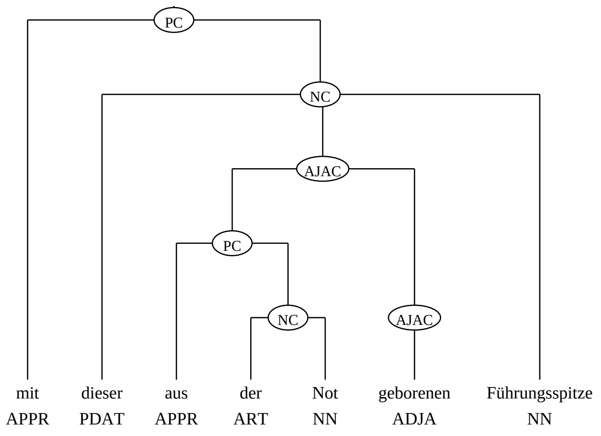
Figure 5.22: Illustration of the top-down annotation of chunks

typically no way to distinguish between adverb-adverb modification and two disjunct adverbials. All these facts account for all adverbs regardless whether they are tagged ADV or ADJD. The only difference is that ADVs receive the chunk label AVC and ADJDs receive the chunk label AJVC because they have the potential to be either adverbials or predicatives. Since AVC and AJVC are not complex, the distinction between top-down and bottom-up parsing does not apply.

Center-embedded structures like the one in figure 5.23 are annotated mainly within the same steps as simple structures. Figure 5.24 illustrates the steps: At first the center-embedded PC is annotated since we apply a top-down strategy. Because we do, in fact, allow PCs in NCs, the NC, which includes the PC, can be annotated after this. However, this works only if the PC appears after a marker of the beginning of an NC such as prepositions, determiners, or attributive adjectives. The other NC which is contained in the PC is also annotated in this step although it is contained in the NC dominating (at that point) the PC. This is possible because the rules for NCs are applied in clauses, fields **and** PCs. The rule is, thus, applied **twice** to the whole string of PoS tags associated with the string in figure 5.24: Once for the string 'mit dieser <PC> geborenen Führungsspitze' and once to 'aus der Not'.

In the next step, the included attributive adjective chunk can be annotated. Attributive adjective chunks are also defined as being able to include PCs. Thus, the adjective 'geborenen' and the preceding PC can be grouped as an AJAC. Since such structures are only annotated **within** the already existing NCs, wrong annotation is ruled out. Until this step, all the rules are the same as with ordinary top-down annotation. Only then, there are two further rules which complete the structure: One rule annotates the attributive adjective as an AJAC and another one renders possible the annotation of PCs which contain center-embedded PCs since, at the time of the annotation of PCs, this complex structure was not visible. It was, thus, possible to integrate center-embedded structures to a large extent into the top-down parsing approach of chunks without adding too many extra annotation rules.

Further structures to be discussed are coordinated chunks: Since these are closer to the root node in the syntactic tree than simple chunks and since we follow a top-down parsing strategy for chunks, we annotate coordinated chunks before simple chunks. In principle, the same strategy can be applied as with simple chunks. The markers for chunks, on the one hand, and the head noun, on the other hand, delimit the simple chunks and these are co-



mit	dieser	aus	der	Not	geborenen	Führungsspitze
with	this	by	the	necessity	born	top management

'with this mangement which was just installed because necessity required it'

Figure 5.23: Center-embedded PC from TüPP-D/Z

ordinated by, e.g., a conjunction. That way, the borders of the coordinated chunk can be annotated before the annotation of simple chunks takes place.

Although this basically means that the same rule is applied twice to certain strings, this strategy is reasonable as the coordinated PC in figure 5.25 shows: If coordination is treated after the annotation of simple chunks, then the full PC cannot be annotated since the PC beginning with the preposition 'zwischen' would end with the noun 'Idee', and the NC 'stumpfer Ausführung', which is coordinated with the NC dominated by the PC, would be left outside the PC. Coordination could, in fact, better be treated by a bottom-up strategy since, with a bottom-up strategy, this problem would be avoided, completely. We have, however, decided to follow a top-down strategy because, on the whole, the advantages of the top-down strategy outweigh the disadvantages.

(72) * [$_{\text{PC}}$ zwischen herrlicher Idee] und [$_{\text{NC}}$ stumpfer Ausführung]

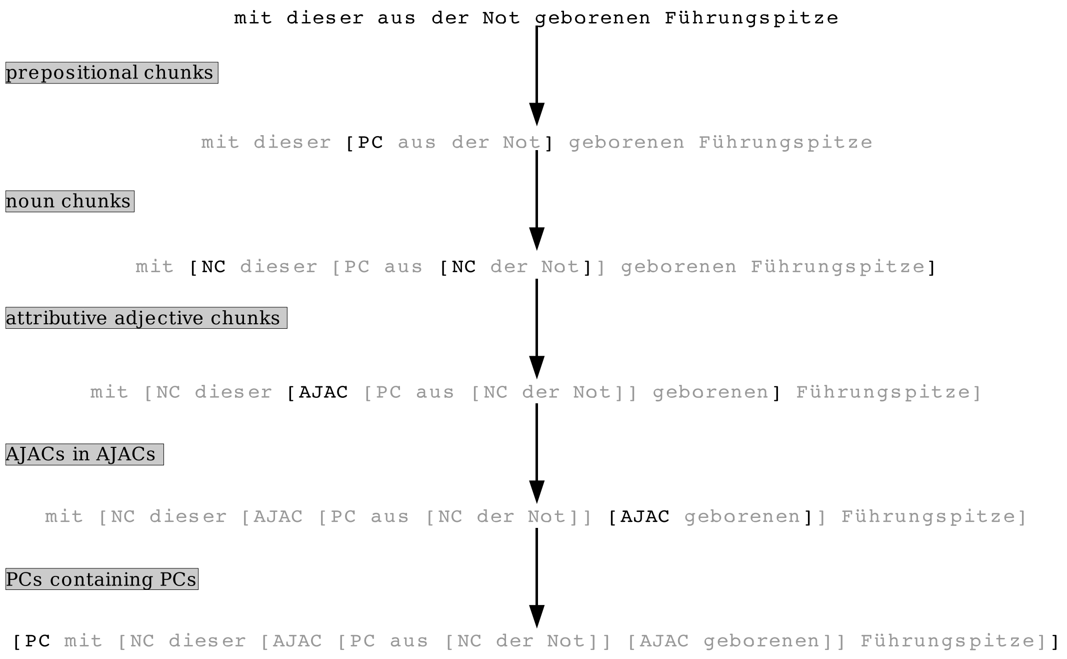
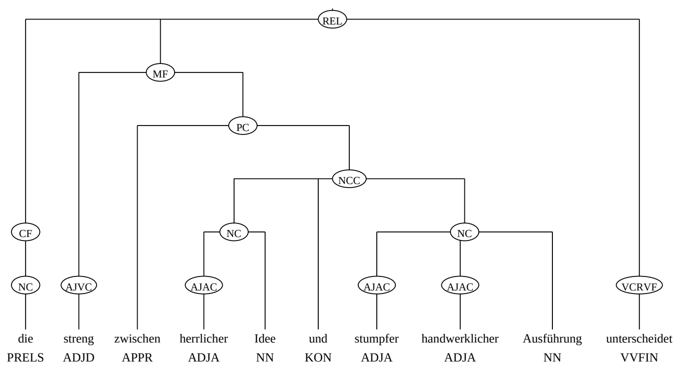

mit dieser aus der Not geborenen Führungspitze
prepositional chunks
mit dieser [PC aus der Not] geborenen Führungspitze
noun chunks
mit [NC dieser [PC aus [NC der Not]] geborenen Führungspitze]
attributive adjective chunks
mit [NC dieser [AJAC [PC aus [NC der Not]] geborenen] Führungspitze]
AJACs in AJACs
mit [NC dieser [AJAC [PC aus [NC der Not]] [AJAC geborenen]] Führungspitze]
PCs containing PCs
[PC mit [NC dieser [AJAC [PC aus [NC der Not]] [AJAC geborenen]] Führungspitze]]

Figure 5.24: Illustration of the annotation of center-embedded chunks



die	streng	zwischen	herrlicher	Idee	und	stumpfer	Ausführung	unterscheidet
which	strictly	between	magnificent	idea	and	dull	realization	differenciates

‘which strictly differenciates between magnificent idea and dull realization’

Figure 5.25: Coordinated PC from TüPP-D/Z

Chapter 6

Annotating Complex Chunks

Abstract: Section 6 explains why we have to annotate structures which we will call *complex chunks* before we begin with the annotation of grammatical functions. Section 6.1 describes the problems which occur if complex chunks are not annotated and, thus, motivates their annotation. Section 6.2 defines appositional constructions and Named Entities which are the main structures which are annotated as complex chunks. Section 6.3 describes the algorithm used to annotate appositional constructions and section 6.4 does the same for Named Entities. Section 6.5 describes the annotation of other structures which can be subsumed under the label of complex chunks.

6.1 Task Description and Motivation

After the annotation of shallow structures, there exist clauses, topological fields and chunks which serve as the basis for the further annotation of grammatical functions. Chunks, which, in brief, can be described as non-recursive core phrases without post-modifying structures, are the target constituents for the annotation of grammatical functions described in part II. Sentences 73 and 74 give examples. In sentence 73, the verb sub-categorizes for a nominative object (ON), an accusative object (OA) and a prepositional object (OPP). Each of these is represented by a chunk. In sentence 74, the verb sub-categorizes for an ON and an OA. However, only the head chunk of the phrase 'den Kolben mit Wasser' is marked as the OA. The attachment of the chunk 'mit Wasser' is left unspecified. It may be resolved in further steps.

(73) [$_{\text{ON}}$ Der Chemiker] füllt [$_{\text{OA}}$ den Kolben] [$_{\text{OPP}}$ mit Wasser] an.
The chemist fills the flask with water in.

'The chemist fills water into the flask.'

(74) [$_{\text{ON}}$ Der Chemiker] sieht [$_{\text{OA}}$ den Kolben] mit Wasser an.
The chemist looks the flask with water at.

'The chemist looks at the flask with water.'

Since NCs are only modifiers in the case of the genitive, most NCs serve as grammatical functions and can be disambiguated by means of morphological case information and linear order. However, this leads to problems in the case of phrases with more than one NC, which mainly occur in *appositional constructions* and *Named Entities* (NEs) as sentences 75 and 76 show. In sentence 75, the verb 'bearbeiten' sub-categorizes for an ON and an OA. There are, however, four chunks, and they all have the morphological potential to be ON or OA. In this case, it would be highly desirable to know that 'Bankdirektor Schneider', on the one hand, and 'die Abteilung Kreditwesen', on the other hand, are one chunk because then there would only be two potential targets for ON and OA.

(75) Er sagte, daß [$_{\text{NC}}$ Bankdirektor] [$_{\text{NC}}$ Schneider] [$_{\text{NC}}$ die Abteilung]
He said that bank manager Schneider the department
[$_{\text{NC}}$ Kreditwesen] bearbeitet.
credit systems handles.

'He said that bank manager Schneider deals with the department of credit systems.'

(76) Ich glaube, daß [$_{\text{NC}}$ er] [$_{\text{NC}}$ Anfang] [$_{\text{NC}}$ April] [$_{\text{NC}}$ Urlaub] gemacht
I think, that he beginning April holiday made
hat.
has.

'I think that he has been on holiday at the beginning of April.'

In sentence 76, 'machen' also sub-categorizes for an ON and an OA but there are also four chunks. The first is unambiguously ON but the following three all have the morphological potential to be the OA. However, the last of the three is the OA and the two chunks 'Anfang' and 'April' constitute a phrase which is an NE which is very likely an adverbial because it is a temporal expression. It would, thus, also be highly desirable to be able to deal with those NEs. The following section 6.2 defines appositional constructions and NEs, and sections 6.3 and 6.4 describe the methods used for their annotation.

6.2 Appositions and Named Entities

The appositional construction is generally seen as a (special) case of modification (cf. Quirk, Greenbaum, Leech, and Svartvik (1985), sec. 17.65; Engel (1996), sec. E 032; Eisenberg, Gelhaus, Wellmann, Henne, and Sitta (1998), sec. 1171; Biber, Johansson, Leech, Conrad, and Finegan (1999), p. 575; Eisenberg (1999), p. 249; Helbig and Buscha (1999), sec. 15.2.6). There are, however, also other sources which define appositional constructions without mentioning modification (cf. Bußmann, 2002; Crystal, 1997). This is due to the confusion noticed by Eisenberg (1999): "Es besteht keine Einigkeit darüber, was unter 'Apposition' zu verstehen ist, [...]"[1]. Linguists mostly agree that an appositional construction consists of two noun phrases which form one constituent. In this constituent, the two noun phrases correspond with each other syntactically and referentially (cf. Bußmann, 2002), i.e. they have the same grammatical function and denote the same object in the extra-language reality. This is reflected in the fact that one of the two constituents in an appositional construction can be left out without the sentence becoming ungrammatical.

Linguists do not always agree, however, which noun phrase modifies the other one and which one is, consequently, the head of the constituent and which one the apposition (i.e. the modifier). This is due to the fact that modification is sometimes defined on the grounds of differing criteria, i.e. sometimes it is defined semantically and sometimes morphologically. Eisenberg (1999) goes as far as to say: "Das Verhältnis ist variabel bis hin zur Gleichberechtigung, die sich grammatisch als Übereinstimmung im Kasus geltend macht."[2] This equality resembles the equality in coordination, in which no head is assigned in TüBa-D/Z. Similarly, no heads are assigned in appositional constructions and both parts of the appositional construction are equally assigned the label 'APP' in the TüBa-D/Z (cf. figure 6.1).

[1] *Linguists do not agree on a common definition of 'apposition'.* (lit. *There is no unity about what is to be understood by 'apposition'.*)

[2] *The relation is variable up to an equality which is reflected grammatically by the agreement in case.*

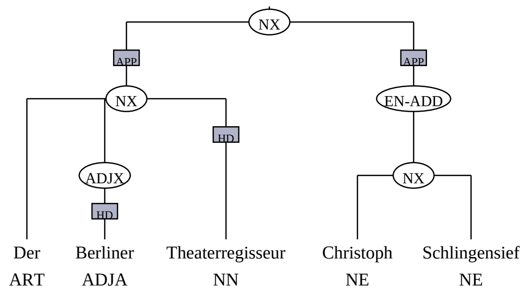


Figure 6.1: Example of tightly constructed appositional construction

The nature of appositional constructions and its annotation in TüBa-D/Z have two consequences for our parser. First, since it is not always generally agreed upon which of the two nouns is the head of the appositional construction and since we propose to annotate that chunk as a grammatical function which contains the head, we need to annotate both parts of an appositional construction. Second, since the two NCs which are in the relation of an apposition may directly follow each other without any marker of apposition just like two NCs which represent two disjunct grammatical functions, we need to find a way to annotate appositional constructions with the help of other features than the pure chunk information. It is not trivial to detect appositives and keep them apart from distinct phrases.

There is yet another group of complex chunks which needs annotation: These are usually referred to as *Named Entities* (NEs). NEs comprise expressions denoting names (e.g. of persons, organizations or locations), temporal expressions (e.g. dates or the time) or numerical expressions (e.g. money or percentages) (cf. Chinchor and Robinson, 1998). For our purposes, those NEs are important which comprise more than one token and which deviate from the standard grammatical distribution because, if an NE fits into one of the standard grammatical distributions, it is annotated by our chunk component. Only if it does not, there is a need for a special component which integrates tokens which form an NE constituent into one structure and, thus, prevent wrong annotation of grammatical functions. Another point is that temporal expressions cannot usually serve as those grammatical functions which we annotate, i.e. sub-categorizable grammatical functions (cf. section 7.2 for details). Thus, it is better to mark them such that they can be separated from other NCs.

6.3 Annotation of Appositional Constructions

There are different types of appositional constructions. Most of the grammars distinguish tightly constructed from loosely constructed appositional constructions. In tightly constructed appositional constructions, the noun phrases directly follow each other without a pause in speech and without a comma in print (cf. example 77). In loosely constructed appositional constructions, the two noun phrases are separated by a pause in speech and by a comma in print (cf. example 78). The second part of the appositional construction is even followed by a comma, which is very useful for annotation because it is a quite reliable sign of a loosely constructed appositional construction. Tightly constructed appositional constructions, however, strongly require a lexicon to render annotation possible. If two NCs directly follow each other, it is not clear whether they are an appositional construction without looking at the lexical heads of those NCs. One of the nouns, which is always a common noun, indicates whether the two chunks form an appositional construction. We call this noun the 'indicating noun'. In order to acquire those indicating nouns, we extract information about appositional constructions from the TüBa-D/Z, excluding that part of the TüBa-D/Z which we use for evaluation.

(77) Bundespräsident Köhler
Federal President Köhler
'Federal President Köhler'

(78) der Leiter, Erik Hohlmeier,
the head, Erik Hohlmeier,
'the head, Erik Hohlmeier,'

Since appositional constructions are annotated in the TüBa-D/Z (cf. figure 6.1), it is easy to extract them from the corpus. We divided them into different classes. The first and easiest class was the class in which the first part of the apposition was headed by a common noun (NN) and the second part was headed by a proper noun (NE). This class included all the title–name appositional constructions like the one in figure 6.1. The first NC contains the title (here: 'Theaterregisseur', *director*) and the second one the name (here: 'Christoph Schlingensief'). Then, we yielded a list of all the nouns in the first (i.e. indicating) part of the appositional construction.

While the extraction of the appositional constructions and the generation of the list are strictly formalized, the following steps rely on linguistic intuition.

Since many of the nouns in the first part of the appositional construction are compounds like ‘Theaterregisseur’ in figure 6.1, it has to be decided whether the whole token should be taken for the lexical entry or just the determinatum (i.e. the last nominal element in the token). In the case of ‘Theaterregisseur’, the string which was extracted was ‘[Rr]egisseur’. During the annotation, our parser checks whether there is a chunk containing a noun ending in the string as specified in the lexicon. In this process, the inflection of the nouns is also taken into consideration.

The extraction process yielded a lot of nouns which were expected to come up like ‘Abgeordnete’ (delegate), ‘Direktor’, ‘Minister’ or ‘Präsident’ which are typical titles. But it also yielded a lot of unexpected nouns like ‘Aufsteiger’ (promoted team (in sports)), ‘Fall’ (case), ‘Haltestelle’ (stop) or ‘Haus’ (house, hall, company). This showed that it was worthwhile applying an automatic extraction algorithm on the corpus to acquire those nouns. Many of the nouns would not have been included in a list from e.g. a grammar. The examples also show that there are many compounds. This was, in fact, one reason, why we chose to apply a mixed formalized/intuitive approach.

(79)	mit	seinem	möglichen	Zweitliga-Aufsteiger	Alemannia
	with	his	potential	second division promoted team	Alemannia

‘with his team Alemannia, a potential promoted team to the second division’

(80)	der	Fall	Jorgic
	the	case	Jorgic

‘the Jorgic case’

(81)	an	der Endhaltestelle	Sebaldsbrück
	at	the end stop	Sebaldsbrück

‘at the terminal stop Sebaldsbrück’

(82)	das Concerthaus	Brandenburg
	the concert hall	Brandenburg

‘the Brandenburg concert hall’

The most important point about the mixed formalized/intuitive approach is that it allows us to abstract from single instances. If the first noun in an appositional construction in question is e.g. ‘Ex-SPD-Bürgerschafts-abgeordnete’, then we can intellectually deduce that it is the noun ‘Abgeordnete’ which is the indicating noun for the appositional construction. If we had just extracted all the nouns, we would have yielded a much larger and more unreliable lexicon since, in some cases, the actual indicating noun for the appositional construction (here: ‘Abgeordnete’) might not even be included in the corpus. It would also not suffice just to use the morphological information about the compound structure because it is not always clear which part of the compound to take.

In the noun ‘Sport|rechte|makler’ (sports|license|trader; i.e. *trader of (TV) licenses for sport events*) it is the last element ‘Makler’ which is included in the lexicon; but, in the noun ‘End|halte|stelle’ (terminal|stop|place; i.e. *terminal stop*), it is the last two elements ‘Haltestelle’ which are included in the lexicon since the last element ‘Stelle’ (place) is far to general. Its inclusion into the lexicon would lead to the annotation of false positives. The advantage of the strategy chosen in our approach is that, by looking at a section of the corpus, it allows us, to a certain extent, to generalize over the corpus because we can annotate structures which are indicated by nouns we have never encountered before. Thus, the adaptation of the noun ‘Vorstandssprecherin’ (*executive board spokeswoman*) into our parser allows the annotation of an appositional construction with the noun ‘Menschenrechtssprecher’ (*human rights spokesman*) which was not included in the training material. The algorithm for the extraction yielded a list of 459 nouns after intellectual processing. The intellectual processing took about two hours.

In the cases above, the appositional construction consisted of a common noun followed by a proper noun. Since the indicating noun has to be a common noun, the case was clear. This is different in appositional constructions in which two common nouns may occur. In these appositional constructions, the first noun may be either a common noun or a proper noun and the second noun is always a common noun. The indicating noun is either the only common noun in the construction (if the other is a proper noun) or any of the two common nouns. Since, in the TüBa-D/Z, both constituents of an appositional construction are equal (i.e. none is annotated as the head), both cases of indication are matched by the same corpus query. Even after the extraction it is, thus, not clear which noun is the indicating noun. Only if one noun occurs more than once in either the first or the second position, it is a hint that this noun is an indicating noun for either the first or the second position. Still, this finding has to be thoroughly checked by linguistic intuition and by considering the corpus samples.

(83)

der	Aktionsplan	Alkohol
the	action plan	alcohol

'the action plan alcohol'

(84)

die	Abteilung	Rechnungswesen
the	department	accounting

'the department of accounting'

(85)

das	immer	gleiche	Thema	Liebe
the	always	alike	topic	love

'the never changing topic of love'

(86)

der	hochgefährliche,	krebserregende	Baustoff	Asbest
the	highly dangerous,	carcinogenic	building material	asbestos

'the highly dangerous and carcinogenic building material asbestos'

The cases in which the second noun in an appositional construction is not a proper noun are not dealt with as central in grammars of German. Generally, the supposedly more typical combinations with common nouns as the first noun and proper nouns as the second are the focus of attention. For instance, aside from time and measure expressions like 'das Jahr 2000' (the year 2000) and 'ein Glas Milch' (a glass of milk), Engel (1996), sec. N 140, does not mention any appositional constructions with common nouns as second element.[3] Only Eisenberg, Gelhaus, Wellmann, Henne, and Sitta (1998), p. 675, allow them as one of two cases of appositional constructions: "Auf ein beliebiges Substantiv kann ein zweites folgen, welches das erste näher bestimmt (determiniert)."[4] We do not exactly follow this approach. In fact, our findings are that at least one of the nouns in tightly constructed appositional constructions has to be an indicating noun although this does not change anything as regards the determination of the first noun by the second one as defined in Eisenberg et al. (1998), p. 675.[5]

[3]Engel calls the tightly constructed apposition 'nomen invarians'; but the linguistic phenomenon and its subclassifications are identical with the ones other linguists list under the heading of appositional constructions.

[4]*Any noun can be followed by a second one which further determines the first.*

[5]Since determination is not annotated in TüBa-D/Z, we do not argue about this point.

Examples 83–86 show that the choice of the nouns which are first in the appositional construction is by no means free. In fact, it is the second noun which is free. Virtually anything can be thought of as an 'action plan' (83) or 'topic' (85). The other two examples are only limited to a certain semantic field. If we try, however, to put any of the second words first, then the second noun is restricted. This is precisely what we mean by indication of appositional constructions. There is always one noun which is restricted and that is the one which has to be found. By taking into consideration which nouns occur more than once as indicators, and by linguistic intuition and corpus checking, we arrive at a list of about 80 indicating nouns which are the first noun in appositional constructions. Examples 83–86 show, again, that indicating nouns may be compounds with the implications which were described for common noun–proper noun compounds.

(87) der Aufschwung Ost
the boom east
'the boom in the east'

(88) das Rathaus Mitte
the town hall center
'the central town hall'

(89) der Nomad Verlag
the Nomad publishing house
'the Nomad publishing house'

(90) die Werbepalette GmbH
the Werbepalette PLC
'Werbepalette PLC'

With respect to the determination as defined in Eisenberg et al. (1998), p. 675, appositional constructions in which the second noun is the indicating noun do not differ from those in which the first one is the indicating noun. It is simply the second noun which is restricted in them. We applied the same algorithm as with constructions with a restricted first noun. The construction occurs quite rarely and we only acquired a list of six words ('AG', 'Farm', 'GmbH', 'Mitte', 'Ost', 'Verlag'). The words roughly come from two classes: Words which describe the organization form mainly of companies and words which describe the location mainly of an institution or company. We added some other organization forms of companies and the missing directions. In appositional construction like this, the first noun may also be a proper noun. Examples 87 and 88 show constructions with common nouns and examples 89 and 90 show examples with proper nouns.

Tightly constructed appositional constructions are recognized using lexical information acquired from the corpus as described above. Loosely constructed appositional constructions are recognized using their unique distributional information. Since the second element of a loosely constructed appositional construction almost exclusively follows the first one included in commas, it is possible to annotate it without using any lexical information. Proper names may be the first or the second element of the loosely constructed appositional construction as can be seen in example 91 and 92.

(91)	Havemann,	Elektriker	und	Autor,
	Havemann,	electrician	and	writer,

‘Havemann, an electrician and writer’

(92)	Radunskis	Sprecher,	Axel	Wallrabenstein,
	Radunski’s	speaker,	Axel	Wallrabenstein,

‘Radunski’s speaker, Axel Wallrabenstein,’

6.4 Annotation of Named Entities

The annotation of NEs is important for two reason: First, some NEs are typically not able to act as grammatical functions. Thus, if these NEs were treated just like other NCs, this would often lead to wrong annotation decisions. This can be illustrated by sentence 93, in which the verb ‘auflösen’ sub-categorizes for an ON and an OA. There are, however, two NCs which have the morphological potential to be the OA. However, because the chunk ‘1982’ is very likely a temporal expression, it is less likely an OA but a lot more likely an adverbial. For that reason, our NE recognition component marks this chunk as ‘temporal expression’ and, thus, our parser first tries to assign another chunk the grammatical function ‘OA’, which, in this case, is ‘seine Band Japan’, which is the correct chunk.

(93)	seit	$[_{\text{ON}}$ er]	$[_{\text{NC-TEMP}}$ 1982]	$[_{\text{OA}}$ seine	Band	Japan]	auflöste
	since	he	1982	his	band	Japan	dissolved

‘since he dissolved his band *Japan* in 1982’

(94) [$_{\text{ON}}$ der] insgesamt [$_{\text{NC-QUANT}}$ 16 Jahre lang] [$_{\text{PRED}}$ Bremer
who all in all 16 years long Bremian
Theaterchef] war
theater boss was

'who has been the Bremen theater director for 16 years'

The second problem with NEs is that they might be complex and, consequently, one phrase can appear as more than one chunk in the pre-annotated corpus. This might also lead to the wrong annotation of grammatical functions. For instance, in sentence 94, the NE '16 Jahre lang' has a word order which deviates from the prevailing word order of a chunk since these can normally not be post-modified by an adjective. The NE recognition component annotates cases like this in which post-modification is, in fact, possible. This excludes 'lang' from being annotated as a predicative since the verb 'sein' (here realized as 'war') may take either an adjective or a noun as a predicative. Since the adjective 'lang' is incorporated into the NE, only the NC 'Bremer Theaterchef' remains as the predicative. This is the correct solution.

The annotation of NEs is lexicon-based like the annotation of appositional constructions. However, unlike with the appositional constructions, the lexicon was not acquired through a corpus search but through the analysis of annotation errors of the grammatical function component. From the NEs which were found, we abstracted and achieved a list of words which together composed NEs. This includes constructions like 'Anfang April', 'Dezember 1992' or 'fast 80 Meter lang'. In contrast to the appositional construction component, the NE component is applied before the general chunking takes place. This is due to the fact that NEs do not adhere to the prevailing word order in chunks and annotating them with or after the chunks would, thus, not be possible.

6.5 Annotation of Other Complex Chunks

There are still some other complex chunk structures which were annotated. As regards the choice of these complex chunks, there were two main criteria: First, the annotation had to be carried out very reliably. This would, e.g., not be the case with post-modifying prepositional phrases. Second, not annotating the complex chunks would lead to errors in the grammatical function annotation component. If not annotating a complex chunk would simply mean to leave a structure underspecified, we took that choice rather than taking the risk of wrong annotation.

The annotated complex chunk structures subsume some quite frequent, generally describable distributions and also some less frequent and specific structures. Among the former are structures in which a noun chunk is followed by another NC in brackets (cf. example 96 and 97). Among the latter are structures with 'darunter', a word which can act as an independent prepositional phrase but which can also form a prepositional phrase which pre-modifies a noun phrase (cf. example 95). What is most important for our parser in this case is that an NC with 'darunter' pre-modification is a modifier of another NC. Thus, this noun chunk can never itself be a grammatical function.

Sentence 95 gives an example: The verb 'ermöglichen' has a sub-categorization frame which demands 'ON', 'OD' and 'OA'. Since the chunk parser has annotated '7000 Tierarten, darunter 5000 Insekten,' as one chunk, the grammatical function annotation component correctly annotates the respective functions as shown in sentence 95. Without the special treatment of 'darunter', '5000 Insekten' forms one independent chunk, which is then incorrectly annotated as the 'OA' of the sentence. Examples 96 and 97 show chunks with following chunks in brackets. The reason for annotating these is that they follow a fixed scheme which can be reliably annotated. Additionally, not annotating those structures would result in more chunks which could obstruct correct annotation of grammatical functions. The structure within these chunks is left underspecified. In some cases, the chunk in brackets is a modifier to the preceding chunk like in sentence 96; in other cases, the following chunk forms an appositional construction with preceding one like in sentence 97, in which case TüBa-D/Z sees the chunks as equal.

(95)

[$_{ON}$ die]	[$_{OD}$ 7000 Tierarten,	darunter	5000 Insekten,]	[$_{OA}$
which	7000 animal species,	among them	5000 insects,	

das Leben]	ermöglicht
the life	allow

'which allows living for 7000 species, among them 5000 (species of) insects'

(96)

Außerdem	lobte	Ludewig	Verkehrsminister	Franz
Apart from that	complimented	Ludewig	traffic minister	Franz

Müntefering (SPD).
Müntefering (SPD).

'Apart from that, Ludewig complimented the minister of transport Franz Müntefering.'

(97) Der Fotograf Dick Avery (Fred Astaire) nimmt die intellektuelle
The photographer Dick Avery (Fred Astaire) takes the intellectual
Buchhändlerin Jo Stockton (Audrey Hepburn) für Modeaufnahmen
bookseller Jo Stockton (Audrey Hepburn) for fashion photos
als Modell mit nach Paris.
as model with to Paris.

'The photographer Dick Avery (Fred Astaire) takes the intellectual bookseller Jo Stockton (Audrey Hepburn) for fashion photos as a model with him to Paris.'

Another case for the annotation of complex chunks are parentheses like, for instance, the parenthesis very common in newspaper language which gives the source of a reproduced statement like in sentence 98. This parenthesis is always introduced by 'so'. In the TüBa-D/Z, it is annotated as material outside the actual sentence. For our parser, it is important that it is marked as a phrase which cannot serve as a grammatical function for the same reason as with the other complex chunks.

(98) Kultursenator Peter Radunski (CDU) habe, so
Culture minister Peter Radunski (CDU) has, according to
Zeitungsberichte, den Leipziger Intendanten als Nachfolger von
newspaper reports, the Leipzigian artistic director as successor of
Götz Friedrich festgelegt.
Götz Friedrich appointed

'According to newspaper reports, the minister of arts Peter Radunski (CDU) has appointed the Leipzig artistic director as a successor to Götz Friedrich.'

Part II

Annotating Grammatical Functions

Chapter 7

Annotating Deeper Structures – Grammatical Functions

ABSTRACT: Chapter 7 introduces the linguistic phenomenon of grammatical functions (GFs) and gives the motivation for its annotation. Furthermore, it puts the annotation of GFs into the context of the whole parser and introduces the linguistic knowledge sources and methods which are used for the annotation. Section 7.1 gives an overview of the task description and explains why morphological information and sub-categorization frames are crucial for mastering this task. Section 7.2 gives a definition of GFs. It also puts them into a context with shallow annotation structures. Section 7.3 gives an account as to which structures are precisely annotated as GFs and in which cases there are problems in the definition of a GF. Section 7.4 explains why we chose GFs as a target for our parser. Section 7.5 points out how GFs are annotated on top of the shallow parsing structures. Additionally, it gives an outlook of the methods which are being used for the annotation of GFs.

7.1 Outline of the Task Description

GFs are annotated on top of the shallow annotation structure of chunks, topological fields and clauses (cf. Müller, 2004). Since GFs and shallow annotation are on different levels of syntactic description, they require different sources of linguistic knowledge to be annotated. Thus, they are annotated by two different components. Figure 7.1 illustrates the input of the GF annotation component, which is the shallow annotation structure and figure 7.2

ON	nominative object
OG	genitive object
OD	dative object
OA	accusative object
OPP	prepositional object
OS	sentential object
PRED	predicative

Table 7.1: Grammatical functions annotated by our parser

shows a syntactic tree from the TüBa-D/Z which contains the GFs which our parser recognizes. Table 7.1 gives an overview of the GFs annotated by our parser.

Shallow structures are subject to syntactic restrictions and, consequently, it is possible to annotate them using just PoS tag information. By contrast, the annotation of GFs also requires information about the morphological features of the potential targets and information about the sub-categorization frame of the verb (or other constituent) to which the GFs relate. Hence, for the annotation of GFs, one needs a component which assigns the correct morphological features to the respective chunks and a resource which contains the sub-categorization frames of the respective verb. The reason why the annotation of grammatical functions needs morphological features and sub-categorization information can best be illustrated by figure 7.1, which is a tree from the TüPP-D/Z, in which only the shallow structure is annotated.

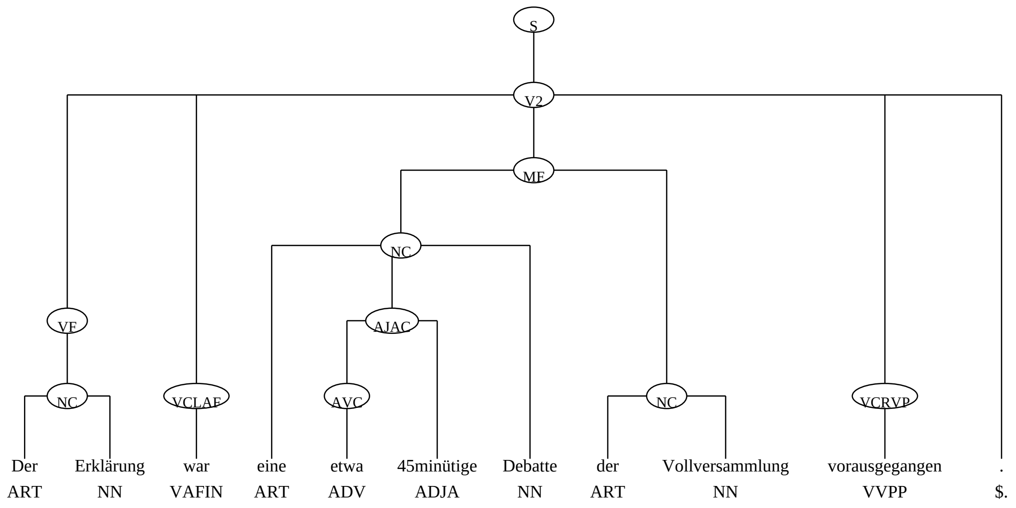
In the syntactic tree, the shallow syntactic structure consisting of chunks, topological fields and the clause is annotated. This is possible using syntactic restrictions like the one that, in the sequence 'VCLAF-any constituents (except those occurring in sentence bracket)-VCRVP', the sequence of any tags is the MF. After the annotation of the fields, it is clear where the boundaries of the V2 clause are (in the case at hand the whole sentence). Furthermore, chunks can be recognized using restrictions like the one that the sequence 'article, adverb, attributive adjective and noun' is an NC (cf. 'eine etwa 45minütige Debatte'). No morphological or lexical information is necessary to annotate this shallow structure.

By contrast, it is not possible to assign the GFs in the sentence in figure 7.1 without information about the morphological features of the three chunks and the sub-categorization frame of the verb 'vorausgehen'. The sub-categorization frame of the verb contains the information about how many and what kind of GFs a verb demands, i.e. sub-categorizes for. A verb may have more than one sub-categorization frame. In the case at hand, the verb 'vorausgehen' has two sub-categorization frames. One just demands a subject for the verb 'vorausgehen' and the other one a subject and a dative object.

If the sub-categorization frame is not taken into account and just non-lexical syntactic restrictions are used like 'assign feature *subject* to first NC, feature *dative object* to the second NC and feature *accusative object* to third one', it is not possible to correctly assign GFs. In the sentence in figure 7.1, the NC 'der Vollversammlung' is a genitive attribute to the NC 'eine 45minütige Debatte'. Together, the two NCs are the subject, the noun 'Debatte' being the head of it. The NC 'der Erklärung' is the dative object. Thus, the non-lexical syntactic restrictions would have yielded a wrong result.

With the information about the sub-categorization frame of the verb, the realization of potential GFs in figure 7.1 is already confined but the sub-categorization information alone does not suffice since there are still three NCs and it is not clear which one has which function. If one has a look at the morphological features, then it becomes clear that the three NCs are locally ambiguous. If, e.g. the NCs 'der Erklärung' and 'der Vollversammlung' are regarded outside any context, they may be either genitive or dative and the NC 'eine etwa 45minütige Debatte' may be either nominative or accusative. However, if one combines the shallow annotation structure, with the sub-categorization information and the morphological information, it is possible to disambiguate the GFs in the sentence. Since there is just one potential subject, the decision is clear in this case. As regards the dative object, the choice is the NC in the VF because this NC cannot be a genitive modifier since there is no other NC to be modified. Thus, with the combination of the three information sources, this problem can be completely solved. In other cases, the number of possible solutions can be reduced considerably.

The example illustrates two important points: On the one hand, it affirms the value of shallow annotation structures for the annotation of deeper structures like GFs and, on the other hand, it shows the importance of the morphological and sub-categorization information for the annotation of grammatical functions. Figure 7.3 shows how these data are related to the GFs annotation component, which is applied after the annotation of shallow structures by the KaRoPars parser. In it, the morphological information, as it is contained in



Der	Erklärung	war	eine	etwa	45minütige	Debatte	der
The	declaration	had	an	approximately	45-minutes	debate	of the

Vollversammlung	vorausgegangen.
plenum	preceded.

‘A talk of approximately 45 minutes had preceded the declaration.’

Figure 7.1: Sentence from TüPP-D/Z with shallow annotation

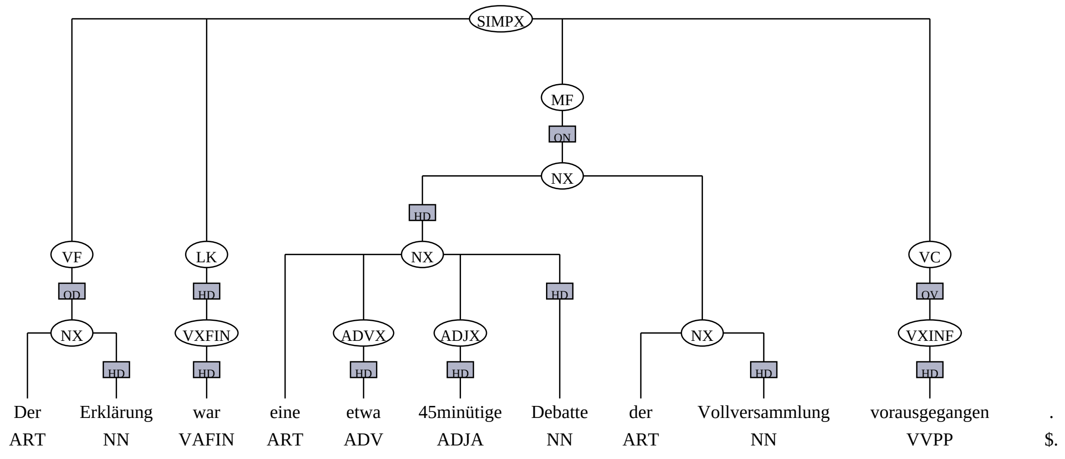


Figure 7.2: Sentence from TüBa-D/Z with full syntactic annotation

the morphological annotation tool DMOR (Schiller, 1995), is added to each token. A component using this morphological information and the shallow annotation structure then disambiguates the morphology of each chunk as far as this is possible using the shallow annotation structure. In the output of this process, the relevant tokens are then annotated with sub-categorization information from the electronic lexicon IMSLex (Eckle-Kohler, 1999), which we use as a data source for this purpose. After this, GFs can be resolved using the previously annotated information in a cascade of FSAs. The output of the GFs annotation component may then be used for further annotation processes.

7.2 Definition of Grammatical Functions

GFs can best be described and defined with respect to grammatical categories. While grammatical categories describe constituents independent of their use and in isolation, grammatical functions describe a constituent relative to other constituents in the sentence or relative to the sentence itself; and while grammatical categories mainly describe constituents on the basis of their inherent features, GFs describes constituents on the basis of their use (i.e. function). Thus, *relation* and *function* are the crucial concepts in the description of GFs. This is also reflected in the different terms for GFs.

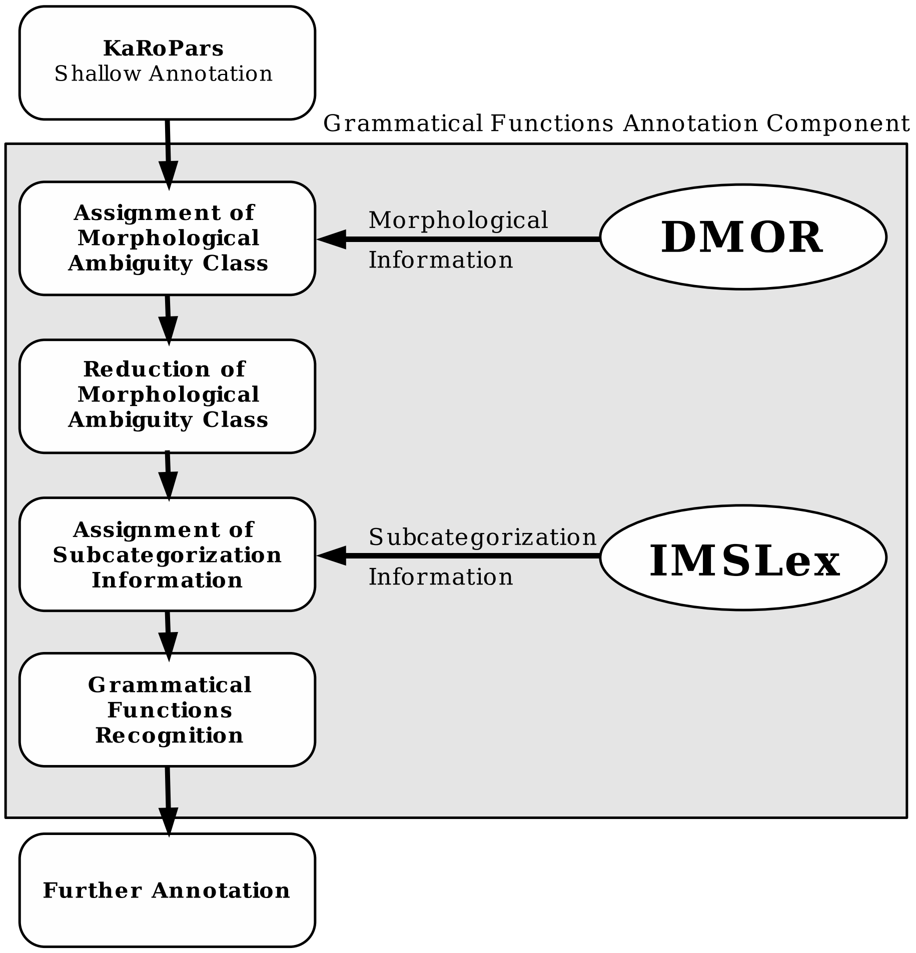

KaRoPars
Shallow Annotation
Grammatical Functions Annotation Component
Assignment of Morphological Ambiguity Class
Morphological Information
DMOR
Reduction of Morphological Ambiguity Class
Assignment of Subcategorization Information
Subcategorization Information
IMSLex
Grammatical Functions Recognition
Further Annotation

Figure 7.3: Grammatical functions annotation component

In Bußmann (1983) ‘syntactic functions’, ‘grammatical functions’, ‘syntactic relations’ and ‘grammatical relations’ are listed as terms which are used for the same concept according to different theoretical approaches. We use ‘grammatical functions’ in this thesis because we explain this phenomenon on both syntactic and morphological grounds (hence ‘grammatical’) and because we rather focus on its function in the sentence rather than simply on its relation to the verb. Like Bußmann (1983), however, we do not postulate a fundamental difference between the four terms. The importance of the two concepts relation and function and their interdependence is also reflected in Eisenberg (1998):

> *Syntaktische Begriffe wie Subjekt und Prädikat sind relational. Sie kennzeichnen eine Konstituente nicht für sich und unabhängig von der Umgebung, sondern sie kennzeichnen, welche Funktion eine Konstituente in einer größeren Einheit bei einer bestimmten syntaktischen Struktur hat. Sie wird damit in Beziehung zu anderen Konstituenten gesetzt und diese Beziehungen oder Relationen werden als Subjekt-Beziehung, Prädikat-Beziehung usw. bezeichnet.* (Eisenberg, 1998, p. 21)
>
> *Syntactic concepts like subject and predicate are* **relational**. *They characterize a constituent not for itself and independent of its environment, but they denote which* **function** *a constituent fulfills in a larger unit with a certain syntactic structure. The constituent is, thus, correlated with other constituents and those relations are referred to as subject relations, predicate relations and so forth.* (our translation and our emphasis)

(99) [$_{\text{V2}}$ [$_{\text{VF}}$ [$_{\text{NC}}$ Ein Fluch]] [$_{\text{VCLVF}}$ lastet] [$_{\text{MF}}$ [$_{\text{PC}}$ auf [$_{\text{NC}}$ dem Turm]]]].
An evil spell has been cast on the tower.

‘An evil spell has been cast on the tower.’

(100) [$_{\text{V2}}$ [$_{\text{VF}}$ [$_{\text{NC}}$ Ein Paar]] [$_{\text{VCLVF}}$ spaziert] [$_{\text{MF}}$ [$_{\text{PC}}$ auf [$_{\text{NC}}$ dem Turm]]]].
A couple strolls on the tower.

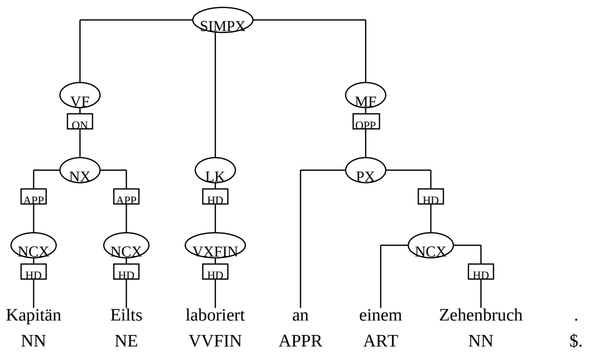
‘A couple is strolling on the tower.’

Sentence 99 and 100 further illustrate the crucial difference between the constituent structure of a sentence and the GFs. Sentence 99 and 100 both have an NP in their initial position in the VF and they both have a PP which is headed by the preposition 'auf' in the MF.[1] Thus, the constituent structure of the two sentences, i.e. the distribution of grammatical categories, is identical. The GFs occurring in the sentences differ, however. In sentence 99, there is a nominative object in the VF and a prepositional object in the MF, while, in sentence 100, there is a nominative object in the VF and an adverbial PP in the MF. The reason for this difference is the different sub-categorization frames of the two verbs. While 'spazieren' just sub-categorizes for a nominative object, 'lasten' sub-categorizes for a nominative object and a prepositional object with the preposition 'auf' and an NP with dative case. The difference of the two sentences which are identical in constituent structure is caused by the fact that GFs are subject to lexical selection while the constituent order is subject to syntactic restrictions. Although grammatical categories and GFs are by no means identical, they are still not totally independent of each other. A NC can, e.g., never function as a prepositional object. However, it is important to point out that "lexical items sub-categorize for function, not constituent structure categories" (Bresnan, 1982, p. 288).

GFs can be sub-divided into "sub-categorizable" and "non-sub-categorizable" GFs (Bresnan, 1982, p. 287). Sub-categorizable GFs subsume complements of verbs or adjectives, and non-sub-categorizable GFs subsume adjuncts like attributes or adverbials. Figures 7.4, 7.5 and 7.6 from the TüBa-D/Z illustrate this distinction with the aid of PPs and their different usages as complement (figure 7.4), as attribute (figure 7.5) and as adverbial (figure 7.6). In the sentence in figure 7.4 the PP 'an einem Zehenbruch' serves as a complement to the verb 'laborieren'; in the sentence in figure 7.5 the PP 'in Grohn' serves as an attribute to the phrase 'Privatuni'; and in the sentence in figure 7.6 the PP 'in Bremen' serves as a adverbial to the verb 'veranstalten'. We will only annotate sub-categorizable GFs, and we will henceforth mean sub-categorizable GFs if we speak of GFs. We rank predicatives (cf. sentences 101–105) among the GFs (cf. Eisenberg, 1999, p. 87–90, for a discussion).

The constituents which functions as the GFs for which a lexical items sub-categorizes have to possess certain features. If an NP functions as a GF,

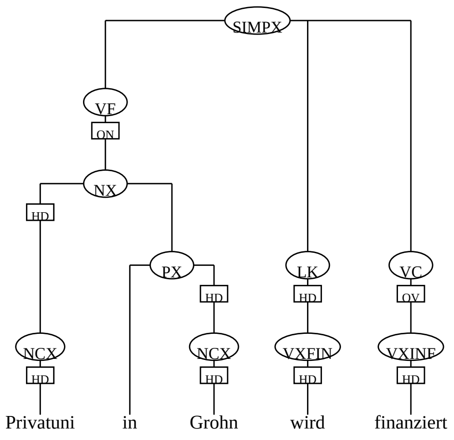
[1]The notation in this sentence is taken from the KaRoPars parser.



Kapitän	Eilts	laboriert	an	einem	Zehenbruch.
Captain	Eilts	labours	at	a	toe fracture.

‘Captain Eilts is plagued by a broken toe.’

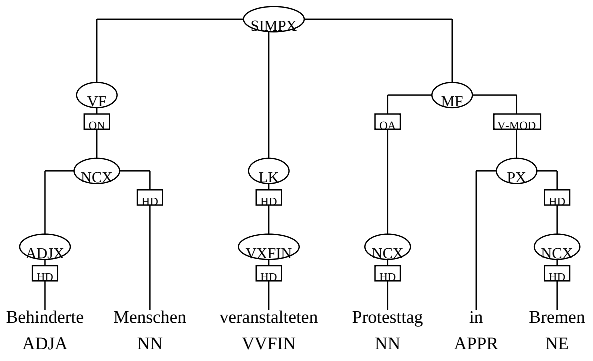
Figure 7.4: PP as prepositional object



Privatuni	in	Grohn	wird	finanziert
Private university	in	Grohn	is	funded

‘Private university in Grohn is (going to be) funded’

Figure 7.5: PP in attributive function



Behinderte	Menschen	veranstalteten	Protesttag	in	Bremen
Disabled	people	organize	protest day	in	Bremen

'Disabled people to organize day of protest in Bremen'

Figure 7.6: PP in adverbial function

case is the decisive feature as regards which lexical items the GF can be subcategorized by. The connection of the morphological features of an NP and its GF is so close that it is termed according to its case as nominative object, genitive object etc. instead of 'subject' or '(in)direct object' (cf. Reis, 1982). Only NPs in the nominative case can have two functions: ON and PRED. If the GF is realized as a PP, the decisive feature is the preposition and the case of the NP governed by the preposition. If it is realized by a clause, the decisive feature is by which nominal GF the clause could be replaced. If it can be replaced by ON or PRED, it is annotated as ON or PRED; if it can be replaced by any other nominal GF, it is annotated OS. For adverb phrases, adjective phrases and PPs functioning as predicatives, occurence with a copular verb and distribution is the decisive feature (cf. especially the difference in sentence 101 and 103)[2]. The precise definitions, in cases of doubt, are taken from the TüBa-D/Z stylebook (Telljohann, Hinrichs, and Kübler, 2003).

[2]We do not mark the relevant case features in the glosses if they are marked in the original sentence by their GF label, e.g. 'ON', here.

(101)	$[_{\text{ON}}$ Alfred]	ist	jetzt	$[_{\text{PRED}}$ da].
	Alfred	is	now	here.

‘Alfred is in, now.’

(102)	$[_{\text{ON}}$ Niemand]	ist	heute	$[_{\text{PRED}}$ krank].
	Nobody	is	today	ill.

‘Nobody is ill, today.’

(103)	$[_{\text{ON}}$ Alfred]	ist	jetzt	da	$[_{\text{PRED}}$ Lehrer].
	Alfred	is	now	there	teacher.

‘Alfred is a teacher there, now.’

(104)	$[_{\text{ON}}$ Seine	Mutter]	ist	$[_{\text{PRED}}$ in	der	Stadt].
	His	mother	is	in	the	city.

‘His mother is in the city.’

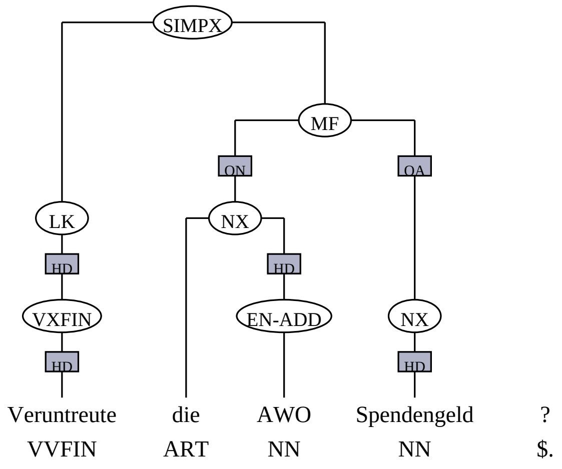
(105)	$[_{\text{ON}}$ Das Problem]	ist,	$[_{\text{PRED}}$ dass	niemand	zuhört].
	The problem	is	that	nobody	listens.

‘The problem is that nobody listens.’

7.3 Detailed Task Description and Problems

Figure 7.7 shows a simple sentence from TüBa-D/Z. The sentence contains three chunks (VXFIN and twice NCX), the two topological fields LK (Linke Klammer) and MF, and the two GFs ON and OA. While the annotation of the chunk and field structure was the task of KaRoPars, it is the task of the GFs annotation component to annotate the functions as they are illustrated in figure 7.7. We annotate GFs (cf. table 7.1) according to the definitions of the TüBa-D/Z, which we also use as a gold standard for evaluation. Some GFs can be realized by more than one constituent structure category, and some constituent structure categories can function as more than one GF. NCs can functions as ONs, OGs, ODs, OAs and PREDs. PCs can function as OPPs and PREDs. Clauses can function as ONs, OSs and PREDs. Adjective chunks and adverbial chunks can only function as PREDs.

The fact that we took over the definitions of the TüBa-D/Z leads to problems in some cases since we use the IMSLex for the sub-categorization information for annotation, and the definitions of TüBa-D/Z and IMSLex



Veruntreute	die	AWO	Spendengeld?
Embezzled	the	AWO	donation money?

‘Did the AWO embezzle donated money?’

Figure 7.7: Sentence from TüBa-D/Z

differ with respect to the precise definition of GFs. As regards the definitions of GFs which are defined on the basis of their case, most GFs can be clearly defined, the exception being OD: While there is a clear-cut line between nominative, genitive and accusative phrases in their function as sub-categorizable GFs and in their function as non-sub-categorizable GFs, there is no such line in the case of dative phrases. Roughly two functions can be made out for dative phrases: Clear dative objects and so-called free datives.

Free datives can be further divided into three groups (cf. Hentschel and Weydt, 1994, p. 159–163), one of which has no clear-cut line with dative objects. This is the *dativus (in)commodi*, which specifies to whose (dis)advantage an action occurs. Sentence 106[3] contains an example of a clear case in which the dative is a dative object. Sentence 107 shows a clear case of a *dativus commodi*. In sentence 108, however, the situation is not so clear: Eisenberg (1999), p. 288, classifies this dative as a dative object while Helbig and Buscha (1999), sec. 2.4.3.4., classify this dative as a *dativus commodi* because it can be replaced by a PP with *für* without changing the meaning of the sentence. Some grammars even count the *dativus commodi* as such among the dative objects (cf. Eisenberg, Gelhaus, Wellmann, Henne, and Sitta (1998), § 1156, and Engel (1996), sec. S 016). However, since the *dativus (in)commodi* can occur quite freely with verbs and its status as a complement is so uncertain, it is not listed in the sub-categorization frame of the verbs in IMSLex. But, since we are using the TüBa-D/Z as our gold standard, we have to annotate those cases as well because, in TüBa-D/Z, all dative phrases are assigned the status of OD (i.e. dative objects). There are, thus, structures which cannot be annotated with the help of the sub-categorization information of IMSLex alone.

(106)	[$_{ON}$ Markus]	schenkt	[$_{OD}$ seiner	Freundin]	[$_{OA}$ einen	Kuchen].
	Markus	presents with	his	girlfriend	a	cake.

‘Markus gives his girlfriend a cake as a present.’

(107)	[$_{ON}$ Markus]	backt	[$_{OD}$ seiner	Freundin]	[$_{OA}$ einen	Kuchen].
	Markus	bakes	his	girlfriend	a	cake.

‘Markus bakes a cake for his girlfriend.’

[3]All example sentences for datives are constructed.

(108) [$_{\text{ON}}$ Markus] kauft [$_{\text{OD}}$ seiner Freundin] [$_{\text{OA}}$ einen Kuchen].
Markus buys his girlfriend a cake.

'Markus buys a cake for his girlfriend.'

Another linguistic phenomenon which is problematic because its definition differs in IMSLex and in the TüBa-D/Z is the prepositional object OPP. The reason for this is that the definition of OPPs is both controversial and vague because the distinction of PPs in their function as prepositional objects and as adverbials is not clear-cut. Generally, prepositional objects can be defined as those PP in which the preposition and the case of the head noun of the NP is not free but demanded by the verb. However, the linguistic tests with which one can try to decide whether the verb governs the preposition and the case do not always yield an unambiguous result.[4] One of the tests proposed is that the prepositional object cannot be left out (i.e. it is obligatory). This is, however, not always true as sentence 109 shows. The prepositional object 'an ihrem Mitschüler' can be left out. Additionally, some adverbial PPs are obligatory as in sentence 110 in which 'in Frankfurt' cannot be left out.

(109) [$_{\text{ON}}$ Sabine] rächt [$_{\text{OA}}$ sich] [$_{\text{OPP}}$ an ihrem Mitschüler].
Sabine avenges herself on her schoolmate.

'Sabine takes revenge on her schoolmate.'

(110) [$_{\text{ON}}$ Mein Bruder] wohnt in Frankfurt.
My brother lives in Frankfurt.

'My brother lives in Frankfurt.'

(111) [$_{\text{ON}}$ Manfred] schreibt [$_{\text{OA}}$ einen Brief] [$_{\text{OPP}}$ an seine Mutter].
Manfred writes a letter to his mother.

'Manfred writes a letter to his mother.'

(112) [$_{\text{ON}}$ Manfred] schreibt [$_{\text{OD}}$ seiner Mutter] [$_{\text{OA}}$ einen Brief].
Manfred writes his mother a letter.

'Manfred writes his mother a letter.'

Another test is to try to replace the PP in question with an NP which clearly is an object. This is possible in sentence 111 which has the same

[4]Eisenberg (1999), p. 293–299, discusses those tests extensively.

meaning as sentence 112. There are, however, just a few verbs with which this is possible. There is, however, a common tendency in prepositional objects: The preposition in them has, to a certain extent, lost its usual meaning. Thus, in sentence 109 the preposition 'an' has lost most of its usual locative meaning (i.e. *at*) while in sentence 110 the preposition in the adverbial PP retains its usual meaning of *in.* This semantic criterion is, however, relatively vague as well because the extent to which the preposition has lost its usual meaning varies. Thus, the problem concerning the definition of prepositional objects as stated by Eisenberg (1999) remains:

> *Tatsächlich ist es noch niemandem gelungen, die Objekts- und Adverbialfunktion der PrGr* [Präpositionalgruppe; our remark] *durchgängig zu trennen.* (Eisenberg, 1999, p. 295)
>
> *In fact, no one ever succeeded in consistently separating object and adverbial functions of the prepositional group.*

The problem of the definition of prepositional objects is very important because it limits the performance of the annotation of OPPs. On the one hand, it is very challenging to consistently manually annotate OPPs for the gold standard, in the first place; and on the other hand, it is, consequently, very hard to manage to automatically annotate them just like in the gold standard. This is even more the case, if, like for our parser, the gold standard treebank (i.e. TüBa-D/Z) was constructed by a different institution than the sub-categorization lexicon IMSLex which was used for the automatic annotation. It can, thus, not be expected that the annotation of OPPs is able to reach the performance of the annotation of ONs or OAs.

7.4 Motivation for Annotation

We have selected the linguistic phenomena of sub-categorizable GFs for annotation because they are vital for the structure of the sentence. While most non-sub-categorizable GFs are optional, GFs sub-categorized by the verb or the adjective are the central elements of the sentence which are obligatory most of the times. An important point for sub-categorizable GFs is connected with the fact that they show relations in the sentence structure and can be seen as the interface between syntax and semantics. Alongside with semantic information, information about GFs can be used, e.g., for information extraction purposes because, very often, sub-categorizable GFs reveal the semantic structure of the sentence according to patterns like ‘who did what to whom’. In sentence 113, the triple ‘verb, ON, OA’ can be extracted which could yield a structure like ‘visit: Pope, Bukarest’. Together with other information from and/or about the text, this structure could be stored in a database to be used in information extraction. The annotation of sub-categorizable GFs can also be used for other applications like machine translation, summary generation or dialogue systems.

(113) Von heute bis Sonntag besucht $[_{\mathrm{ON}}$ Papst Johannes Paul II.] $[_{\mathrm{OA}}$
From today until Sunday visits Pope John Paul II
die rumänische Hauptstadt Bukarest].
the Romanian capital Bukarest.

‘Pope John Paul II is staying in the Romanian capital Bukarest from today until Sunday.’

An important point for the motivation to annotate GFs in German can be made by comparing the nature of GFs in German to their nature in English. While, in English, GFs are almost exclusively expressed by distribution, in German, GFs are expressed by a mixture of distributional and morphological information. This phenomenon reflects the fact that English is an analytic (or configurational) language while German is rather a synthetic (or non-configurational) language. Thus, while in English, the GFs can be abstracted from the constituent structure, this is not possible for German as can be seen in sentences 114 and 115. The subject in English can be defined as the NP directly dominated by the S, and the object as the NP directly dominated by the VP. This is, however, not possible for German in which, due to a more varied constituent order, a constituent like a VP cannot even be introduced without running into problems like crossing edges. Thus, GFs have to be explicitly marked in German to be able to search them in an annotated corpus.

(114) $[_{\mathrm{S}}$ $[_{\mathrm{NP}}$ Nobody] $[_{\mathrm{VP}}$ exceeds $[_{\mathrm{NP}}$ this actor]]].

(115) $[_{\mathrm{S}}$ $[_{\mathrm{VF}}$ $[_{\mathrm{NP}}$ Diesen Schauspieler]] $[_{\mathrm{VC}}$ übertrifft] $[_{\mathrm{MF}}$ $[_{\mathrm{NP}}$ niemand]]].
This actor exceeds nobody.

‘Nobody exceeds this actor.’

7.5 Methods of Annotation – An Outlook

For the annotation of GFs, the parser makes use of the already existing shallow annotation structure, the morphological information from DMOR and the sub-categorization frames from IMSLex (cf. figure 7.3). Thus, the parser makes use of a lot more information than KaRoPars, which basically uses PoS tag information for the shallow annotation. This raises the question whether it is possible to use the same annotation methods and tools for the GFs annotation as for the shallow annotation. For the shallow annotation in KaRoPars, an FSA approach was used which allowed a fast and robust annotation of these structures. Since the GFs annotation component is also intended to annotate large quantities of data, it would be reasonable to try to implement this task in a FSA approach as well.

A very crucial step in the annotation of GFs is the disambiguation of morphological information. In the ideal case, each chunk would have just one morphological reading (full disambiguation); in many cases, however, it is only possible to reduce the number of morphological readings for one chunk (partial disambiguation). The disambiguation process has to make use of the morphological information which was assigned by DMOR to each token and the shallow annotation structure assigned by KaRoPars. In order to maximally reduce the ambiguity of chunks, the morphological features of the tokens in each chunk have to be compared. In the case of verb-subject number agreement, the features of different chunks have to be compared. For this process, an FSA approach is not an adequate solution since it involves ranking the morphological features and changing of elements which occurred earlier in the parse tree. Another formalism will, thus, be used.

After the partial or full disambiguation of the morphological features, the annotation of GFs proceeds using distributional information and sub-categorization frames in addition to the already existing structure. In order to annotate the grammatical functions with FSAs, it has to be checked whether their annotation can be achieved by a regular expression grammar. If the sub-categorization information from IMSLex can be integrated into the corpus such that it is a feature of the verb, then it is possible to match the sequence of potential GFs and verb by a regular expression and thus apply FSAs for the annotation. Figure 7.8 shows a simplified example of how the linguistic data can be structured in such a way that it can be matched by a regular expression grammar. There are tuples of NC, field information, case information, on the one hand, and tuples of verb chunk, sub-categorization

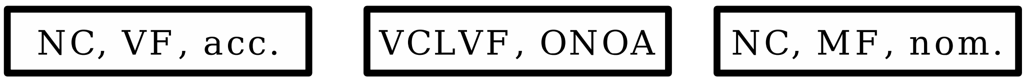


Figure 7.8: Sequence of noun chunk, verb chunk, noun chunk

frame, on the other hand (n.b. that the lexeme of the verb is not used). The example shows a pattern of an NC with the features VF (for occurs in VF) and accusative which is followed by a the finite verb in the VCLVF which sub-categorizes for the frame 'ON OA' which is again followed by an NC with the features MF and nominative.

Figure 7.8 shows that the linguistic information is structured in such a way that it can be matched by a regular expression grammar and, hence, the annotation can be achieved by FSAs. Since the morphological information is only partial in some cases, there remains the question of how to decide in cases in which more than one annotation solution would be possible. We have decided in favour of a ranked order of rules, in which the most typical ordering of GFs is tried first. The rules are applied until one matches. This ranking is the crucial disambiguation mechanism in the GF component.

Chapter 8

Reducing Morphological Ambiguity

ABSTRACT: Chapter 8 deals with the component in the annotation system which assigns morphological information to each token and reduces the number of morphological readings in the morphological ambiguity class of the noun chunks in order to allow and improve annotation of grammatical functions. Section 8.1 defines the task of the reduction of morphological ambiguity and introduces the approach which is used. Section 8.2 explains the details of the disambiguation component which disambiguates morphology using chunk-internal restrictions; section 8.3 explains the details of the disambiguation component working with chunk-external restrictions, i.e. restrictions on clause level. Section 8.4 deals with the enriching of those tokens with morphological information which were not dealt with by DMOR. Section 8.5 gives a conclusion of our approach and an outlook on the remaining task of GF annotation by comparing our approach to Hinrichs and Trushkina (2002).

8.1 Task Description

Since five out of seven grammatical functions we annotate are associated with case (i.e. ON, OG, OD, OA and PRED), the assignment of case information to the respective target noun chunks is an important step towards grammatical function annotation. In the case of OPPs, the preposition is an indicator of the functional potential of the chunk but only together with sub-categorization information of the verb. The same holds true of the subordinator and the clause type in case of OSs. In the case of the grammatical functions ON, OD and OA, however, the full disambiguation of the morphological features of a chunk would, in most cases, result in the detection of its grammatical function. Since the full disambiguation of the morphological features does, however, require distributional information and, very often, sub-categorization information, we just use two straightforward procedures to reduce the number of elements in the ambiguity class of the target chunks for grammatical function annotation. In the case of grammatical functions associated with case, the final disambiguation is achieved by the grammatical function annotation component.

The integration and partial disambiguation of the morphological information can best be illustrated by figure 8.1, which is another representation of the tree in figure 7.1 with morphological information added. The form of each token links it with a morphological potential, which, in the case of determiners, attributive adjectives and nouns, consists of a feature combination of 'case, number, gender'. We assign each token a set of such a feature combination using the tool DMOR (Schiller, 1995). The form 'der' of the definite article, for instance, is associated with a set of four feature combinations representing 'nominative, singular, masculine' and 'dative or genitive, singular, feminine' and 'genitive, plural, all genders'. This set of feature combinations is the morphological ambiguity class of the token. Since DMOR assigns the morphological information independently of any contextual information purely on the basis of its form and its PoS tag, the information of a token may be highly ambiguous in some cases and it is not possible to resolve this ambiguity without contextual information.

The set of all distinct morphological feature combinations of all tokens in a chunk can be seen as its undisambiguated morphological ambiguity class. It is possible to use the shallow annotation structure annotated by KaRoPars to reduce the number of elements in this ambiguity class. This can be achieved on two levels (cf. figure 8.2): On the one hand, the morphological features are compared within one chunk, i.e. chunk-internal; and, on the other hand, the morphological features of chunks are compared with the number feature of the respective verb in order to check whether a potential ON and the verb agree in number. This can be illustrated by the structure in figure 8.1. It shows the morphological ambiguity class of each token and the shallow annotation structure. Using the chunk structure it is, e.g., possible to reduce the morphological ambiguity class of the chunk 'der Erklärung' which contains

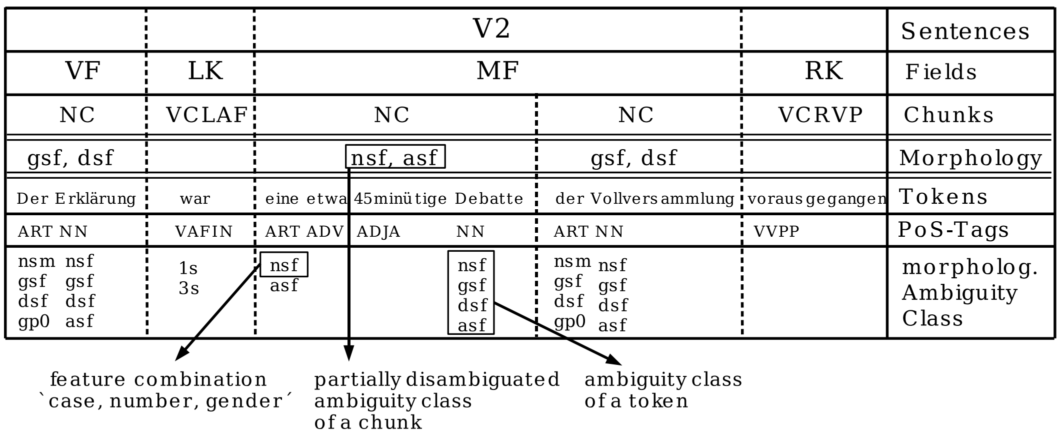


Figure 8.1: Reducing morphological ambiguity using shallow structure

six distinct feature combinations to the two feature combinations shown in the row 'Morphology'. The number of potential cases has been reduced from four to two. For the chunk-external ambiguity reduction, the clause and the field structure may be used since the finite verbs are assigned the feature combination 'person, number': In the case at hand, the finite verb is singular and, thus, all plural nominatives could have been discarded from the ambiguity classes.

It is, thus, the task of the morphological disambiguation component to reduce the morphological ambiguity of the potential grammatical functions using the morphological information assigned by the tool DMOR and the shallow annotation structure assigned by KaRoPars. Although full disambiguation is not possible using chunk-internal disambiguation and verb-ON number agreement, the achieved reduction considerably facilitates grammatical function annotation. This can be illustrated by figure 8.1 in which only one chunk retains the feature 'nominative' in its partially disambiguated ambiguity class although all chunks contain it in their undisambiguated ambiguity class. Since nearly all sentences contain ONs, this chunk can be seen as fully disambiguated because there is no way the following grammatical function annotation component would assign any other chunk than this one the grammatical function ON. Thus, no separate process is required which uses restrictions like this one to further reduce morphological ambiguity since these restrictions are checked by the grammatical function annotation com-

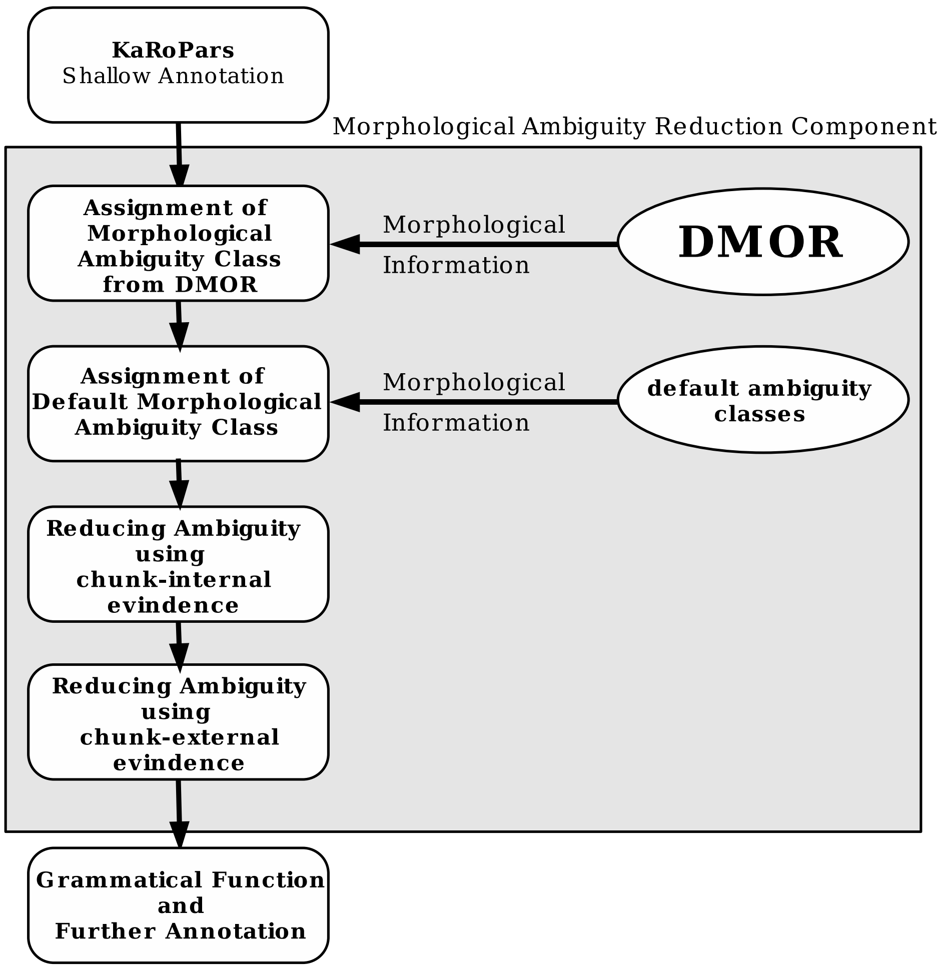

KaRoPars
Shallow Annotation
Morphological Ambiguity Reduction Component
Assignment of Morphological Ambiguity Class from DMOR
Morphological Information
DMOR
Assignment of Default Morphological Ambiguity Class
Morphological Information
default ambiguity classes
Reducing Ambiguity using chunk-internal evindence
Reducing Ambiguity using chunk-external evindence
Grammatical Function and Further Annotation

Figure 8.2: Structure of morphological disambiguation component and position in the annotation cascade

ponent.

8.2 Reducing Ambiguity on Chunk Level

The first component disambiguating morphology solely works chunk-internally, i.e. it just uses information available within the chunk for which it disambiguates the morphology. This makes it very fast and effective. On the basis of the morphological features of the individual tokens, the morphological features of the whole chunk are computed. The idea behind the process is that determiners, adjectives and nouns have to agree in number, gender and case within one noun chunk. Those feature combinations which occur with all relevant tokens in a chunk are, thus, those feature combinations which show the morphological potential of a chunk, i.e. the grammatical functions which a chunk might realize if regarded without contextual information. In order to compute the feature combinations which occur with all tokens we use a ranking method, in which we count the number of times a feature occurs with different tokens. The feature combination which occurs most frequently is then selected as the feature combination which is assigned to the whole chunk. In case the most frequent number of occurrences is occupied by more than one feature, the chunk is assigned the set of feature combinations occurring most frequently.

Table 8.1 presents an example of morphological disambiguation in a noun chunk. The noun chunk contains three tokens: the article ‘der’, the adjective ‘erste’ and the noun ‘Besuch’. The article may derive either from the lemma ‘der’ or the lemma ‘die’. DMOR assigns a different ambiguity class for each of the lemmas. In the case at hand, the nominative reading for the lemma ‘der’ and the genitive and dative readings for the lemma ‘die’. For each of the assigned morphological feature combinations of both lemmas, the ranking value is increased by 1. However, if a feature combination would occur in the ambiguity class of different lemmas and would, thus, be assigned more than once, it would still only be counted **once** per token. Counting it more than once would bias the ranking because, although the token might potentially be derived from two lemmas, it is actually only the instantiation of one lemma. Consequently, the ranking process covers the full morphological potential of a token evenly. This procedure has also been tested with the grammatical function annotation component and shown to be improving results.

Each instance of a feature combination as represented in table 8.1 is counted once. In the case at hand, the feature combination with the highest score is ‘nsm’, which is, in fact, the only feature combination which occurs in all three tokens (highlighted by a box in table 8.1). The token ‘der’ and ‘erste’ in table 8.1 contain a peculiarity: The feature combinations ‘gp0’, ‘np0t’ and ‘ap0t’ are unmarked for gender (as represented by the ‘0’ in the gender slot of the feature combination code). This means, however, that they may represent any of the three genders in German. In order to make possible a comparison between feature combinations which are unmarked for gender and those in which gender is explicitly marked, it is necessary to spell out the feature combinations which are unmarked for gender. (The added feature combinations are highlighted with italics in table 8.1.)

der	{$\boxed{\text{nsm}}$, gp0, gsf, *gpf*, *gpm*, *gpn*, dsf}
erste	{$\underline{\text{np0t}}$, nsf_, $\boxed{\text{nsmw}}$, nsnw, $\underline{\textit{npft}}$, $\underline{\textit{npmt}}$, $\underline{\textit{npnt}}$, $\underline{\text{ap0t}}$, asf_, asnw, $\underline{\textit{apft}}$, $\underline{\textit{apmt}}$, $\underline{\textit{apnt}}$}
Besuch	{$\boxed{\text{nsm\_}}$, dsm_, asm_}

Table 8.1: Chunk-internal disambiguation by ranking feature combinations

The principle of the ranking mechanism is very simple, but there are some more details to consider. In the actual ranking process, the component just considers case, number and gender. But there remains the feature ‘inflection type’ (which is represented by a fourth letter in some feature combinations). Adjectives and some (mostly deadjectival) nouns in German can be classified along the lines of the distinction between strong and weak forms (cf. Helbig and Buscha, 1999, section 3.1.1.). Sentence 116 and sentence 117 give examples: The first one shows a weak inflection type and the second one a strong one. In this case, the difference can just be seen in the adjective. There are five classes: In our morphological disambiguation component, they are represented by the features ‘strong’ (‘t’), weak (‘w’), ‘mixed’ (‘m’), ‘strong/mixed’ (‘T’) and ‘weak/mixed’ (‘W’). Since the distinction between strong and weak inflection type applies only to some nouns there is also the feature ‘unmarked for inflection type’ (‘_’).

(116)

Damals	feierten	sie	die	großen	Erfolge.
In those days	celebrated	they	the	great	successes.

‘In those days, they celebrated their great successes.’

(117)

Damals	feierten	sie	große	Erfolge.
In those days	celebrated	they	great	successes.

‘In those days, they celebrated great successes.’

The distinction between those two inflection types is important since they interact in a defined way. The definite article only occurs with the weak inflection type and the zero article occurs with the strong one (the indefinite article is mixed). For the example in table 8.1, this means that the features of the strong inflection type (those ending in ‘t’ and underlined) are irrelevant for the ranking since they cannot go together with the definite article. They are, thus, not included in the ranking process.

Another detail of the ranking process for morphological feature combinations is that, in our chunk definition, phrases which are center-embedded in another phrase form a chunk together with the non-recursive phrase they are embedded in. But, although these chunks are part of the embedding chunk, they do not contribute to the morphological analysis because they are morphologically autonomous. Consequently, the morphology of center-embedded chunks is ignored. The morphology of those chunks is of no importance for grammatical function annotation because they cannot act as one of the grammatical functions we annotate. Sentence 118 gives an example. The prepositional chunk ‘auf dieser Seite’ is a modifier of the adjective ‘erscheinenden’ and, thus, part of an adjectival chunk with it. This chunk is embedded in the chunk beginning with ‘die’ and ending with ‘LeserInnenbrief’. As regards the morphology, only the determiner ‘die’ the adjective ‘erscheinenden’ and the noun ‘LeserInnenbrief’ are relevant. Consequently, only these are taken into account for the morphological ranking.

(118)

[$_{\text{NC}}$ Die	[$_{\text{AJAC}}$ [$_{\text{PC}}$ auf	[$_{\text{NC}}$ dieser	Seite]]	[$_{\text{AJAC}}$ erscheinenden]]	
The	on	this	page	published	
LeserInnenbriefe]	geben	nicht	notwendigerweise	die Meinung	der
letters to the editor	give	not	necessarily	the opinion	of the
taz	wieder.				
taz	back.				

‘The letters to the editor published on this page do not necessarily represent the opinion of the taz [i.e. the newspaper].’

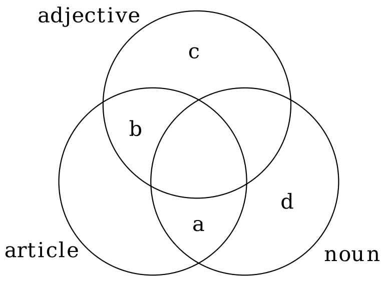


Figure 8.3: Intersection method – no result

Since the result of our disambiguation approach is typically the set of feature combinations which occur in all relevant tokens in a chunk, there remains the question why we did not choose an approach in which the intersection of the feature combinations of all relevant tokens is taken. The reason for this is that the ranking method is very robust because it also yields a result in the case that one of the relevant tokens does not contain the correct feature combination. A reason for a missing or a wrong feature combination may be a simple typing error in a word in the input sentence or an error in the DMOR analysis. Although we do assign default ambiguity classes in the case that DMOR has not assigned any morphology to a token, there might still be the possibility that DMOR has assigned a wrong ambiguity class. In this case the correct analysis would be missing and this would lead to system failure with the intersection method as outlined in the following example which is illustrated in table 8.2 and figures 8.3 and 8.4.

word category	morphological feature
article	$\{\boxed{a}, \boxed{b}\}$
adjective	$\{\boxed{b}, c\}$
noun	$\{\boxed{a}, d\}$

Table 8.2: Advantages of ranking over intersection

We envisage a hypothetical chunk with a determiner, an adjective and a noun and the correct morphological analysis of the chunk would be the feature combination ‘a’. However, the ambiguity class of the adjective would have been incorrectly assigned as {b, c} instead of {a, c}. The intersec-

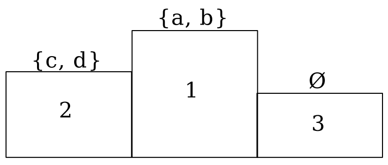
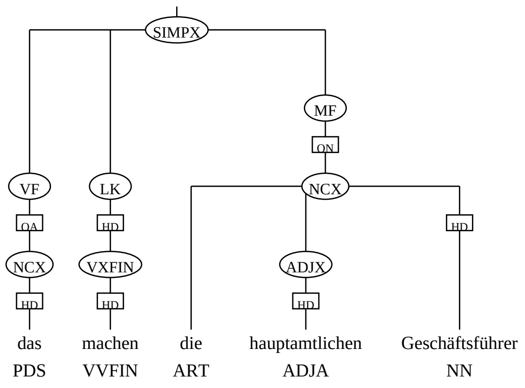


Figure 8.4: Ranking method – solution contained in result

tion method would yield no result (cf. figure 8.3) or – as a possible backup – the ambiguity class of all feature combinations {a, b, c, d}. The ranking method, however, yields {a, b} as a result, (cf. figure 8.4) since 'a' and 'b' both occur twice (highlighted by boxes in table 8.2) in contrast to the other feature combinations which occur just once. The result of the ranking method is an ambiguity class which contains the correct morphological feature combination 'a' and is smaller than the one put out by the possible intersection backup method. For the process of grammatical function annotation this considerable reduction of the ambiguity class might already be of great help although the result is still two features and one of them is wrong. However, since the grammatical function annotation component also relies on other factors like contextual information in its annotation decision, this robust treatment of morphological disambiguation allows working even with partially erroneous input.

8.3 Reducing Morphology on Clause Level

While the first component for the disambiguation of morphological information deals with restrictions on chunk level, the second component deals with a phenomenon which has a wider scope, namely the agreement in number between finite verb and ON, which ranges over the level of the clause. In German, the position of the ON in the clause is relatively free. It may be either in the VF or the MF of a clause. The finite verb may be either in the left part of sentence bracket (V1/V2 clauses) or in the right part of the sentence bracket (VL clauses). Since, at the point of disambiguation, it is not clear which of the noun chunks is the ON (as otherwise there would be no need for disambiguation), more than one noun chunk might be taken into account. Thus, the component has to consider the whole clause when disambiguating the morphology of the potential ON.



Das	machen	die	hauptamtlichen	Geschäftsführer.
That	do	the	full-time	managers

‘The full-time managers are taking care of that.’

Figure 8.5: Potential ONs are ambiguous as regards case features but not as regards number features

The disambiguation component works with the morphological ambiguity class which has been computed by the previous morphological disambiguation component dealing with ambiguity on chunk level. It checks the morphological feature combinations which contain the feature ‘nominative’ with respect to the feature ‘number’ against the number feature of the verb. In case they do not agree, the respective feature combination in the noun chunk is deleted. If, for instance, the finite verb just carries the feature ‘plural’, all the singular readings in the potential ONs are ruled out. Sentence 8.5 gives an example: Both noun chunks in the sentence have got the morphological potential to be either the ON or the OA of the sentence. The verb, however, demands a plural ON and, thus, ‘das’ is ruled out as the ON because it only carries the feature combination ‘nsn’.

Table 8.3 visualizes the process applied to the sentence in figure 8.5: The features ‘1p’ and ‘3p’ of the verb ‘machen’ show that the verb goes together with a nominative object with the feature combination ‘first person, plural’ or ‘third person, plural’. Thus, the singular reading of ‘das’ (i.e. nsn) can be excluded. It is deleted from the morphological ambiguity class of the verb. Since ‘das’ contains no nominative plural feature in its ambiguity class,

das	{<u>nsn</u>, asn}
machen	{1p, 3p}
die hauptamtlichen Geschäftsführer	{npm, apm}

Table 8.3: Using ON-finite verb number agreement for further morphological disambiguation

it no longer qualifies for ON. This process saves the grammatical function annotation component the task of checking ON-finite verb agreement. If the agreement had to be checked in the grammatical function annotation component, this would increase the size of this component and, thus, slow down the annotation process and make the system harder to manage.

8.4 Further Enriching the Corpus with Morphological Information

Some tokens are not assigned any morphological ambiguity class by the tool DMOR. This includes tokens with the PoS tag CARD (cardinal number) or FM (foreign material). For this reason, we assign these tokens a default morphological ambiguity class which comprises all the potential case, number and gender features (i.e. all four cases, singular or plural and all three genders). This is done because the grammatical function annotation component works with morphological features and, without any morphological information, it cannot assign any grammatical function. Even if the morphological feature combinations cannot be further reduced (e.g. by other tokens in the chunk), this information is still useful because the grammatical function annotation component also relies on distributional information and on the morphological ambiguity classes of the other potential grammatical functions.

Cardinal numbers (CARD) have not been assigned any morphology because they do not have any. We assume, however, that cardinal numbers (except ‘1’) in attributive adjective chunks carry the feature ‘plural’. Therefore, we assign all the feature combinations which include the feature ‘plural’ so that this information helps disambiguating the morphology of noun chunks in the chunk-internal component. Sentence 119 and sentence 120 give an example. The noun ‘Verdächtige’ may be singular or plural. Using the information that cardinals are plural deletes the singular readings which may lead to the deletion of some non-applicable cases like in sentence 121. The token ‘Teilnehmer’ may have the cases nominative, dative and accusative for its singular reading and nominative, genitive and accusative for its plural reading. Thus, if the reading singular is ruled out, OD can be ruled out as a potential grammatical function.

(119) 13 Verdächtige wurden von der Polizei festgenommen.
13 suspects were by the police arrested.

‘13 suspects were arrested by the police.’

(120) Verdächtige in diesem Fall war auch die Besitzerin des Ladens.
A suspect in this case was also the owner of the shop.

‘The owner of the shop was also a suspect in this case.’

(121) 350 Teilnehmer der zuvor friedlichen
350 participants of the before peaceful
”Reclaim the Streets”-Party wurden wegen Ordnungswidrigkeit
”Reclaim-the-Streets” party were because of regulatory offence
festgesetzt.
arrested.

‘350 participants of the ”Reclaim-the-Streets” party which was peaceful until then were arrested for regulatory offences.’

Furthermore, foreign material (FM) has not been assigned any morphology because it does not fit into the inflectional system of the German language. It does, however, receive the default ambiguity class because it may also act as (part of) a grammatical function (cf. sentence 122). In this case, the default ambiguity class is also of importance for the morphological disambiguation which again is important for grammatical function recognition. The same holds true of nouns which have been not been assigned an ambiguity class because they are, e.g., abbreviations like ‘USA’. These also receive the default ambiguity class.

(122) Die interne Credibility ist wichtig.
The internal credibility is important.

‘The internal ‘credibility’ is important.’

8.5 Conclusion, Outlook and Comparison with other Approaches

Our approach allows to reduce the number of elements in the morphological ambiguity class of a target chunk for grammatical function annotation. This partial disambiguation facilitates and improves grammatical function assignment because the ambiguity of potential grammatical functions is reduced. However, with the exception of verb-ON number agreement, we only use chunk-internal information to reduce ambiguity since chunk-external distributional information is integrated into the component following the morphological ambiguity reduction component, the grammatical function annotation component. Checking this information in the morphological ambiguity reduction component as well would lead to double processing of information and, thus, slow down the annotation process. The morphological ambiguity reduction component can, hence, be seen as a preprocessing step towards grammatical function annotation while the final disambiguation of morphology with regard to grammatical functions is the grammatical function annotation component.

A comparison with another approach further illustrates why some morphological ambiguity is not disambiguated in our morphological disambiguation component. Hinrichs and Trushkina (2002) try to disambiguate most of the morphological information in their morphological disambiguation component. The reason for this is the different goal of this component in their system. Hinrichs and Trushkina (2002) use a dependency based approach for syntactic annotation. In this approach "morphological disambiguation is a crucial step in narrowing down the search space for the correct assignment of dependency structures". By contrast, in our approach, we strongly use distributional information for the annotation of grammatical functions. The partial morphological disambiguation is, in fact, a crucial step for the annotation of grammatical functions but the final decision about the annotation of grammatical functions is taken by the component which makes use of distribution and sub-categorization frames of the verb. Since the annotation of grammatical functions is also a decision about morphology for the grammatical functions ON, OG, OD, OA and PRED, the final morphological disambiguation is done by that component and not in the morphological disambiguation component. An example illustrates the difference to the approach of Hinrichs and Trushkina (2002).

In sentence 123, the chunk containing the personal pronoun 'wir' is unambiguously nominative while the other chunk may be of any case (cf. table 8.4)[1]. This means for the other noun chunk(s) that they cannot be nominative since a clause typically just contains one nominative chunk unless there is a copula verb in the clause. Hinrichs and Trushkina (2002) use this information to eliminate all nominative readings in all other noun chunks than the unambiguous one. Our morphological disambiguation component does not deal with cases like this because the ambiguity in the chunks is finally disambiguated in the grammatical function annotation component. The verb 'finden' sub-categorizes (among other frames) for the frame 'ON OA PRED_ADJ' (i.e. it takes a nominative and an accusative object and a predicative adjective). By using the ambiguity class of the respective chunks in conjunction with distributional features, we assign grammatical functions. The assignment of the grammatical function ON makes it clear that the chunk is nominative and the grammatical function OA shows that the chunk is accusative.

(123) Es ist wichtig, daß $[_{\text{ON}}$ wir] $[_{\text{OA}}$ Candan Ercettin] $[_{\text{PRED}}$ gut]
It is important that we Candan Ercettin good
finden.
consider.

'It is important that we like Candan Ercettin.'

(124) Es ist egal, daß $[_{\text{OA}}$ Candan Ercettin] $[_{\text{ON}}$ keiner] $[_{\text{PRED}}$ gut]
It is all the same that Candan Ercettin nobody good
findet.
consider.

'It does not matter that nobody likes Candan Ercettin.'

wir	{np}
Candan Ercettin	{gs, ds, as, np, gp, dp, ap}

Table 8.4: Unambiguous and ambiguous morphological information

[1]In the feature combinations, the feature 'gender' is now left out in the representation since it is no longer of any importance in grammatical function annotation.

Candan Ercettin	{ns, gs, ds, as, gp, dp, ap}
keiner	{ns}

Table 8.5: Unambiguous and ambiguous morphological information

Our system first applies the standard constituent order ON OA PRED if the respective chunks include suitable morphological features in their ambiguity class (possibly among some other morphological features). In the case at hand, this strategy would succeed and correctly assign the grammatical functions and, thus, fully disambiguate the morphology. It is, however, possible that there is another distribution than the standard constituent order. However, this would not change anything in the case at hand. Sentence 124 shows another distribution with the same morphological ambiguity classes.[2] In this case, our system would not assign the standard constituent order. The first noun chunk 'Candan Ercettin' does in fact contain 'nominative' in its morphological ambiguity class, but the second one does not contain accusative in its ambiguity class (cf. table 8.5). Consequently, our system would not try to assign the chunk 'Candan Ercettin' the label 'ON'. Thus, the correct annotation is guaranteed without the previous disambiguation of the chunk 'Candan Ercettin'.

Since we use a different approach, many of the cases disambiguated in Hinrichs and Trushkina (2002) in the morphological disambiguation component are disambiguated by our grammatical function annotation component. While Hinrichs and Trushkina (2002) need the morphological annotation at an early stage for their dependency annotation, we can leave the morphology partially ambiguous (underspecified) until it is resolved in the grammatical function annotation component.

[2]The reason that sentence 124 is lexically different from sentence 123 is that sentence 123 with the constituent order of sentence 124 would only be marginally acceptable.

Chapter 9

Annotating Grammatical Functions (GFs)

Abstract: Section 9 deals with the annotation of GFs using the shallow annotation structure, the reduced morphological ambiguity classes and subcategorization (SC) frames for the relevant verbs and adjectives. Section 9.1 introduces the basic issues relevant for the annotation of GFs (cf. figure 9.1). Section 9.2 explains how lexical entries are converted into finite-state automata (FSAs). In section 9.3, we present the mechanism which allows us to disambiguate between different annotation possibilities with respect to linear order; and section 9.4 shows the disambiguation mechanism for the case that it is possible to apply different SC frames. Section 9.5 describes the linguistics principles which are fed into the disambiguation mechanism presented in section 9.3. In section 9.6, we illustrate how we extend our approach to cover cases in which there is no SC frame for the respective verb or adjective. Section 9.7 shows how our approach can be modelled with the help of OT constraints.

9.1 Task Description

It is our goal to annotate GFs using finite-state (FS) transducers. We make use of the shallow structure added by the shallow parsing component and of the morphological ambiguity class reduced by the morphological reduction component. Furthermore, we use the SC frames contained in IMSLex (Eckle-Kohler, 1999). The basic issues in the annotation of GFs using FSTs are illustrated in figure 9.1. They will be discussed in this section in the order in which they emerge in the construction of the parser. The first problem is how to convert the lexical entries which contain the SC information into FS rules which cover the different instantiations of a SC frame.

One of the main issues of any annotation system is the way ambiguous structures are dealt with. Thus, a further central issue is the problem of ambiguity in the occurrence and linear order of GFs. First, there is the question of the occurrence of different GFs for verbs and adjectives which have more than one SC frame. Second, there is the question of ambiguity in the linear order of GFs which arises because the ambiguity in morphology is reduced but not resolved, and GFs in German can occur in various linear orders. Third, another difficulty is how to deal with verbs or adjectives which do not have any SC frame at all in the IMSLex. This problem concerns both the occurrence and the order of GFs since the problem of ambiguous morphology and missing SC frames is likely to co-occur, at times. A last, very central issue is the question of which mechanism to use in order to disambiguate structures which are ambiguous because of the afore mentioned points.

Transforming lexical entries into FSTs A lexical entry for a verb or an adjective consists of the lexical item itself and of one or more sets of GFs (i.e. the SC frames) it sub-categorizes for. For each set of GFs, there is more than one realization in the language. There are two reasons for this: First, the GFs may be realized in different contexts because they can occur in sentences of different types or functions like V1, V2 or VL clauses and, e.g., interrogative clauses, relative clauses or affirmative clauses. Second, there may be different linearizations of the GFs in the clauses. Both phenomena can be seen in sentences 125–127. As a consequence, more than one FST has to be generated from each SC frame. There are two crucial points as regards the transformation of the lexical entries into FSTs: First, a decision has to be made to which extent the conversion should be done manually and to which extent it can be done automatically. Second, it has to be decided whether the automatic conversion should be done before the annotation process or *on-the-fly* during the process.

(125)	[$_{\text{V1}}$ Versteht	[$_{\text{OA}}$ ihn]	[$_{\text{ON}}$ keiner]?]
	Understands	him	no-one?

‘Does not anybody understand him?’

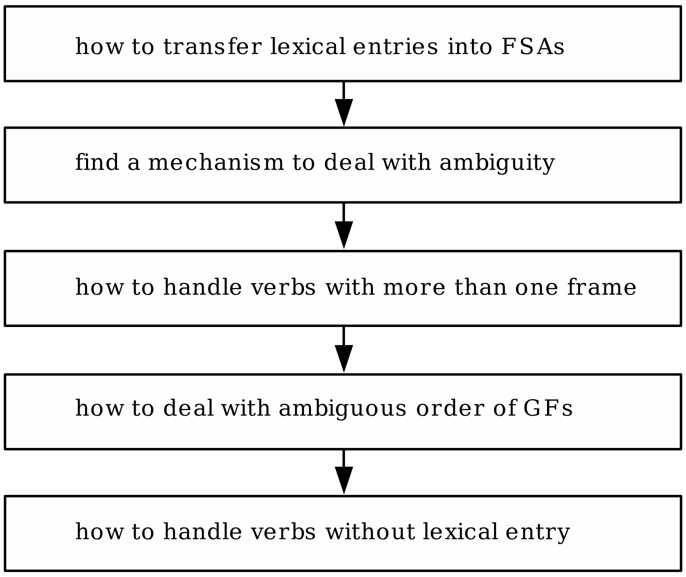

how to transfer lexical entries into FSAs
find a mechanism to deal with ambiguity
how to handle verbs with more than one frame
how to deal with ambiguous order of GFs
how to handle verbs without lexical entry

Figure 9.1: Basic issues in the annotation of grammatical functions in the order of their emergence in the annotation process

(126)	[$_{V2}$ [$_{ON}$ Klaus]	versteht	[$_{OA}$ seinen	Freund].]	
	Klaus	understands	his	friend.	

'Klaus understands his friend.'

(127)	Ich	glaube,	[$_{VL}$ dass	[$_{ON}$ er]	[$_{OA}$ seinen	Freund]	versteht].
	I	think	that	he	his	friend	understands.

'I think that he understands his friend.'

Some additional SC frames can be derived in a strictly defined way from the SC frame(s) associated with a verb. This phenomenon is called diathesis or alternation (cf. Levin, 1993) and it is defined by lexical rules. The passive is an example of a diathesis: If a verb sub-categorizes for an ON and an OA, it can be deduced that it also sub-categorizes for an ON and an optional OPP in the passive. Sentences 128 and 129 give examples: The verb 'abdrehen' demands an ON and an OA in the active. In the passive, the OA of the active sentence has become the ON of the passive sentence and the ON of the active sentence has become an optional OPP (with the preposition 'von') of the passive sentence.

(128) [$_{\text{ON}}$ Klaus] drehte [$_{\text{OA}}$ den Verschluss] ab.
Klaus took the seal off.

'Klaus took off the seal.'

(129) [$_{\text{ON}}$ Der Verschluss] wurde ([$_{\text{OPP}}$ von Klaus]) abgedreht.
The seal was (by Klaus) taken off.

'The seal was taken off (by Klaus).'

Some but not all of those diatheses are contained in the IMSLex. As regards the treatment of diatheses, there are the same questions as with the different instantiations of a SC frame, i.e. whether they should be implemented manually or automatically and whether they should be generated before annotation or *on-the-fly*. Meurers and Minnen (1997) show that it is possible to model lexical rules and their interaction into FSAs. It would, thus, also be possible to integrate this process into the system just using the power of FSAs.

Furthermore, there is the question which diatheses can be safely integrated into the parser. The passive diathesis is a very unequivocal example of a diathesis. It is not contained in the IMSLex (nor typically contained in any other lexicon), and it can be reliably generated because there are few restrictions on it. The same is true of diatheses like *object deletion* which can be applied to all verbs which sub-categorize for the SC frame 'ON OA'. Object deletion describes the fact that the OA can be dropped without the sentence becoming ungrammatical (cf. sentences 130 and 131). In cases like these, the diathesis can simply be added to all verbs which sub-categorize for one distinct SC frame.

(130) [$_{\text{ON}}$ Steffi] schießt [$_{\text{OA}}$ den Ball].
Steffi kicks the ball.

'Steffi kicks the ball.'

(131) [$_{\text{ON}}$ Steffi] schießt.
Steffi kicks.

'Steffi kicks (it).'

This is different for other diatheses like the *dative-shift* (Dativierung). The dative-shift describes a rule according to which verbs which sub-categorize for 'ON OD OA' also sub-categorize for 'ON OA OPP' (cf. sentences 132 and 133). However, the application of this rule is far more restricted. While, as regards the passive, only the preposition ‘von’ is possible in the OPP, various prepositions are possible for the OPP, as regards the dative-shift. In sentence 133 the OPP is realized with the preposition ‘an’ and in sentence 135 with the preposition ‘zu’. Furthermore, dative-shift can only be applied to some verbs. Dative-shift is, for instance, not possible for ‘geben’ (cf. sentences 136 and 137). While a diathesis like the passive is, thus, to a large extent solely restricted to the SC frame, a diathesis like dative-shift depends on the lexical item in question (here: the verb).

(132) [$_{\text{ON}}$ Stella] schickt [$_{\text{OD}}$ ihrer Freundin] [$_{\text{OA}}$ einen Brief].
Stella sends her friend a letter.

‘Stella sends her friend a letter.’

(133) [$_{\text{ON}}$ Stella] schickt [$_{\text{OA}}$ einen Brief] [$_{\text{OPP}}$ an ihre Freundin].
Stella sends a letter to her friend.

‘Stella sends a letter to her friend.’

(134) [$_{\text{ON}}$ Stella] bringt [$_{\text{OD}}$ ihrer Freundin] [$_{\text{OA}}$ einen Kuchen].
Stella brings her friend a cake.

‘Stella brings her friend a cake.’

(135) [$_{\text{ON}}$ Stella] bringt [$_{\text{OA}}$ einen Kuchen] [$_{\text{OPP}}$ zu ihre Freundin].
Stella brings a cake to her friend.

‘Stella brings a cake to her friend.’

(136) [$_{\text{ON}}$ Stella] gibt [$_{\text{OD}}$ ihrer Freundin] [$_{\text{OA}}$ ein Buch].
Stella gives her friend a book.

‘Stella gives her friend a book.’

(137) * [$_{\text{ON}}$ Stella] gibt [$_{\text{OA}}$ ein Buch] [$_{\text{OPP}}$ an ihre Freundin].
Stella gives a book to her friend.

‘Stella gives a book to her friend.’

The example of the dative-shift has shown that it is not always possible to straightforwardly derive the diathesis of a SC frame. In contrast to the passive, in which a single lexical rule can be used for a large group of verbs, there has to be a distinct lexical rule for every single verb for the SC frame ‘ON OD OA’. In this case, the respective diathesis could just as well be stored in the lexicon. Treating it with a lexical rule would neither theoretically nor technically be a simplification. For this reason, the diathesis of the dative-shift is contained in the lexical entry of some verbs in the IMSLex. Since the IMSLex is an automatically generated lexicon, there are, however, also some diatheses missing and it would be possible to use the lexical rules to generate them. This is, however, only reasonable if the respective information is available electronically.

Finding a disambiguation mechanism A disambiguation mechanism is crucial because the ambiguity class of the morphological features of a chunk is just reduced and not resolved. Thus, in those cases in which more than one FST would match a structure, preference has to be given to one of them. There are two important points with regard to this problem: First, the criteria according to which the preference is given have to be specified and, second, a mechanism has to be devised which triggers off the respective FST according to the specified criteria. Since we have chosen to use an FS approach for well-founded reasons, it is important that the disambiguation mechanism fully complies with the FS approach.

Treatment of lexical items with more than one SC frame The verb 'abdrehen' can also serve as an example for a lexical item with more than one SC frame. Sentences 138–143 give examples. Like nearly all lexical items, 'abdrehen' requires at least an ON in all its instantiations. Additionally, sentences 138 and 139 show it with a reflexive accusative object with and without an OPP. Sentences 140 and 141 show 'abdrehen' with a simple accusative object and, again, with and without an OPP. In sentence 142 the verb sub-categorizes for an OD and an OA. Only in sentence 143, 'abdrehen' just sub-categorizes for an ON. The crucial question as regards this phenomenon is how to deal with verbs which allow more than one instantiation of GFs in cases in which both instantiations would be possible as regards the potential GFs. One of the issues in dealing with such verbs is whether some of the SC frames can be collapsed if some GFs are simply regarded as optional like in the sentence pairs 138, 139 and 140, 141 or whether these instantiations should be regarded as disjunct. The fact that some of the SC frames are connected with different meanings of the verb cannot be taken into account because semantic information is not taken into consideration in our parser.

(138) [$_{\text{ON}}$ Die Schraube] dreht [$_{\text{OA}}$ sich] ab.
The screw turned itself off.

‘The screw turned off.’

(139) [$_{\text{ON}}$ Die Schraube] dreht [$_{\text{OA}}$ sich] [$_{\text{OPP}}$ von dem Blech] ab.
The screw turned itself from the plate off.

‘The screw turned off the plate.’

(140) [$_{\text{ON}}$ Kim] drehte [$_{\text{OA}}$ die Schraube] ab.
Kim turned the screw off.

‘Kim turned off the screw.’

(141) [$_{\text{ON}}$ Kim] drehte [$_{\text{OA}}$ die Schraube] [$_{\text{OPP}}$ von dem Blech] ab.
Kim turned the screw from the plate off.

‘Kim turned off the screw from the plate.’

(142) [$_{\text{ON}}$ Kim] drehte [$_{\text{OD}}$ dem Blech] [$_{\text{OA}}$ die Schraube] ab.
Kim turned the plate the screw off.

‘Kim turned off the screw from the plate.’

(143) [$_{\text{ON}}$ Das Flugzeug] drehte ab.
The plane peeled off

‘The plane peeled off.’

A special case of verbs having more than one SC frame are *support verbs* (‘Funktionsverben’). In a support verb construction, the verb has to a large extent lost its original meaning and mostly serves as a medium of morpho-syntactic features while the lexical content lies mainly in the GF (cf. Helbig and Buscha, 1999). The GFs for which a support verb sub-categorizes are lexically restricted. What is important from the point of view of the annotation of GFs is that the SC frame of a verb in its non-support verb function and in its support verb function may differ. Thus, support verbs cannot reliably be annotated just using the non-support verbs SC frames.

Sentences 144 and 145 give an example of a support verb which has the same SC frame as its lexical verb counterpart. The lexical verb ‘aufnehmen’ sub-categorizes for an ON and an OA (sentence 144). The support verb ‘aufnehmen’ also sub-categorizes for an ON and an OA. However, the support verb ‘aufnehmen’ can only sub-categorize for OAs with lexical heads like ‘Beziehung’(relation), ‘Kontakt’(contact), ‘Verbindung’(contact) or ‘Verhandlungen’(negotiations) (cf. sentence 145). These cases are unproblematic because the support verb readings are syntactically a subset of the lexical verb readings and can, thus, be annotated by the lexical verb component.

(144) [$_{\text{ON}}$ Das Schiff] nimmt [$_{\text{ON}}$ einen weiteren Passagier] auf.
The ship takes a further passenger in.

‘The ship takes in a further passenger.’

(145) [$_{\text{ON}}$ Die Regierung] nimmt jetzt [$_{\text{OA}}$ Verhandlungen] auf.
The government takes now negotiations up.

‘The government takes up negotiations, now.’

Some support verbs do however differ in their SC frame from their lexical verb counterparts. Sentence 146 shows the support verb ‘ziehen’, which demands an ON, an OA and an OPP. The OPP is restricted to the preposition ‘in’ and some few nouns. But the lexical verb ‘ziehen’ just demands an ON and an OA. Thus, unlike in the case with the verb ‘aufnehmen’, the SC frame for the lexical verb ‘ziehen’ does not suffice for the support verb ‘ziehen’. The annotation component would consequently fail to annotate the grammatical functions correctly. Besides the question of how to acquire the support verbs SC frames, which are not contained in the IMSLex, there remains the question of integrating this information into the parser.

(146) [$_{\text{ON}}$ Mein Nachbar] zieht [$_{\text{OA}}$ eine Klage] [$_{\text{OPP}}$ in Betracht].
My neighbour takes a lawsuit into consideration.

‘My neighbour takes legal action into consideration.’

(147) [$_{\text{ON}}$ Die Kinder] ziehen [$_{\text{OA}}$ einen Wagen] die Straße hoch.
The children pull a cart the street up.

‘The children are pulling a cart up the street.’

Treatment of ambiguous order of GFs We have used the fact that the linear order of some constituents in German is syntactically restricted for the shallow annotation. The constituents which are syntactically restricted and were, thus, annotated by the shallow parser subsume noun chunks, prepositional chunks, adjective chunks and adverbs chunks and also the constituents of the sentence bracket, i.e. the verbal elements and the subordinators. While the structure of the topological fields and the internal structure of chunks is syntactically restricted, the linear order of chunks (and their grammatical functions) within the topological fields is varied, i.e. it varies with respect to certain factors which are not purely syntactical (cf. Eisenberg, 1999, sec. 13.1.2). From the parsing point of view, this is precisely the reason why the possible linearizations of GFs in a sentence are ambiguous and why we have chosen to handle this task with another component than the shallow structures. Since the case features of a potential GF cannot be fully disambiguated but just reduced using local information, the linearization of GFs can neither be fully determined by syntactic features nor by case features. Sentences 148–150[1] illustrate the varied linearizations of GFs which have to be covered by the parser. In the case at hand, only the chunk 'das Bild' is ambiguous since, as regards the case features, it could be ON or OA.

(148)	[$_{ON}$ Der Schüler]	hat	[$_{OD}$ dem Lehrer]	[$_{OA}$ das Bild]	beschrieben.
	The pupil	has	the teacher	the picture	described.

'The pupil has described the picture to the teacher.'

(149)	[$_{OD}$ Dem Lehrer]	hat	[$_{ON}$ der Schüler]	[$_{OA}$ das Bild]	beschrieben.
	The teacher	has	the pupil	the picture	described.

'The pupil has described the picture to the teacher.'

(150)	[$_{OA}$ Das Bild]	hat	[$_{OD}$ dem Lehrer]	[$_{ON}$ der Schüler]	beschrieben.
	The picture	has	the teacher	the pupil	described.

'The pupil has described the picture to the teacher.'

There may, however, also be sentences in which all potential chunks are ambiguous as regards their case features. Sentence 151 shows such a sentence. From a point of view of the case features, 'eine Schauspielerin' can be ON or OA, 'Peter' may be any nominal GF (except genitive), and 'die Regisseurin' may be either ON or OA. The reading shown here is the most natural one, but ON and OA might also be inverse. This can be seen in sentence 152, in which 'der Regisseur' is unambiguously ON. Sentence 153, if compared to sentence 151, shows again that a morphological change in one potential GF can affect more than one GF. The reason for this is that names are

[1]The three sentences have the same truth value but different functions from a pragmatic point of view. The three translations do not differ because the pragmatic aspect of the utterances cannot be covered without contextual information.

typically morphologically unmarked (unless for genitive). This may lead to high ambiguity if more than one name occurs in a sentence. This is not uncommon in newspaper text.

(151) [$_{\text{ON}}$ Eine Schauspielerin] hat [$_{\text{OD}}$ Peter] [$_{\text{OA}}$ die Regisseurin]
An actress has Peter the director-fem
beschrieben.
described.

'An actress has described the director to Peter.'

(152) [$_{\text{OA}}$ Eine Schauspielerin] hat [$_{\text{OD}}$ Peter] [$_{\text{ON}}$ der Regisseur]
An actress has Peter the director-masc
beschrieben.
described.

'The director has described an actress to Peter.'

(153) [$_{\text{OD}}$ Einer Schauspielerin] hat [$_{\text{ON}}$ Peter] [$_{\text{OA}}$ die Regisseurin]
An actress has Peter the director-fem
beschrieben.
described.

'Peter has described the director to an actress.'

Since the linearization of GFs is not syntactically restricted and their morphology cannot be reduced any further, it is necessary to detect the laws which determine the linearization of GFs in German and to scrutinize whether they can be integrated into an FS parser. Grammars of German agree that there are rules which determine the linearization of the GFs. As regards the order of complements and adjuncts, Engel (1996), sec. S 4.0, states: "Ihre [die der Folgeelemente] Anordnung unterliegt im Deutschen präzisen Regeln".[2] It is, thus, more precise to speak of a *varied* linearization of GFs than of a *free* one. However, grammars do not agree about the extent as to which these rules and their interaction can be covered. Eisenberg (1999), sec. 13.1.2, however, states: "Welche Faktoren auf welche Weise die Abfolgeregularitäten im Mittelfeld bestimmen und wie sie zusammenwirken, ist nur teilweise geklärt".[3] It is precisely the interaction of the factors or

[2] *In German, their* [i.e. complements and adjuncts] *order is subject to precise rules.*"
[3] *It is only partially clarified which factors in which way determine the order regularities in the middle field and how they interact.*

principles for the order of GFs in German which makes formalizing rules for the linearization of GFs a problem. In addition, both Engel (1996) and Eisenberg (1999) often speak of tendencies rather than rules when they refer to order principles.

Lenerz (1977), sec. 1.3, posits an unmarked GF order ("unmarkierte Abfolge"). However, the GF order is only subject to "weak rules" and may, thus, be violated under certain conditions. If this is the case, it has to be scrutinized what these conditions are and how they can be integrated into the FS parser. Furthermore, Lenerz (1977), sec. 4.4.2, notices that the unmarked GF order is different with some verbs. Thus, it has to be checked whether this phenomenon can also be integrated into the parser. Höhle (1982) rejects the findings of Lenerz (1977) about the unmarked order of GFs as being of "very doubtful relevance" ("äußerst fragliche Relevanz") because the findings are not based on assumptions of stylistically normal constituent orders but on structural assumptions which do not take into account pragmatic aspects. It has to be analyzed in how far the objections in Höhle (1982) challenge the findings of Lenerz (1977) or whether Lenerz (1977) can still be used as a mechanism for the disambiguation of GF order.

Furthermore, Eisenberg (1999), sec. 13.1.2, gives eight different principles for the order of GFs. These are the tendency of ONs to precede ODs and OAs or the tendency for *theme* to precede *rheme*. Further principles are 'focus follows non-focus', 'pronoun precedes non-pronoun', 'definite precedes indefinite', 'animate precedes inanimate', 'departure precedes destination' (as regards directions) and 'longer constituents follow shorter ones'. As regards our task, it is the question which of these principles can be made accessible to an FS parser. One obvious problem is that some of the concepts used for phrasing principles are relatively vague like the *theme–rheme* dichotomy or the longer–shorter distinction, and another problem is that the principles refer to different areas of linguistic description like syntax, semantics, morphology or information structure.

Treatment of missing SC frames Since a SC lexicon only contains a finite number of lexical items, there will always be lexical items in the corpus for which there is no SC frame. The reason for this may be that lexicon is incomplete or that the lexical item is a neologism. Thus, a strategy has to be found to deal with those cases. There may be different solutions to this problem: Lexical items without a SC frame could be assigned default

SC frames. This process could be enriched by semantic information about classes of verbs. On the other hand, the problem could be approached from the perspective of the potential GF and it could be tried to assign GFs independent of the items they are sub-categorized by. Furthermore, the relation between the component using SC frames from the lexicon (henceforth lexical component) and the component working without SC frames (henceforth non-lexical component) has to be specified. It is the question whether the non-lexical component should solely work on sentences not treated by the lexical component or whether it should also refine sentences already seen by the lexical component.

9.2 Converting Lexical Entries into FSTs

We extract the lexical entries which we convert into FSTs from the IMSLex electronic lexicon, which contains SC frames for verbs and adjectives (among other information) (cf. Eckle-Kohler, 1999). There are 32,597 SC frames for 12,688 verbs (2.6 on average) and 3,314 SC frames for 2224 adjectives (1.5 on average) in the IMSLex version we use (cf. figure 9.2 for an example). In order to integrate the sub-categorization information from the IMSLex into the FSTs, it is necessary to transform the lexical entries into the formalism of the FSAs. Figure 9.2 shows a sample lexical entry for the verb 'abdrehen'. The entry contains various pieces of information. The lemma contains information about the morphological structure of the verb. In the case at hand, it shows that 'ab' is a separable verb affix (*abtrennbarer Verbzusatz*). Thus, the verb may be distributed over both parts of the sentence bracket. The main information for our purposes are the GFs for which the verb sub-categorizes which are encoded as a feature combination of function (e.g. subject), category (e.g. NP) and case (e.g. nominative). Optional GFs are not explicitly marked. In those cases, the verb is coded once with and once without the optional GF.

Figure 9.3 shows the lexical entry in our format. Each string separated by a blank represents a SC frame as used by our parser. The string 'ONOA', e.g. shows that the verb sub-categorizes for an ON and an OA. Some of the concepts in IMSLex had to be transfered into the relevant concepts in our system, e.g. *subject* into *nominative object*. We left some information implicit, e.g. that an ON is realized by an NP but an OPP by a PP. We also discarded some information, e.g. the *haben* or *sein* variants of the auxiliary

```
ab#drehen haben RULE1 (subj(NP_nom),arg(PRON_refl-acc)) SADAW
ab#drehen haben RULE1 (subj(NP_nom),arg(PRON_refl-acc),p-obj(PP_von_dat)) SADAW
ab#drehen haben RULE1 (subj(NP_nom),obj(NP_acc)) HGC
ab#drehen haben RULE1 (subj(NP_nom),obj(NP_acc)) SADAW
ab#drehen haben RULE1 (subj(NP_nom),obj(NP_acc),iobj(NP_dat)) HGC
ab#drehen haben-variant SADAW (subj(NP_nom)) HGC
ab#drehen sein-variant SADAW (subj(NP_nom)) HGC
```

Figure 9.2: Lexical entry for the verb 'abdrehen' in IMSLex format

```
ab#drehen :: ON ONOA ONODOA ONOA_refl ONOA_reflOPP_von
```

Figure 9.3: Lexical entry for 'abdrehen' in our format

and the origin of the lexical entry, e.g. HGC for "Huge German Corpus". Some SC frames could thus be collapsed into one. In the case of OPPs, the preposition they are restricted to is specified and in the case of OSs the subordinator. Some GFs are further specified if they, e.g., require the head word to be a reflexive pronoun or dummy 'es'.

A central question for the conversion of lexical entries into FSTs is how to link the verb with the respective SC frame. It is one possibility to convert every verb-SC frame pair into FSTs which match the various linearizations of the GFs of this verb-SC frame pair. In the case of the verb 'abdrehen' in figure 9.3, there would be five sets of FSTs. However, this procedure would yield a far too large grammar and it is, thus, not feasible. A straightforward solution to link the verb with its respective SC frame(s) is to tag the verb with the SC frame(s) just like tokens are tagged with PoS tags. Just like the shallow syntactic structures can be annotated just using the PoS tags of the tokens without taking into account the lexical items, the FSTs for the annotation of GFs can be applied just taking into account the sub-categorization potential of a verb as it is included in its SC frame tag. The SC frame tagging is a simple lexical look-up which can be integrated into the parser as one FST. As a result, just one set of FSTs per SC frame has to be generated and the grammar is kept much smaller. Furthermore, this procedure keeps the parser modular and flexible since it allows the integration of more than one SC lexicon.

Figure 9.4 shows a schematic representation of the transformation of SC frames into annotation patterns which serve as the grammar of the FSTs.

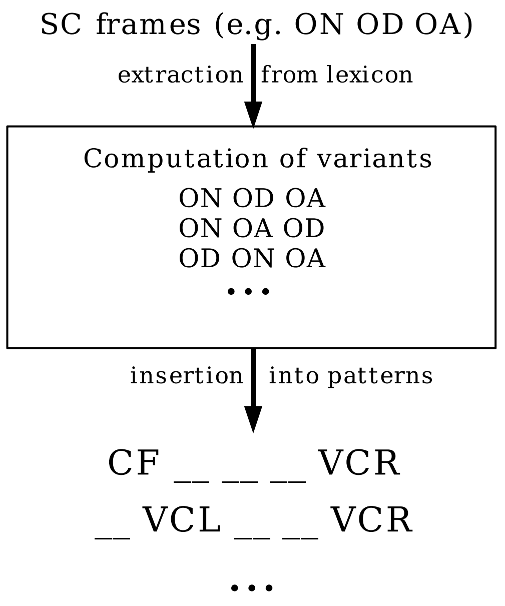


Figure 9.4: Transformation of SC frames into annotation patterns for FSTs

The SC frames are extracted from the IMSLex and all variants of the linearization of GFs are computed. These variants are inserted into different patterns which match the already existing shallow annotation structure. Thus, 6164 patterns are yielded, 1205 for VL clauses and 4959 for V1/V2 clauses. (Another 2812 patterns are yielded for the support verb component, 583 for VL clauses and 2229 for V1/V2 clauses.) If these patterns are applied to a sentence, each of the recognized target chunks will be assigned its respective GF label.[4] Thus, the shallow annotation structure serves as an ideal pre-requisite to keep the complexity of the GF annotation component as low as possible. Chunks serve as potential targets of GFs and the topological fields and clause structure reduces and defines the search space.

The annotation pattern in figure 9.5 and the sentence in figure 9.6 may serve as an example to illustrate the connection between SC frame, the annotation patterns and the structures to be matched. The pattern matches a sequence of elements in the shallow annotation structure which has the potential to be the realization of the GFs which are in the respective SC frame.

[4]NB Optional elements are also treated by the parser but left out here for reasons of clarity.

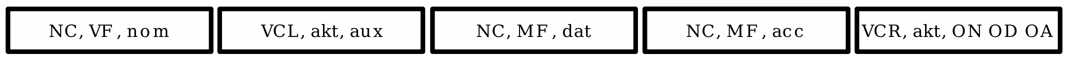


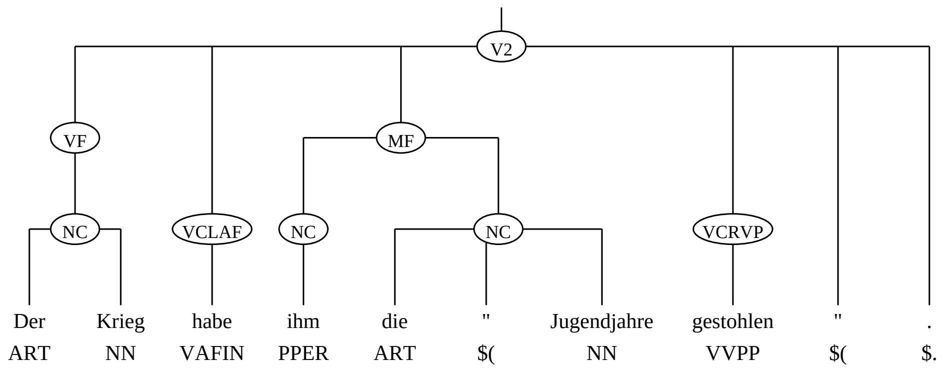
Figure 9.5: Pattern with integrated SC frame for V2 clause

If the pattern matches, the GFs are assigned. The first element in the pattern matches a noun chunk in the VF with the feature ‘nom’ in its morphological ambiguity class. This element matches the chunk ‘der Krieg’ in figure 9.6. If the whole pattern is matched, then this element assigns the GF ON to ‘der Krieg’. The following element matches a verb chunk as the left part of the sentence bracket which contains an auxiliary verb. This chunk just matches a structure but does not assign anything. The following two elements match the potential OD ‘ihm’ and the potential OA ‘die Jugendjahre’ in the MF. The last element matches a verb chunk as the right part of the sentence bracket which contains a verb which is SC-tagged ‘ONODOA’. This is true of ‘gestohlen’ in figure 9.6 (which is the past participle of ‘stehlen’) because the SC frame ‘ONODOA’ is one of the frames it sub-categorizes for.

Patterns like the one in figure 9.5 have to be realized for all possible instantiations of a SC frame. The patterns are simple sequences of elements. The elements are realized as feature combinations. Some of those feature combinations differ depending on the actual pattern. The sub-categorizing verb can, for instance, be realized in the left part of the sentence bracket or in the right one. Some of the information, however, is constant. This is the information contained in the SC frame. The context of the GFs may be different as the pattern in figure 9.7 and the sentence in figure 9.8 show but the GFs themselves and the verb are an invariant set.

Figure 9.7 shows the pattern which matches the sentence in figure 9.8. The first element matches the subordinator in the CF. The feature ‘fin’ means that the element just matches subordinators introducing finite VL clauses, i.e. not the subordinator ‘um’, which introduces non-finite VL clauses. The following three elements match the GFs in the MF of the clause. The last element matches the right part of the sentence bracket which contains the verb, which has been assigned the SC frame for ON OD OA. Further patterns for annotation would include other functions of clauses like non-finite or relative VL clauses of other types of clauses like V1 clauses.

The annotation of diatheses like the passive plays a special role in the transformation of lexical entries into FSTs. While in the pattern discussed



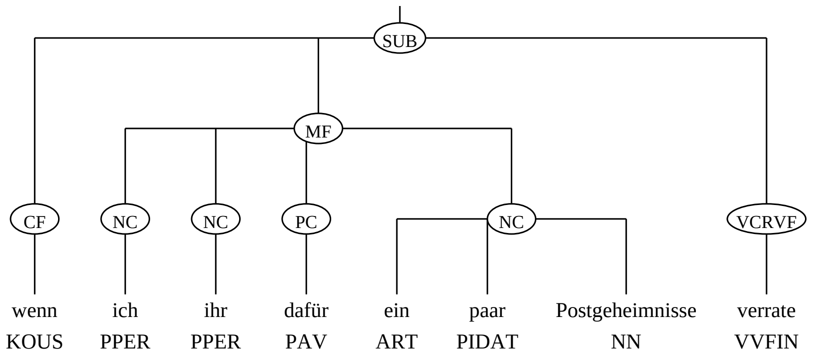
$[_{ON}$ Der	Krieg]	habe	$[_{OD}$ ihm]	$[_{OA}$ die	Jugendjahre]	gestohlen.
The	war	have	him	the	days of his youth	stolen.

‘The war has stolen him the days of his youth.’

Figure 9.6: Instantiation of the SC frame ‘ON OD OA’ in a V2 clause

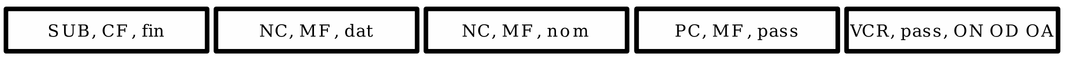
SUB, CF, fin	NC, MF, nom	NC, MF, dat	NC, MF, acc	VCR, akt, ON OD OA

Figure 9.7: Pattern with integrated SC frame for VL clause



wenn [$_{\text{ON}}$ ich] [$_{\text{OD}}$ ihr] dafür [$_{\text{OA}}$ ein paar Postgeheimnisse]
if I her for that a few secrecies of the post
verrate
give away
'if I give away some secrecies of the post to them [i.e. the mafia]'

Figure 9.8: Instantiation of the SC frame 'ON OD OA' in a VL clause



wenn [$_{\text{OD}}$ ihr] dafür [$_{\text{ON}}$ ein paar Postgeheimnisse] [$_{\text{OPP}}$ von mir]
if her for that a few secrecies of the post by me
verraten würden
give away were
'if some secrecies of the post were given away to them [i.e. the mafia] by me'

Figure 9.9: Pattern for passive with integrated SC frame for VL clause

above the number and type of GFs is constant[5] and just the linearization changes, in the passive for the SC frame for 'ON OD OA', there is an ON, an OD and an optional OPP with the preposition 'von' as illustrated in the pattern in figure 9.9. The type of GFs, thus, differ. The fact that the OA of the active clause has become the ON of the passive clause and the ON of the active clause the OPP of the passive clause is of no importance for the structure of the annotation pattern. Figure 9.9 shows one annotation pattern for the passive of the frame 'ON OD OA' in a finite VL clause. The SC frame, which is checked in the right part of the sentence bracket, is still 'ON OD OA'. However, the feature 'pass' shows that the pattern just matches those verbal structures which express a passive. The GFs which are checked by the pattern are those corresponding to the passive diathesis of the frame 'ON OD OA' as figure 9.9 shows (which also gives the passive version of the sentence in figure 9.8 as an example). The element with the feature combination 'PC, MF, pass' matches an optional OPP with the preposition 'von'.

A central question as regards the construction of the grammar for the annotation of the GFs is whether the process which computes the variants of the linearization of the GFs and the process which inserts them into the patterns which match the different types and functions of clauses (cf. figure 9.4) can be automated. In principle, this is possible. It is, however, not clear whether an automatic conversion gives the system an advantage over a manual conversion. On the one hand, with an automatic conversion, it is possible to generalize over the linearization of GFs in different SC frames and in different types of clauses. On the other hand, this generalization is not always possible because the linearization of GFs differs depending e.g. on the type of clause and the diathesis. In V2 clauses, which have a VF, the linearization is different from VL clause, which do not have a VF. Furthermore, in passive clauses the linearization is different from active clauses. We have, thus, chosen to perform the processes which transform the lexical entries into FSTs manually.

This gives us the full control over the different instantiations of the lexical entries, which is a clear advantage for the maintainability of the system, since, in case of errors in the test corpus, it can be checked which rule has caused the error and, in the ideal case, the error can be mended. This would be more complicated if the rules were automatically generated because it would not be clear which mechanism had generated the rule in question. The individual

[5]An exception is the non-finite clause in which the ON is not realized.

ziehen :: ONOAOPP_in_(Betracht|Erwägung|Mitleidenschaft|Vertrauen|Zweifel)

Figure 9.10: Lexical entry for support verb ‘ziehen’

NC, VF, nom	VCL, akt, ziehen	NC, MF, acc	PC, MF, in_Betracht, Erwägung, ...

[$_{ON}$ AEG]	zieht	[$_{OA}$ eine	Klage]	[$_{OPP}$ in	Betracht].
AEG	takes	a	lawsuit	into	consideration

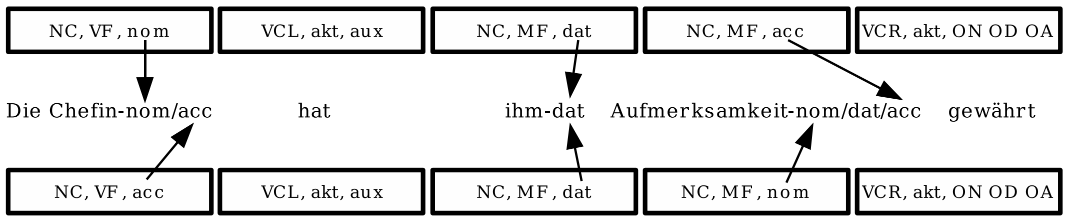
‘AEG takes a lawsuit into consideration.’

Figure 9.11: One annotation pattern for the support verb ‘ziehen’

transformation of the lexical entries allows this modular treatment of the problem, which includes advantages like easier maintainability and also easier evaluation. If rules are generated by a complex mechanism, then it is much harder to single out certain rules to evaluate their effect. It is, thus, a lot better to keep complex interaction of different components low and to keep components as modular as possible.

A special case of the conversion of lexical entries into FSTs are the support verbs. Since support verbs are not included in IMSLex, a special lexicon was compiled. We used Helbig and Buscha (1999), which have an extensive list of support verbs and their SC frames. It is not possible to use the same strategy as with the lexical verbs for the support verbs because the SC frames for the support verbs are lexically restricted (cf. figure 9.10). In fact, the relation between verb and SC frame is a one-to-one relation. Thus, a set of annotation patterns had to be generated for every single support verb. This was done automatically using the lexical entries as in figure 9.10 and the annotation patterns for the respective lexical verbs which were automatically enriched with the lexical information for the support verbs.

Figure 9.11 gives an example: The pattern is structured like the one for the SC frame ‘ON OA OPP_in’. However, instead of the SC frame of the verb, the verb itself is encoded in the element matching the verb chunk. Furthermore, the PC matching the OPP is restricted to the preposition ‘in’ **and** the few nouns which the support verb sub-categorizes for. Since there is a set of FSTs for every support verb, the support verb component is considerably large. It comprises 27% of all annotation patterns in the system.



[$_{\text{ON}}$ Die Chefin] hat [$_{\text{OD}}$ ihm] [$_{\text{OA}}$ Aufmerksamkeit] gewährt
The boss-fem has him attention paid

'The boss has paid him attention'

Figure 9.12: One structure may be matched by more than one pattern

9.3 Disambiguation by Ranking Rules

Ambiguity is one of the basic problems in NLP. Every grammar has to be able to deal with it. For our task, ambiguity arises because a shallow annotation structure which is to be annotated with GFs can in many cases be matched by more than one of the patterns which have been described in section 9.2. This can be illustrated by the structure in figure 9.12 which can be matched by at least two different annotation patterns. If the patterns for the SC frame 'ON OD OA' are applied, the NCs 'die Chefin' and 'Aufmerksamkeit' can both be either the ON or the OA because they both include the case features nominative and accusative in their ambiguity class. There are, thus, at least two possible analyses for the sentence. Since we just want to put out one analysis, we need to decide for one of them. There are two important points to consider here: the criteria for the decision, and the mechanism by which these criteria are implemented.

There are various ways to decide between different analyses for one structure: One way is to apply a machine learning approach and learn from instances in a corpus and then decide for the most probable one. In this case, the instances are the decision criteria and the calculation of the most probable one is the decision mechanism. Another approach is to apply constraints to certain linguistic phenomena and to rank and/or count the constraints and, thus, decide for a certain analysis. Then the constraints are the criteria and the calculation of them is the mechanism. In our approach, the criteria are linguistic knowledge about e.g. the unmarked order of GFs, and

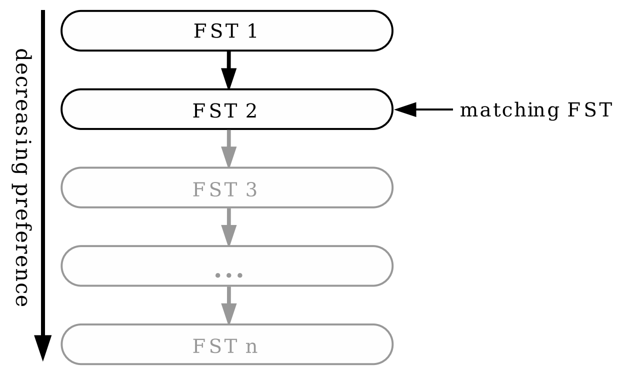


Figure 9.13: Ranked cascade of FSTs

the disambiguation mechanism is a ranked order of FSTs in which e.g. the most unmarked annotation pattern is tried first and the following annotation patterns are applied in an order of increasing markedness until one of them matches.

We have decided in favour of this mechanism because it can easily be integrated into an FS approach. The annotation patterns are arranged in a cascade of FSTs which are ranked according to the specified criteria and which are applied until one of them matches. After this, further annotation is blocked. In contrast to some approaches, the parser, thus, just generates one analysis. There is also no component which calculates the correct analysis since the decision is simply inherent in the cascade of FSTs.

9.4 A Longest-Match Strategy for the Disambiguation of SC Frames

Since a verb can sub-categorize for more than one SC frame, a strategy has to be found with regard to what SC frame to apply in such cases. Some SC frames are disjunct, i.e. the GFs contained in them are not identical. In those cases, the annotation is not ambiguous. In other cases, however, the complements of one SC frame constitute a subset of the complements of another frame and, consequently, one structure could be matched by patterns which belong to more than one frame. The different analyses of the sentence in 154 and 155 give an example: The verb 'halten' can occur with the SC frame 'ON OA OPP', and it can occur with the SC frame 'ON OA' (cf. figure 9.14). We, thus, rank the patterns for the SC frame 'ON OA OPP' higher in the hierarchy of the annotation patterns than the patterns of the SC frame 'ON OA' so that they are the preferred annotation pattern. In the annotation cascade these patterns will then be tried first (cf. figure 9.15) and, if one of them applies, further annotation is blocked. This strategy can be compared to the longest-match strategy in chunking. It avoids losing complements like the OPP 'für eine tragbare Lösung' in figure 154.

halten :: ONOAOPP_für ONOA

Figure 9.14: The Lexical Entry for the Verb 'halten' (extract)

(154)

Angesichts	der	unterschiedlichen	Nutzungsinteressen	halte	$[_{ON}$
In the light of	the	various	utilization interests	regarded	
er]	$[_{OA}$ das]	$[_{OPP}$ für	eine	tragbare	Lösung].
he	that	as	an	acceptable	solution.

'In the light of the various utilization interests, he considered this an acceptable solution.'

(155)

Angesichts	der	unterschiedlichen	Nutzungsinteressen	halte	$[_{ON}$
In the light of	the	various	utilization interests	regarded	
er]	$[_{OA}$ das]	für	eine	tragbare	Lösung.
he	that	as	an	acceptable	solution.

'In the light of the various utilization interests, he considered this an acceptable solution.'

9.5 Using Markedness to Disambiguate GF Order

Since the potential GFs are not fully morphologically disambiguated, we need further criteria to disambiguate them. We have decided to use the concept of *markedness* as the crucial criterion for the disambiguation of GFs. Since our disambiguation mechanism is the ranking of the more preferred linearizations over the less preferred ones, the most unmarked linearizations are applied first and the remaining linearizations are applied in the order of increasing markedness. This is done for every single SC frame (cf. figure 9.15).

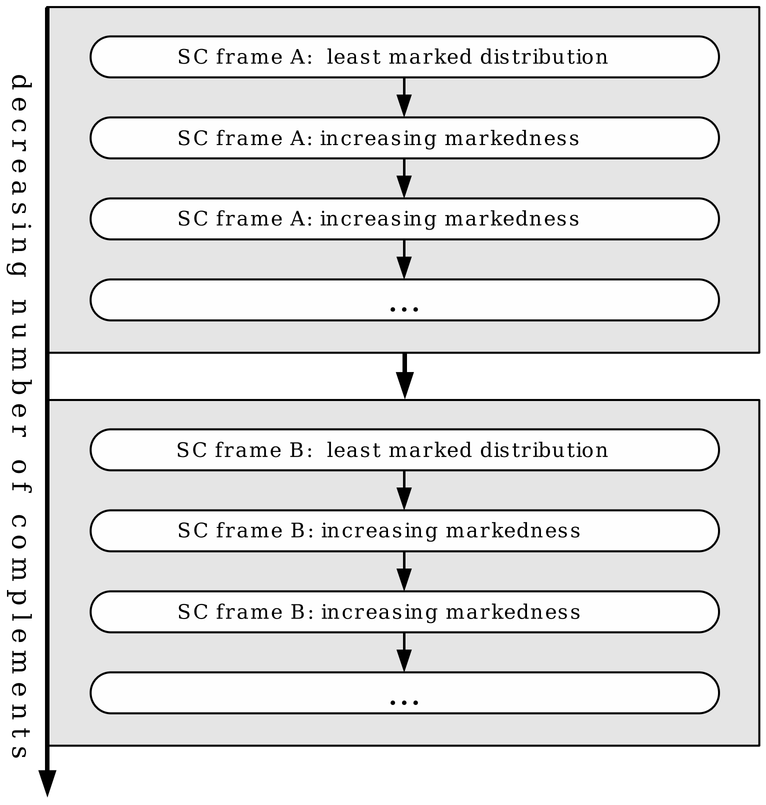

decreasing number of complements
SC frame A: least marked distribution
SC frame A: increasing markedness
SC frame A: increasing markedness
...
SC frame B: least marked distribution
SC frame B: increasing markedness
SC frame B: increasing markedness
...

Figure 9.15: Decreasing number of complements and increasing syntactic markedness in the ranking of FSTs

There are two questions with regard to the concept of markedness and the annotation of GFs. The first question is how one should find the unmarked order of GFs and the second question is why one should take the concept of markedness as a criterion for the disambiguation of GFs. As regards the first question, Lenerz (1977) gives a precise definition:

> *Wenn zwei Satzglieder A und B sowohl in der Abfolge AB wie in der Abfolge BA auftreten können, und wenn BA nur unter bestimmten, testbaren Bedingungen auftreten kann, denen AB nicht unterliegt, dann ist AB die "unmarkierte Abfolge" und BA die "markierte Abfolge".* (Lenerz, 1977, p. 27)

If two constituents A and B can occur in the sequence AB and in the sequence BA, and if BA can only occur under defined and testable conditions which AB is not subject to, then AB is the "unmarked order" and BA the "marked order". (our translation)

The definition can be illustrated by the examples given in Lenerz (1977). The assumption is that the indirect object (OD) precedes the direct object (OA) in the unmarked GF order. Lenerz (1977) gives two sets of example sentences:[6]

(2) $[_{OD}$ Wem] hast $[_{ON}$ Du] $[_{OA}$ das Geld] gegeben?
Whom have you the money given?

'To whom did you give the money?'

(2) a) $[_{ON}$ Ich] habe $[_{OD}$ *dem Kassierer*] $[_{OA}$ das Geld] gegeben.
I have the cashier the money given.

'I gave the money to the cashier.'

(2) b) $[_{ON}$ Ich] habe $[_{OA}$ das Geld] $[_{OD}$ *dem Kassierer*] gegeben.
I have the money the cashier given.

'I gave the money to the cashier.'

(3) $[_{OA}$ Was] hast $[_{ON}$ Du] $[_{OD}$ dem Kassierer] gegeben?
What have you the cashier given?

'What did you give to the cashier?'

(3) a) $[_{ON}$ Ich] habe $[_{OD}$ dem Kassierer] $[_{OA}$ *das Geld*] gegeben.
I have the cashier the money given.

'I gave the money to the cashier.'

(3) b) ?* $[_{ON}$ Ich] habe $[_{OA}$ *das Geld*] $[_{OD}$ dem Kassierer] gegeben.
I have the money the cashier given.

'I gave the money to the cashier.'

[6]The sentences are taken from Lenerz (1977), p. 43. The rheme is high-lighted by italics. We have added our GF annotation.

$\boxed{\text{ON} \quad \text{OD OA} \quad \text{OPP}}$

Figure 9.16: Unmarked order of GFs

Sentence (2) is a question asking for the OD, and sentence (3) is a question asking for the OA. Consequently, in sentences (2) a) and b), the OD is the rheme, i.e. it is the constituent in the sentence which expresses the largest amount of extra meaning (cf. Crystal, 1997, → *rheme*) and in sentences (3) a) and b), the OA is the rheme (highlighted by italics). Sentences (2) a) and b) show that the OD and the OA can occur in either sequence. If, however, the OA is the rheme, the sequence 'OA OD' is only marginally grammatical. Thus, the linearization 'OA OD' can only occur under defined and testable conditions ('Thema-Rhema-Bedingung', *theme–rheme condition*) which the linearization 'OD OA' is not subject to, i.e. it can only occur if OA is not the rheme. Thus, the linearization 'OD OA' is the unmarked order. The 'Definitheitsbedingung' (*condition of definiteness*) is another test which confirms the order 'OD OA' as the unmarked order. This test shows that the order 'OA OD' is only possible under the condition that the OA is not indefinite and is, thus, the marked order (cf. sentence (19) b) from Lenerz (1977), p. 54). These tests can be extended such that finally the unmarked order of GFs is 'ON OD OA OPP' (cf. Eisenberg, 1999, sec. 13.1.2).

(19) b)	* [$_{\text{ON}}$	Ich]	habe	[$_{\text{OA}}$ ein	Buch]	[$_{\text{OD}}$ dem	Schüler]	geschenkt.
		I	have	a	book	the	pupil	given.

'I gave the pupil a book as a present.'

Höhle (1982) criticizes Lenerz (1977) for not taking the pragmatic aspects of a sentence into consideration when determining the unmarked GF order. According to Höhle (1982), p. 134, in sentences like (2) a) and b), both GF orders are stylistically normal (i.e. unmarked). Sentence (2) a), however, has stylistically non-normal accent. Furthermore, Höhle (1982), p. 135, critizes that sentence (3) b) is not tested in any other context than the one provided and, thus, put to the pragmatic test. Still, we follow Lenerz (1977) to determine the unmarked order of GFs since we cannot take pragmatic aspects into consideration because they would take us far beyond the FS formalism. Furthermore, Höhle (1982), p. 137, admits that Lenerz (1977) shows the GF order which is “subject to no structural constraints” (“keinen strukturellen Einschränkungen unterliegen”).

Since the unmarked order of GFs can be precisely defined with Lenerz (1977), there remains the question of why one should take the concept of markedness as the criterion for the disambiguation of GFs. The reason is that this criterion is the only one which directly relates to the order of the GFs in the sentence. All other criteria may only be applied on the basis of the unmarked order. Sentence (2) a) from Lenerz (1977), for instance, is the unmarked order of the SC frame ‘ON OD OA’, although the OD is the rheme and there is a tendency for the rheme to be after the theme. In sentence (2) b), this is, in fact, the case and the unmarked order is changed. Thus, the theme–rheme condition applies on the basis of the unmarked word order and can be an indicator for the existence of a marked order. It is, however, not possible to make predictions about the order of GFs independent of the unmarked GF order. Furthermore, as sentence (2) a) has shown, the theme–rheme condition may also have no effect at all.

The problem that they may only make predictions about the order of GFs on the basis of the unmarked order is a problem which is inherent in all the principles (cf. Eisenberg, 1999, sec. 13.1.2) influencing the GF order – apart from the fact that, in order to apply them, one is in need of information exceeding the syntactic and morphological information we are using. The *theme–rheme principle*, the *focus–non-focus principle*, the *animate–inanimate principle* and the *departure–destination principle* require at least semantic information; but we do not want to make use of such kind of information because it would considerably enlarge the size of the parser and would thus slow it down. However, even the principles which might be captured with the information available to us, i.e. the *pronoun–non-pronoun principle* and the *definite–indefinite principle* do not enhance the disambiguation process as sentences 156–161 show.

(156) Sieht [$_{\text{ON}}$ der Vater] [$_{\text{OA}}$ den Sohn]?
Sees the father the son?

‘Does the father see the son?’

(157) Sieht [$_{\text{OA}}$ ihn] [$_{\text{ON}}$ der Vater]?
Sees him the father?

‘Does the father see him?’

(158) Sieht $[_{\text{ON}}$ der Vater$]$ $[_{\text{OA}}$ ihn$]$?
Sees the father him?

'Does the father see him?'

(159) Sieht $[_{\text{ON|OA}}$ die Mutter$]$ $[_{\text{OA|ON}}$ die Tochter$]$?
Sees the mother the daughter?

'Does the mother see the daughter?'

(160) Sieht $[_{\text{ON|OA}}$ sie$]$ $[_{\text{OA|ON}}$ die Mutter$]$?
Sees her the mother?

'Does the mother see her?'

(161) Sieht $[_{\text{ON|OA}}$ die Mutter$]$ $[_{\text{OA|ON}}$ sie$]$?
Sees the mother her?

'Does the mother see her?'

In sentences 156–158 the situation is clear because the potential GFs are morphologically unambiguous. Sentence 156 shows the unmarked GF order with proper nouns. This GF order is only altered if any of the afore mentioned principles can be applied. This is, for instance, the case in sentence 157 in which the OA precedes the ON because the OA is a pronoun. However, the unmarked word order is still possible as sentence 158 shows. This might be the result of another principle conflicting with the *pronoun–non-pronoun principle*, e.g. the *focus–non-focus principle.* Thus, even if we apply the principles which are within the scope of our annotation system, it is not guaranteed that we succeed in predicting the correct GF order.

Sentences 159–161 show a more fundamental problem with the principles. In contrast to sentences 156–158, in sentences 159–161 the GF order is ambiguous because both NCs may be either nominative or accusative. However, the unmarked GF order is 'ON OA'. In sentence 160, the first NC is pronominal. If we regard sentence 160 in contrast to sentence 159, we might conclude that the GF order is 'OA ON' according to the *pronoun–non-pronoun principle.* However, in a corpus, sentences do not occur together with their permutations. If we regard sentence 160 independent of sentence 159, we might just as well assume that it is the ON which was pronominalized and that the sentence is still in its unmarked order. Thus, in case of ambiguity, the principles which only indirectly apply to the GF order do not add any further information to the disambiguation process. Therefore, we only use the unmarked GF order as a criterion for GF disambiguation.

(162) $[_{\mathrm{OD}}$ Hamburg] droht $[_{\mathrm{ON}}$ Neuwahl]
Hamburg threatens re-election

'Hamburg threatened with rerun of election'

Lenerz (1977) furthermore points out that there are some classes of verbs like 'psychic verbs' (cf. Lenerz, 1977, sec. 4.2.2.) which do not adhere to the typical unmarked GF order. An example can be seen in sentence 162 in which the GF order is 'OD ON'. This is, in fact, the unmarked GF order of the verb 'drohen'. The reason for the untypical GF order is that the 'Mitteilungszentrum' (*center of information*) is in the OD. This difference is not coded in the IMSLex. A further problem is that some of those verbs have more than one reading, which cannot be distinguished without taking into consideration semantic information. The evaluation will show in how far the typical unmarked GF order is still an adequate concept for the annotation of GFs.

9.6 Extending Disambiguation to non-Lexical Patterns

In the previous sections, we have described how we annotate GFs using FSTs and SC frames. The SC frames for lexical verbs were taken from the IMSLex and the SC frames for the support verbs from Helbig and Buscha (1999). However, for some verbs in the corpus, there are no SC frames in the lexicon and, thus, the parser fails to assign GFs. Since we intend to construct a robust parsing system, we have to deal with those cases as well. While the lexical component uses the SC frames stored in the lexicon to trigger off the annotation of grammatical functions, the non-lexical component does not use any such kind of information. It solely relies on the morphological information disambiguated by previous components and on linearization features.

Since the lexical component relies on more specific information because it uses SC frames in addition to morphological and linearization features, it can annotate GFs more reliably. Thus, it is applied first following an *easy-first parsing* strategy. Since the support verb component works with

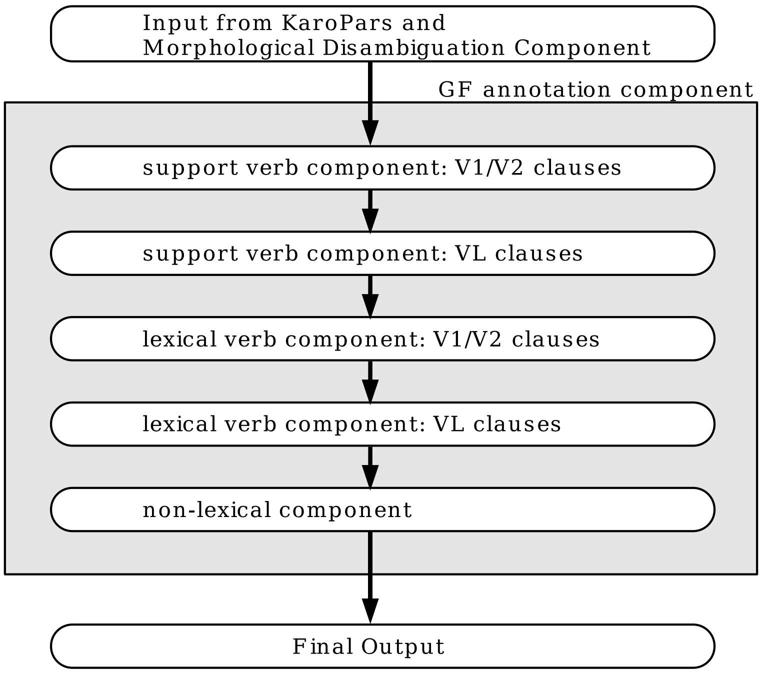

Input from KaroPars and
Morphological Disambiguation Component
GF annotation component
support verb component: V1/V2 clauses
support verb component: VL clauses
lexical verb component: V1/V2 clauses
lexical verb component: VL clauses
non-lexical component
Final Output

Figure 9.17: Annotation cascade

lexically restricted SC frames and is, thus, more specific than the lexical verb component, it is applied first in the lexical component. The non-lexical component is applied last and only if the lexical component fails and no annotation has been added to a structure (cf. figure 9.17). By contrast to the other two components, the non-lexical component just has very few rules, i.e. 41, which is about 0.5% of the rules of the whole GF parser. (6164 patterns are generated from the IMSLex and 2812 from the support verb lexicon.)

The component deals with nominative, accusative and dative objects and with predicatives. It does not deal with genitive objects since they are too rare to be annotated reliably without SC information. It does not deal with prepositional and sentential objects, either, since these objects occur in similar contexts as homonymous adjuncts. Generally speaking, the non-lexical component does not deal with grammatical functions which are ambiguous with constituents which occur more often in another function. This is not the case for nominal chunks (except for genitives) since nominal chunks do not frequently occur as adjuncts.

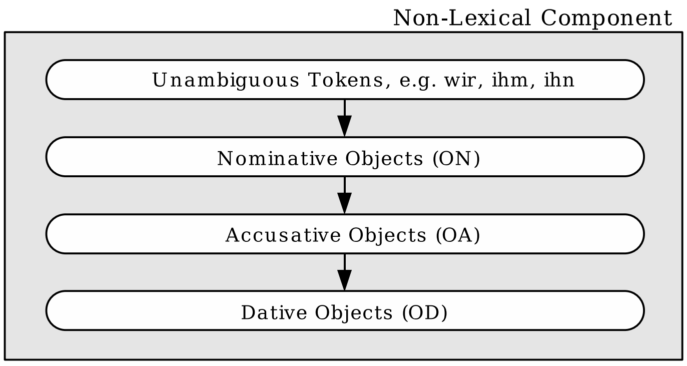


Figure 9.18: Inner structure of the non-lexical component

In the non-lexical component, two ranking strategies are pursued: At first, rules are ordered according to the frequency of the grammatical functions which they annotate, and, within the rules for one grammatical function, there is an annotation hierarchy in which rules are ordered according to the degree of ambiguity of the candidate constituent to be annotated. The sequence of annotation of the grammatical functions is ON, OA, OD. Within the modules for each function, the rules are applied until one of them matches. In between the modules, the architecture is a cascade of annotation, in which the output of the preceding module is the input to the following. That way, e.g. ONs are annotated first and OAs, which are very often ambiguous with ONs, afterwards, because ONs are more frequent and also more crucial for a sentence than OAs.

In each module for a grammatical function, there are, at the top of the annotation hierarchy, all those NCs which contain tokens which can unambiguously be assigned a grammatical function. For ON, these tokens subsume the personal pronouns 'ich, du, er, wir', the interrogative pronoun 'wer' and the indefinite pronouns 'irgendwer, jemand, man'. For OA, there is the personal pronoun 'ihn' and the reflexive pronoun 'mich'. For OD, there are the personal pronouns 'ihm, ihnen', the relative pronoun and the article 'dem', and the token 'mir', which may be either a reflexive or a personal pronoun. Only after the annotation of these unambiguous grammatical functions does the annotation proceed with the more general rules in the annotation hierarchy, which use the morphological information annotated in the corpus.

Figure 9.19 gives an overview of the annotation hierarchy for the gram-

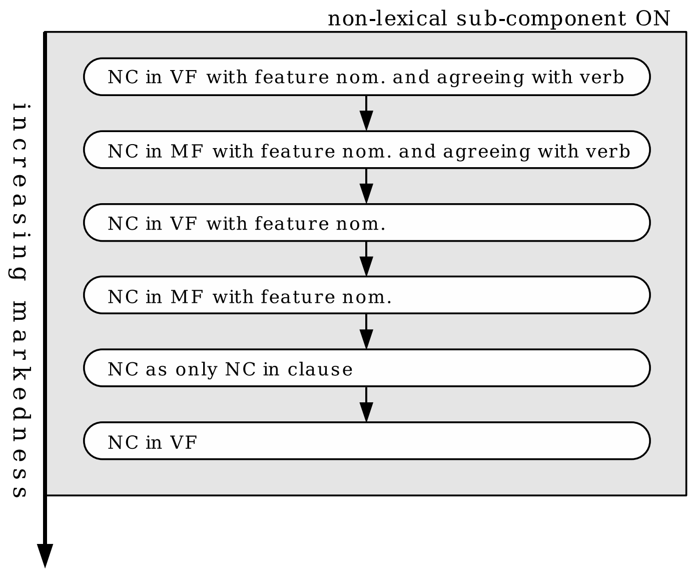

non-lexical sub-component ON
NC in VF with feature nom. and agreeing with verb
NC in MF with feature nom. and agreeing with verb
NC in VF with feature nom.
NC in MF with feature nom.
NC as only NC in clause
NC in VF
increasing markedness

Figure 9.19: The ranked sequence of rules for ON

matical function ON. With every rule, the constraints put on the annotation are relaxed. At first, the NC which is tested as the potential ON is checked for the feature nominative and agreement with the verb (first in VF and then in MF). Then, the NC is just checked for nominative. In the last but one step, an NC as the only NC in the sentence is annotated since we assume that every sentence has a subject. (Except for sentences like 'Mir ist schlecht', which we annotate before.) In the very last step, if still no ON is found in the sentence, an NC in the VF is annotated without checking any other features. This is done because the VF is the most typical position of the ON.

9.7 Modeling Ranking by Using OT Constraints

The annotation philosophy of preferring less marked GF orders to more marked GF orders can be compared to an OT approach. In an OT approach as it is presented in Frank, Holloway King, Kuhn, and Maxwell (1998), different candidate analyses are generated. For these analyses, a record is kept which includes the 'optimality marks'. These optimality marks are ranked on a scale. According to a fixed scheme, the respective analyses are then evaluated with respect to the optimality marks which have been assigned to them.

Ranking and OT The main difference between this approach and the one in our system is that, in the OT approach, more than one analysis is generated and these different analyses are then evaluated according to the kind of optimality marks they are assigned with. By contrast, in our system, every sentence just gets one analysis but the rules with which the sentence is analyzed are relaxed with respect to the constraints which they apply. A fit example to describe the difference between the OT approach as in Frank et al. (1998) and our approach is the problem of checking verb-subject agreement: Obviously, checking of verb-subject agreement helps enhancing the accuracy of annotation. There may, however, be cases in which subject and verb do not agree because of a grammatical mistakes of (in our case) the writer. Still, it may be useful to annotate such structures. In the OT approach such structures are marked with a 'negative' optimality mark. If, however, no other analysis exists, then even a structure with a negative optimality mark is accepted.

In our approach, we first try to annotate the structure checking verb-subject agreement. In this step, we also use the SC information and we apply the ranking of constituent orders. If the annotation fails, because no potential ON agrees with the verb in number, the following rules of the non-lexical component are applied. In this component, SC information can be no longer used and grammatical functions are not annotated together but one after the other. In this component, verb-subject agreement constraints are finally abandoned and the ON candidates are just checked for their case. The system first looks for a potential ON candidate in the VF (in V2 clauses) and, if it does not succeed, it tries to find a candidate in the MF. Since nearly all V2 clauses have an ON, a last solution also includes the possibility that the first noun chunk in the clause is the ON regardless of morphological features.

The constructed example 163 illustrates the procedure: The ON of the sentence is 'die Schwester der beiden' and it should agree with the verb 'sehen' (i.e. the verb should be 'sieht' not 'sehen'); the OA of the clause is 'eine Erscheinung'. Both noun chunks may be either the ON or the OA as regards their morphological ambiguity class, and both noun chunks are singular. The system first tries to annotate the sentence with the constituent order 'ON verb OA' including an agreement check of verb and potential ON preceding the verb. This is the most optimal solution. The second optimal solution is the constituent order 'OA verb ON' (i.e. 'die Schwester der beiden as accusative object and 'eine Erscheinung' as the nominative object). Since both solutions do not work because the verb does not agree in number with any of the potential ONs, grammatical functions are not annotated in this component.

(163) [$_{ON}$ Die Schwester der beiden] sehen [$_{OA}$ eine Erscheinung].
The sister of the two see an apparition.

'The sister of the two sees an apparition.'

Since the lexical component has not annotated sentence 163, it is handed over to the non-lexical component. In the non-lexical component, there are also ranked solutions for the annotation of the sentence. Since we do not keep a parsing record, the system treats all sentences alike whether they were not annotated before because the SC information was missing or for any other reason (e.g. missing subject-verb agreement). Thus, at first, verb-subject agreement is checked as well and, again, an ON in the VF is preferred to an ON in the MF. These rules do, again, not apply to our sentence since they still involve agreement of subject (ON) and verb. In the following step, the verb-subject agreement constraint is abandoned and the system tries to annotate chunks with nominative in its morphological ambiguity class as ON. Both 'die Schwester der beiden' and 'eine Erscheinung' would work, now; but, again, the first possible noun chunk in the clause is the more optimal solution and, in this case, this is the correct solution and 'die Schwester der beiden' is assigned the grammatical function ON. In a further step with a similar structure, the accusative object (OA) is recognized.

There is still a further step in the annotation process, which need not be applied to sentence 163, because after the annotation of the grammatical functions, the annotation process for this sentence stops. In this step, the constraints are relaxed even more. Here, the parser would annotate just any first noun chunk in a V2 clause as ON. This rule would only be applied if the correct noun chunk had no 'nominative' among its features. This could be the case for various reasons: The DMOR analysis could be inaccurate, the morphological disambiguation component could have failed, the chunking could be imperfect such that the morphological disambiguation could have failed, etc. Since we want our system to be robust, we try to annotate every sentence if possible. Thus, the relaxation of constraints is also an instrument to achieve robustness and to remedy errors which were introduced by preceding levels of annotation.

Sentence 164 gives a good realistic example of the advantages of an approach which also allows to a certain extent ungrammatical constructions. The sentence is from the TüBa-D/Z. It is a letter to the editor. While newspaper articles mostly contain grammatical constructions, letters to the editor or newspaper genres with less formal character (e.g. glosses) may also to a higher degree contain language with some ungrammatical material. But the reason for the ungrammatical construction in this sentence may also be another one: The singular ON 'diese Diskussion', which is supposed to agree in number with the verb is very far apart from the respective plural verb 'spielen' (which should be 'spielt') and also intervened by a lot of phrases which are actually plural (but not the ON). This grammatical construction in connection with an inexperience of the writer, which is also reflected in the semantic content of the sentence, seems to have led to this mistake. Since language like that does occur, it is advisable to have instruments to deal with it so that the parser can produce an outcome.

(164) In diesem Zusammenhang habe ich geantwortet, daß $[_{\mathrm{ON}}$ diese
In this context have I answered that this
Diskussion] um Ziele von NGOs wie soziale Gerechtigkeit,
discussion about aims of NGOs like social justice,
menschenwürdiges Leben, die ich als Menschenrechtsverletzungen
humane life, which I as human rights violations
zusammengefaßt habe, meines Erachtens $[_{\mathrm{OA}}$ keine Rolle] bei der
subsumed have, to my mind no role with the
aktuellen Spendenbereitschaft spielen.
current donation willingness play.

'In this context I have answered that this discussion about aims of NGOs like social justice [and] humane living conditions, which I have subsumed under human rights violations, does – to my mind – not play any role as regards the current willingness to donate [money].'

OT constraints in the ranked FSTs The constraints as they are applied in our system can be modeled using Optimality Theory (OT). This does not mean that the constraints as they are depicted in table 9.1 are directly encoded in the annotation system; but they are, in fact, inherent in the ranking of the cascaded FSTs. Considering OT constraints in table 9.1

helps understanding the our annotation strategy. It shows the advantages of the system of ranked FSTs but also reveals the parts in which additional information could be used. Some examples illustrate how the ranking of FSTs can be expressed by using OT constraints.

C_1	ON must be NC or CL
C_2	ON must have feature 'nominative'
C_3	ON must agree with verb in number
C_4	ON must have feature 'nominative' as only feature
C_5	ON must occur before all other potential grammatical functions
C_6	ON must occur in VF

Table 9.1: Constraints for the assignment of ON

Table 9.1 shows the OT constraints from the point of view of the assignment of ONs. OT constraint C_1 is the highest-ranked constraint and constraint C_6 the lowest-ranked. The principles which are used in the evaluation of constraints are the ones outlined in Kager (1999), sec. 1.4. The most important one in our case is the principle of *strict domination*, which states that "violation of higher-ranked constraints cannot be compensated for by satisfaction of lower-ranked constraints" (sec. 1.4). In our case, this means that, e.g., if a chunk has not got 'nominative' in its morphological ambiguity class (C_2), then it does not matter whether the lower-ranked constraints are violated or not, as long as there is a solution which does not violate C_2. The solution not violating C_2 would then be the one which would be selected. Sentence 165 gives an example which is further illustrated in table 9.2. All noun chunks are tested as candidates for the ON; this is C_1. The noun chunk 'ihn' is the one which already fails to satisfy C_2 because it has not got the feature 'nominative' in its morphological ambiguity class. This is fatal as indicated by the accompanying '!'. In this case, it does not matter that, in total, 'die Beamtin' satisfies less constraints then 'ihn'.

(165) [$_{\text{OA}}$ Ihn] sieht [$_{\text{ON}}$ die Beamtin] nicht.
Him sees the officer not.

'The officer does not see **him**.'

The mechanism which annotates the grammatical functions as reflected by the OT constraints in table 9.1 can best be illustrated by giving some simple constructed examples which range from the most optimal structure to the

ON candidates	C_1	C_2	C_3	C_4	C_5	C_6
ihn		*!		*		
die Beamtin				*	*	*

Table 9.2:

least optimal one. Sentence 166 shows one of the most optimal cases: If we assume that every material in topological fields, i.e. every chunk except verbal chunks and conjunctions[7] can be a potential grammatical function, then there are two potential ONs in the clause: 'der Lehrer' and 'den Schülern'. However, while 'der Lehrer' satisfies all constraints for an ON, 'den Schülern' just satisfies the constraint that it is a noun chunk. Cases like these are the easiest to decide. They are also the most unmarked in terms of OT (cf. Kager, 1999, sec. 1.1.2). This becomes very clear in table 9.3, which shows that the ON 'der Lehrer' does not violate any of the constraints.

(166) [$_{ON}$ Der Lehrer] antwortet [$_{OD}$ den Schülern].
The teacher answers the pupils.

'The teacher gives the pupils an answer.'

ON candidates	C_1	C_2	C_3	C_4	C_5	C_6
der Lehrer						
den Schülern		*!	*	*	*	*

Table 9.3:

Sentence 167 is different from sentence 166 with respect to markedness. Again, C_2 solves the ambiguity, but unlike in sentence 166 the ON violates C_3, C_4 and C_5. Sentence 167 is, thus, more marked than sentence 166. Kager (1999), sec. 1.1.2, attributes such markedness to the expression of contrast and this is, in fact, the case in sentence 167. The noun chunk 'dem Schüler' is the one in focus. What is important for our system, if one compares the two sentences 166 and 167 as regards the violations of constraints as the are reflected in table 9.3 and table 9.4, is that the violation of C_2 alone decides about the annotation of the ON. This shows the importance of morphology

[7]Relative pronouns would not count among conjunctions, in this case.

for the annotation of grammatical functions. In order for a potential noun chunk to be annotated as ON, it must have the feature ‘nominative’ in its ambiguity class. Yet, what is revealed by the two tables as well is that the potential noun chunk need not be fully disambiguated (C_4). Sometimes, it suffices that all of the other potential noun chunks are unambiguously not ON.

(167)	[$_{\text{OD}}$	Dem	Schüler]	antwortet	[$_{\text{ON}}$	die	Lehrerin].
		The	pupil	answers		the	teacher.

‘It is the pupil who gives an answer to the teacher.’

ON candidates	C_1	C_2	C_3	C_4	C_5	C_6
die Lehrerin			*	*	*	
dem Schüler		*!		*		

Table 9.4:

Sentence 168 shows what happens if it is impossible to decide on the grounds of morphology which of the potential chunks is the ON. This is the case if none of the potential ONs is either unambiguously ON or unambiguously not ON. In sentence 168 the chunks ‘die Lehrerin’ and ‘die Schülerin’ may each be either the ON or the OA. In this case, an ON in the VF would be the most unmarked solution, i.e. the one with the fewest violated OT constraints (just C_4). The next one is the one with the more general constraint that the ON be the first of any potential ONs. This is the one in sentence 168. Sentence 168 in conjunction with table 9.5 show that in German the morphology is the decisive feature for the detection of grammatical functions. This is underlined by sentence 169, which shows that the ON can even come last if only the morphology of the ON is unambiguous. Thus, only if the morphology (including verb-subject agreement) is ambiguous, the constituent order is the decisive feature for disambiguation.

(168)	Gestern	hat	[$_{\text{ON}}$	die Lehrerin]	[$_{\text{OA}}$	die Schülerin]	unterrichtet.
	Yesterday	has		the teacher		the pupil	taught.

‘Yesterday, the teacher taught the pupil.’

ON candidates	C_1	C_2	C_3	C_4	C_5	C_6
die Lehrerin				*		*
die Schülerin				*	*!	*

Table 9.5:

(169) Gestern hat [$_{OA}$ die Lehrerin] [$_{ON}$ keiner] gesehen.
Yesterday has the teacher no one seen.
'Yesterday, no one saw the teacher.'

ON candidates	C_1	C_2	C_3	C_4	C_5	C_6
die Lehrerin				*!	*	
keiner					*	*

Table 9.6:

Although morphology is the decisive feature for grammatical function detection in German, the number of cases which are disambiguated by constituent order is relatively high because the most frequent grammatical functions ON and OA are also the ones which are very often mutually ambiguous. This ambiguity is not resolvable without either knowledge about the full syntactic structure of the clause, which is a vicious circle because the annotation of this structure is precisely final goal of the morphological disambiguation, or semantic and/or world knowledge which is not accessible in our system (see below). Thus, in those cases the system has to resort to linearization information.

If morphological features do not disambiguate grammatical functions, linearization information alone can, in some cases, fail to identify the correct structure as sentence 170 and table 9.7 show. Although this sentence might also be regarded as genuinely ambiguous since it can either mean that the sister has seen the town or figuratively that the town has seen the sister, it is very likely that, in case at hand, the reading as annotated in sentence 170 is the correct one. Thus, obviously, our system has a lack of information since in it C_5 incorrectly rules out 'meine Schwester' as the ON, as is shown in table 9.7. With the help of the OT constraints table, it can also be made out at which point of the constraint hierarchy the information is missing:

between C_4 and C_5. If the sentence were ‘Die Altstadt hat meine Schwestern schon gesehen’ (*The old town has seen my sisters*), ‘meine Schwestern’ would violate C_3 and, thus, ‘die Altstadt’ would unambiguously be ON; and this would, in fact, be the only reading of the sentence.

(170) [$_{\text{OA}}$ Die Altstadt] hat [$_{\text{ON}}$ meine Schwester] gestern schon
The old town has my sister yesterday already
gesehen.
seen.
‘My sister has already seen the old town, yesterday.’

ON candidates	C_1	C_2	C_3	C_4	C_5	C_6
die Altstadt				*		
meine Schwester				*	*!	*

Table 9.7:

The kind of information which is missing in the OT constraint hierarchy is semantic. The reason why humans do not misinterpret sentences in which ON and OA are not distinguishable by morphological features and in which the constituent order is such that the OA occupies a typical ON position and vice versa is that there are restrictions with respect to the semantic features of the grammatical functions sub-categorized by a verb. In the case of the verb ‘sehen’ this would be that the agent of the verb (i.e. the head of the ON in active clauses and the nominal head of the OPP in passive clauses) has the feature ‘+animate’. This information is situated after the morphological information and before the linearization information in the OT constraint hierarchy.

In order to avoid erroneous annotations like the one in sentence 170, we would have to add information about the semantic selectional restrictions to the verb, and information about the respective semantic features to each potential grammatical function as it is also proposed in Frank et al. (1998). One source of such information could be the GermaNet (Kunze, 2001; Kunze and Wagner, 2002), which is a lexical-semantic wordnet. The main content of GermaNet are the semantic relations between lexical items. For the problem at hand, hyponymy could be used to identify the major semantic class of a potential grammatical function. The entries for the verbs do not contain information about the selectional restrictions, but there is semantic verb classification like ‘feeling, communication or cognition’. These classes could be used to connect the verbs with the correct grammatical functions.

There are, however, some problems with this approach: One technical limitation is that a lot of words can come up which are not contained in GermaNet. Another technical problem is anaphora resolution for pronouns for which the referred word has to be found in order to assign the correct semantic features. A genuine problem of the approach is that if just one of the elements in a verb-grammatical function set is used figuratively, the approach runs into problems. These phenomena do not occur rarely and are not as unusual as the term ‘figurative’ suggests as sentence 171 and 172 show, in which the verb ‘sehen’ is used with a [-animate] subject. Furthermore, structures like the one in sentence 170, which cause mistakes, are not so frequent as to justify boosting the system with a component which would surely considerably increase the size of the system and also substantially slow it down. To include full semantic information would, thus, be uneconomical. An alternative to be tested could be to include partial semantic information into the system. This is also done in our system when we identify temporal expression to exclude them from the annotation of grammatical functions.

(171) Als vordringlichstes Problem sieht das Verkehrsbündnis den
As most urgent problem sees the traffic league the
Nord-Süd-Verkehr durch den Raum Hamburg [...]
north south traffic through the area Hamburg

‘The traffic league regards the north-south traffic through the Hamburg area as the most urgent problem.’

(172) Der Kreuzberger Antrag sieht hingegen keine
The Kreuzberg motion sees on the contrary no
ernstzunehmende Rechtfertigung für den Verstoß [gegen die
serious justification for the violation [of the
UN-Charta]
UN charta]

‘On the contrary, the motion from Kreuzberg cannot find any serious justification for the violation of the UN charta’

Finding OT constraints in ranking As has already been mentioned, OT constraints as they are listed in table 9.8[8] are not directly encoded in the annotation system. OT constraints are just one way of modeling the ranking approach in a generally accepted theory. In this section, the actual procedure of disambiguation is described. We choose the rules for a verb-second affirmative clause for the SC frame 'ON OA' as an example (cf. figure 9.20) because this type of sentence is one of the most frequent and most typical examples of a sentence (of the kind 'The boy kisses the girl' or 'John loves Mary').

The ranked rules are called successively. This process begins with the most optimal solution (i.e. the one with the fewest violations of constraints) and is applied either until the least optimal solution or until one solution matches. At the point of assignment of grammatical functions, the chunks in the corpus already contain information about their category, topological field and case. This information is used in the annotation process. The first rule in the ranked hierarchy is, consequently, the one which tries to assign the element in the VF with ON since the weakest constraint C_6 demands the ON to be in the VF. All the other constraints have to be fulfilled as well because they are the stronger constraints, or either they have to be violated by all other potential ONs. This will be exemplified below with C_4.

C_1	ON must be NC or CL
C_2	ON must have feature 'nominative'
C_3	ON must agree with verb in number
C_4	ON must have feature 'nominative' as only feature
C_5	ON must occur before all other potential grammatical functions
C_6	ON must occur in VF

Table 9.8: Constraints for the assignment of ON

The first choice is exemplified by the rules marked 'A' (cf. figure 9.20). There are two of them because the lexical verb which triggers off the annotation may be either in the left part of the sentence bracket or in the right one. These two solutions are disjunct. For this reason, they do not have to follow each other directly and are separated in two blocks to enhance maintainabilty. The second choice is 'B'. In this case, ON still precedes OA but

[8]Table 9.8 is identical to table 9.1 and just replicated here for convenience.

the VF is occupied by optional elements like in rules 2, 6, 9 and 10. An example for the lexical verb in the VF is sentence 173 covered by rule 9. In the second choice, C_6 is violated but all other constraints are fulfilled. C_6 and C_5 are violated in 'C' and 'D'. In these rules, the OA comes before the ON, either in the VF ('D') or in them MF ('C'). Constraints C_2 and C_3 are never violated in the component of the system using SC information. C_1 is never violated in all the system.

```
A  1. ON_VF_W VCL_akt_fin_AM_T OA_MF_W VCR_akt_nfin_ONOA_T
B  2. FREE_VF_T VCL_akt_fin_AM_T ON_MF_W OA_MF_W VCR_akt_nfin_ONOA_T
C  3. FREE_VF_T VCL_akt_fin_AM_T OA_MF_W ON_MF_W VCR_akt_nfin_ONOA_T
D  4. OA_VF_W VCL_akt_fin_AM_T ON_MF_W VCR_akt_nfin_ONOA_T

A  5. ON_VF_W VCL_akt_fin_ONOA_T OA_MF_W
B  6. FREE_VF_T VCL_akt_fin_ONOA_T ON_VF_W OA_MF_W
C  7. FREE_VF_T VCL_akt_fin_ONOA_T OA_MF_W ON_VF_W
D  8. OA_VF_W VCL_akt_fin_ONOA_T ON_MF_W

B  9. VCF_akt_nfin_ONOA_T VCL_akt_fin_AM_T ON_MF_W OA_MF_W
C 10. VCF_akt_nfin_ONOA_T VCL_akt_fin_AM_T OA_MF_W ON_MF_W
```

Figure 9.20: Plain text rules for a V2 affirmative clause for the SC frame 'ON OA'

(173) Gesehen habe ich den Postboten heute noch nicht.
Seen have I the postman today yet not.
'I have not yet seen the postman, today. [But the mail is already there.]'

C_4 is the one constraint which makes it very clear how much the OT constraints are only indirectly encoded in the system. The abbreviations for the grammatical functions represent a feature bundle of chunk category, distribution and case feature. For instance, the abbreviation 'ON_VF_W' represents a noun chunk in the Vorfeld with the feature 'nominative' among its morphological features. It is not checked whether the feature 'nominative' is the only feature in the set of morphological features. This would yield at least twice the rules: once with the feature 'nominative' among other features and once with the feature 'nominative' as the only feature (and a combination of the same dichatomie with the other grammatical functions). This is, however, not necessary since in the case that the ON occurs before any other grammatical function (as the most optimal solution), a wrong annotation is excluded by the distribution, and, in the case that the ON occurs after the other grammatical functions, a wrong annotation is excluded by the fact that we do not allow chunks with the feature 'nominative' to occur among the optional chunks and by the plain fact that the correct ON simply cannot be annotated as anything else since it has just got the feature 'nominative' for case.

This can be made clear by looking at the rules in figure 9.20 and at an example. If, for instance, the OA is in the VF and the ON in the MF and the OA has the features 'nominative' and 'accusative' but the ON is unambiguously 'nominative' like in sentence 174, then the system cannot erroneously assign the ON an OA label. This has the effect that the OA, which is, in fact, ambiguous, cannot erroneously be assigned ON since none of the other potential grammatical functions qualifies for an OA. Thus, C_4 is checked indirectly. Although the system does not not check whether 'nominative' is the only feature in a potential ON, the coaction of the annotation of all grammatical functions and the ranking of the rules guarantees the consideration of C_4.

This can be exemplified by sentence 174 in connection with figure 9.20: Rules 1–4 do not apply at all since the lexical verb is in the left part of the sentence bracket. In rule 5, the system matches 'seine Schwester' as ON since it has both 'accusative' and 'nominative' as case features. It fails, however, to assign 'der Mann' OA since this chunk has no case feature 'accusative'. Thus, rule 5 is blocked because the correct ON 'der Mann' has 'nominative' as its only feature. Since rules 6 and 7 do not match the structure either, rule 8 annotates it correctly as OA in the VF and ON in the MF (cf. sentence 174). Sentence 175 illustrates the case that both potential ONs violate C_4. In this case, the violation of either C_5 or C_6 disambiguates the structure. In sentence 175 both C_4 and C_6 are violated and, thus, it is C_5 which finally decides about the correct structure. This example illustrates the fact that if a stronger constraint is violated by all potential structures (like C_4 in the case at hand), then the violation of the next weakest constraint is the crucial one (cf. table 9.9).

(174)	[$_{OA}$ Seine	Schwester]	erkennt	[$_{ON}$ der Mann]	nicht	mehr.
	His	sister	recognizes	the man	no	more.

‘The man does not recognize his sister any more.’

(175)	Meistens	beachten	[$_{ON}$ die Leute]	[$_{OA}$ die Kinder]	nicht.
	Most of the times	notice	the people	the children	not.

‘Most of the times, the people are not taking notice of the children.’

ON candidates	C_1	C_2	C_3	C_4	C_5	C_6
die Leute				*		*
die Kinder				*	*!	*

Table 9.9:

Constraints C_1–C_3 are never violated in the component using SC information. These are tested in a further component, the non-lexical component. In this component, C_4 is no longer taken into consideration. At first all, e.g., ONs are annotated using C_1–C_3, C_5 and C_6. The dropping of C_5 and C_6 is organized just like in the lexical component. Then, in a next step, C_3 is abandoned and verb-subject agreement is no longer checked. In the last step C_2 is violated and in every sentence with a noun chunk an ON is annotated. The last two steps annotate just a few structures since most of the times a verb agrees with the ON, and it occurs very rarely that the ON does not have the feature ‘nominative’ in its ambiguity class since all tokens not covered by DMOR automatically receive ‘nominative’ in its ambiguity class. The only candidates for the last step are tokens which were falsely annotated as not ‘nominative’, e.g. names which were historically genitives and which are also common nouns, e.g. ‘Kellers’.

Part III

Evaluation and Comparison

Chapter 10

Shallow Annotation

ABSTRACT: Chapter 10 describes the evaluation of the shallow parser which builds the input for the parser for grammatical functions. Section 10.1 depicts the system of evaluation by giving a description of the annotation task and introducing the gold standard corpus with which the annotated structures are compared. Furthermore, it is shown how the annotated data and the gold data are converted into a common representation for evaluation. At last, the methods of evaluation are presented. Section 10.2 then presents the actual test setting, i.e. the test section of the gold corpus and the way we deal with some evaluation problems. Section 10.3 presents and discusses the results of the evaluation for each of the three groups of shallow phenomena, i.e. chunks, topological fields and clauses.

10.1 The Evaluation System

10.1.1 Task Description

There are two main motives for the evaluation of a parser: First, the evaluation of a parser points out its weaknesses and the results of the evaluation can be used to improve the performance of the parser. The evaluation component can, thus, serve as a diagnosis tool to improve the parser during the development phase. Second, the evaluation indicates the performance of the parser in relation to other parsers. It can, thus, be used for the comparison with other parsers. Furthermore, users can draw conclusions about the quality of the parser from the evaluation results. The evaluation component can, consequently, also serve as an assessment tool in order to give information about the quality of the output of the parser which can be expected.

In order to automatically evaluate the output of a parser, one typically needs correctly annotated data (*gold data*) against which the output can be checked and a method to compare the output of the parser with the gold data. The portion of gold data used for development should not be used for evaluation as otherwise the parser would be tuned to the test set. However, the gold data should also reflect the target text type(s) and the domain(s) the parser is intended for. We have chosen the *taz (die tageszeitung)* newspaper corpus as the target of our annotation since it contains a variety of text types and is, thus, a relatively varied section of the German language. The TüBa-D/Z provides a syntactically annotated sub-section of the taz newspaper corpus. This can be split into two sections and used as a gold corpus both for the development and the evaluation of the parser.

The method of evaluation should be both meaningful for the user and adequate for the task. Thus, an evaluation method has to be found which points out the strong points and the weaknesses of the parser. For the users, it is important to know the overall error rate of the parser and the error rate with respect to the individual annotated categories. For the development of the parser, a detailed error analysis is important, which shows the most frequent error types and which also shows in detail the effects of different annotation strategies. Otherwise, mending errors at one point may lead to inserting errors at others. Taking into consideration the results of the evaluation system will prevent this disadvantageous mechanism. We have decided to split the tasks of the evaluation of constituent annotation and GF annotation because they are annotated by different components of the parser.

In conclusion, an evaluation system needs three main components: An annotated corpus which serves as a gold standard, a method to map the gold standard data to the output of the parser, and evaluation methods which yield a quantitative error analysis and allow for a detailed qualitative error analysis:

- gold standard corpus
- conversion into common representation
- methods for error analysis

10.1.2 The TüBa-D/Z Gold Corpus

We use the TüBa-D/Z as a gold standard corpus for the evaluation of GFs. The TüBa-D/Z is a syntactically annotated corpus.[1] The development of the TüBa-D/Z has been financed by the *Kompetenzzentrum für Text- und Informationstechnologie (KIT)* and has received additional support by the project *A 1 – Representation and Automatic Acquisition of Linguistic Data* of the *Special Research Program (Sonderforschungsbereich) 441 – Linguistic Data Structures.* At the point of evaluation, the TüBa-D/Z contained 15,000 sentences or 260,000 tokens. The annotation was achieved manually. The text of the TüBa-D/Z is taken from the *taz* newspaper corpus and, at the time of evaluation, ranged from May 3rd to May 7th 1999. The corpus contains various text types like press news reports, press editorials, reportages, letters to the editor or squibs. The domains of the texts are as varied as politics, culture, sports, pop-music or every-day life. The texts in the TüBa-D/Z are, thus, a lot more varied than the texts in, e.g., the *Wall Street Journal* part of the *Penn Treebank* (Marcus, Santorini, and Marcinkiewicz, 1993) which is restricted to the domain of financial news and which is often used for parser evaluation.

The annotation scheme of the TüBa-D/Z (Telljohann, Hinrichs, and Kübler, 2003) is based upon an annotation scheme used in the *Verbmobil* project (Stegmann, Telljohann, and Hinrichs, 2000), a long-term machine translation project funded by the German *Federal Ministry for Education, Science, Research and Technology (BMBF).* There are four levels of syntactic constituents in the annotation scheme: lexical items, phrases, topological fields and clauses. Special attention has been given to keep the annotation scheme as compatible as possible with major syntactic theories. The annotation scheme is surface-oriented in the sense that the primary ordering principle of the clause is the topological fields structure and in the sense that crossing edges and traces do not occur. Besides constituents, GFs are marked by edge labels so that they are essentially attributes of the constituents from which the arcs originate. The GFs could not have been annotated as edges themselves because this would have resulted in crossing edges since GFs can occur across topological field boundaries. Furthermore, in the TüBa-D/Z, the distinction between heads and non-heads is marked within phrases.

Figure 10.1 gives an example of a tree in the TüBa-D/Z. The tree consists of a V2 clause into which a VL clause is embedded. Both clauses are labeled

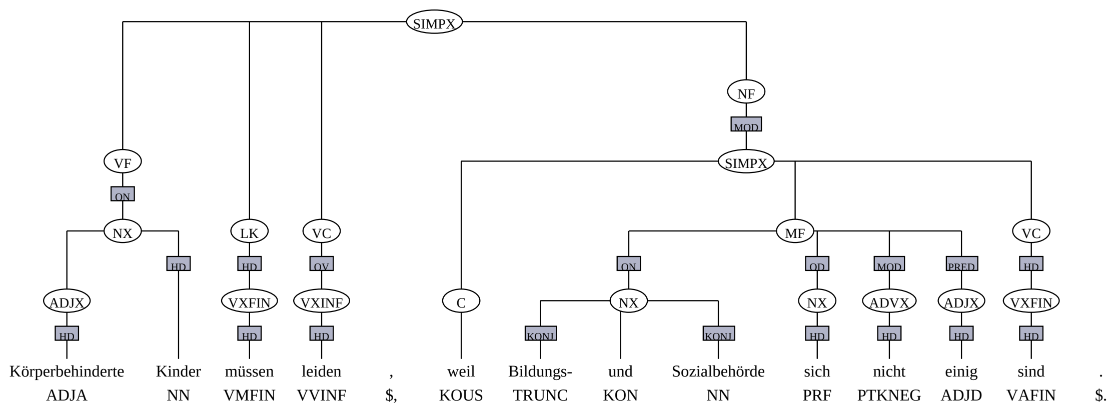
[1]The inventory of the TüBa-D/Z is listed in appendix C.

‘SIMPX’. The embedding SIMPX is divided into topological fields by the sentence bracket consisting of the left part of the sentence bracket (LK) and the right part (VC). There is an initial field (*Vorfeld*; VF) in front of the LK, no middle field (*Mittelfeld*; MF) in between the LK and the VC (as ‘empty MFs’ are not annotated in the TüBa-D/Z) and a final field (*Nachfeld*; NF) after the VC. Punctuation is not attached.

This topological field description is surface-oriented in that it does not show any of the grammatical relations in the sentence. These relations are shown by the edge labels. The SIMPX in the NF is, for instance, marked as ‘MOD’ which means that it is either an ambiguous modifier or a modifier of the whole sentence. In the case that, e.g., the SIMPX was a relative clause, the modified element would, in addition, be marked by the element which it modifies, e.g. OAMOD for modifier of accusative object. The scope of the MODs is confined by the structures in which they occur. Thus, in the embedded SIMPX, the ADVX (adverbial chunk or phrase) ‘nicht’, which serves as a MOD, can only be a modifier of either the whole embedded SIMPX or any elements within it. In the same way, the relations of the two subjects (ON) which occur in the sentence are confined. The ON occurring in the VF of the embedding SIMPX can only be the subject referring to the finite verb in the LK and the ON in the embedded SIMPX can only refer to the finite verb in the VC of this very SIMPX.

10.1.3 Conversion into common representation

From the linguistic phenomena annotated in the TüBa-D/Z, only the constituent structures are relevant for the evaluation of the shallow annotation, i.e. phrases, topological fields and clauses. Figure 10.2 shows the sentence from the TüBa-D/Z in figure 10.1 as it is annotated by the shallow parser. For the evaluation of the shallow parser, we now have to map the relevant structures in figure 10.1 to the ones in figure 10.2. In order to do so, we use a representation type which is based upon the one proposed in Ramshaw and Marcus (1995). In it, constituent structure is represented by tagging each token with its constituent type. Since this might lead to ambiguous structures in cases in which two constituents of the same type immediately follow each other, the beginning of the constituent is particularly marked, in those cases. Tokens outside any constituent are marked as ‘outside’. According to the criteria for their assignement, i.e. **i**nside, **o**utside or **b**eginning of a constituent, these tags are often referred to as IOB-tags.



Körperbehinderte	Kinder	müssen	leiden,	weil	die	Bildungs-
Physically disabled	children	must	suffer,	because	the	education

und	Sozialbehörde	sich	nicht	einig	sind.
and	social agency	themselves	not	at one	are.

‘Physically disabled children have to suffer because the education agency and the social agency are incapable of coming to an understanding.’

Figure 10.1: Syntactically annotated tree from the TüBa-D/Z

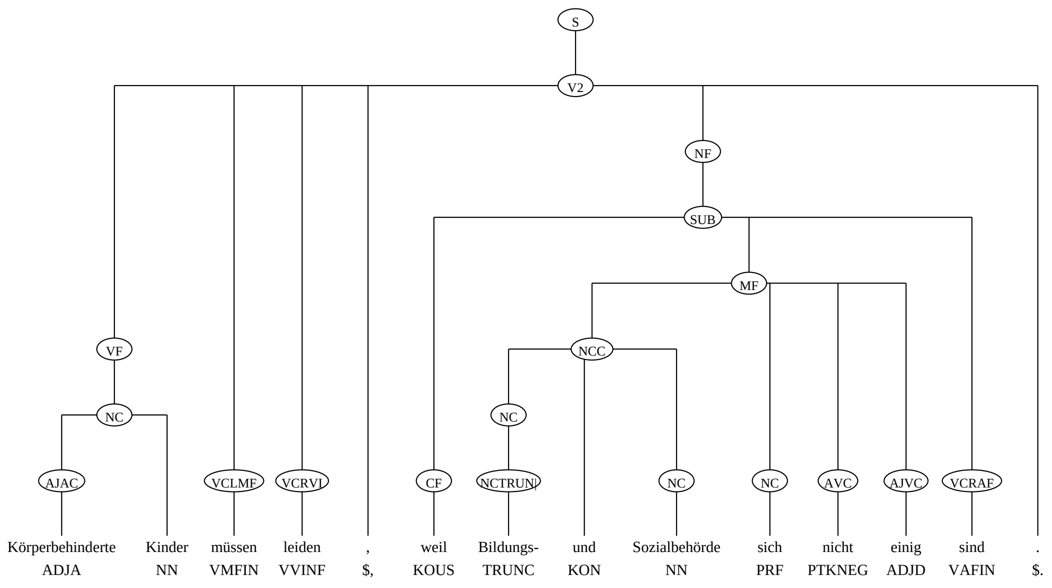


Figure 10.2: Syntactically annotated tree from the TüBa-D/Z

We mark each token according to the conventions used for the shared task of the CoNLL-2000, which was chunking (cf. Tjong Kim Sang and Buchholz, 2000), i.e. the first word of **each** constituent X receives the label B-X, all non-initial tokens receive the tag I-X. Tokens outside any constituent receive the label O. It is straightforwardly possible to convert the constituents in the tree structure of the TüBa-D/Z and the output of the shallow parser into the IOB-tag representation. The different shallow phenomena chunks, topological fields and clauses can be treated in different layers of annotation, i.e. a different IOB-tag representation is yielded for each of those three phenomena. In the case of clauses, two layers are generated, one for subordinate clauses and one for matrix clauses. Figure 10.3 shows the representation of chunks with IOB-tags as an example. Only a full (i.e. exact) match is counted as correct, everything else is counted as an error. Since maximal chunks, i.e. chunks which are not contained in any other chunk, are the targets of GF annotation, their annotation has been evaluated. Since maximal chunks are, by definition, non-recursive events, they can be evaluated by one layer of IOB-tags.

```
Körperbehinderte B-NC B-NC
Kinder I-NC I-NC
müssen B-LK B-LK
leiden B-VC B-VC
, O O
weil B-C B-C
Bildungs- B-NC B-NC
und I-NC I-NC
Sozialbehörde I-NC I-NC
sich B-NC B-NC
nicht O O
einig B-AC B-AC
sind B-VC B-VC
. O O
```

Figure 10.3: Chunks as represented with IOB-tags

10.1.4 Methods of Evaluation

In order to assess the performance of a parser, it is necessary to know in how many cases the parser has assigned the correct linguistic annotation and in how many cases it has failed. In order to gain more precise conclusions, it is necessary to give the figures for each linguistic phenomenon (in our case, for each chunk which serves as the target for GF annotation). A problem is that there is usually a trade-off between annotating as many of the occurring linguistic phenomena as possible and annotating them as accurate as possible. This dilemma can best be depicted by the concepts of *precision* and *recall*, which should be included in an evaluation system.

Counting the errors is achieved using the data in the representation as shown in figure 10.3. From the data, we calculate three measures: precision, recall and *F-Measure*. Precision is defined as the percentage of detected tags which are correct, i.e. the ratio of true positives to all annotated positives. Recall is defined as the percentage of gold standard tags which are correctly detected, i.e. the ratio of true positives to all gold positives. The two measures are illustrated by figure 10.4. Using precision and recall for the evaluation of NLP systems has become very common (cf. Manning and Schütze, 1999, sec. 8.1) and there are good reasons for it: The measures

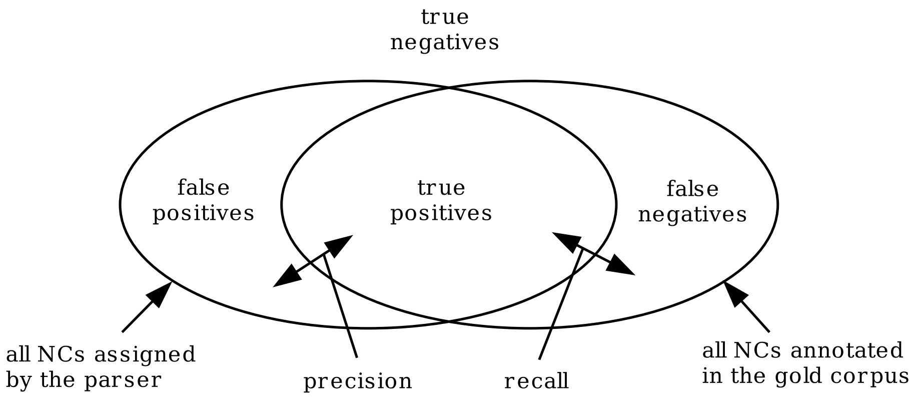


Figure 10.4: Precision and recall (for NCs)

of precision and recall illustrate very vividly the strengths and weaknesses of a parser. If, in our case, we just focused on the unambiguous cases of chunks, e.g. just the cases of a preposition directly followed by a noun for PCs, then the precision of the parser would be high because there would be few false positives. However, the recall would be low because the number of false negatives would be high.

For our parser, we try to keep precision and recall as balanced as possible. The reason for this approach is the goal of our annotation. We are annotating a corpus for linguistic research. For the linguist querying a corpus it would be demotivating if a large number of the results were incorrect (low precision). On the other hand, the results would be spoiled if they did not contain all intended structures (low recall). Typically, if recall is kept low, it is especially true that the structures not matched by the parser are not evenly distributed among the linguistic phenomena. Furthermore, they are different in nature from those matched by the parser and not just a random subset. If we just matched PCs with a preposition directly followed by a noun, then those PCs would not represent the full spectrum of PCs. Thus, usually, it is regarded as optimal if precision and recall are balanced unless favouring one of the two is of advantage for some sort of application.

The goal of the evaluation should, thus, be to keep precision as well as recall high and as balanced as possible to avoid a trade-off. This is easier to manage with just one measure for precision **and** recall. The F-measure is an ideal measure for this task (cf. Manning and Schütze, 1999, sec. 8.1). It is

$$F_{\beta=1} = \frac{(\beta^2+1)PR}{\beta^2 P+R} = \frac{2PR}{P+R}$$

Figure 10.5: F-Measure with $F_{\beta=1}$ (P=precision; R=recall)

	precision	recall	$F_{\beta=1}$
overall	96.08%	96.10%	96.09
AC	94.87%	95.42%	95.14
C	99.73%	99.18%	99.46
LK	99.90%	99.14%	99.52
NC	94.10%	94.39%	94.24
PC	95.58%	94.86%	95.22
VC	98.29%	99.35%	98.82

Table 10.1: Quantitative evaluation results for chunks

calculated as shown in figure 10.5. If $F_{\beta=1}$, precision and recall are equally weighted. Since this is what we intended, we can simplify the F-Measure as is shown in figure 10.5. The F-Measure is preferable to the arithmetic mean of precision and recall because it tends towards the lower of the two and, thus, punishes a bigger difference between precision and recall.

From the data as it is represented in figure 10.3, the three measures can now be calculated for overall performance and for the performance of every single category as shown in table 10.1, which shows the results of the shallow annotation for chunks (which will be discussed in section 10.3.1). This procedure is a useful diagnosis tool for the detection of weaknesses in the annotation algorithm. The grammar constructor is able to make inferences from these results and work on the crucial problems in the algorithm.

10.2 The Test-Setting for Evaluation

The TüBa-D/Z serves as the gold standard data against which the annotation of shallow structures is evaluated. The TüBa-D/Z provides a syntactically annotated sub-section of the taz newspaper corpus comprising (in the first release) 15,260 sentences (266,441 tokens) and ranging from May 3rd to May

7th 1999. We have split the corpus into a development set and a test set. The development set consists of the days of May 3rd, 4th, 5th and 7th, and the test set consists of May 6th. May 6th comprises 4,174 sentences (73,926 tokens) which is about 27% of the sentences in the whole TüBa-D/Z. The test set has not been used for the development of the parser but solely for its evaluation.

For the evaluation of chunks, we use a subsection of the test set since chunks are not coded in the TüBa-D/Z and the conversion of the phrases in the TüBa-D/Z into chunks as annotated by the shallow parser would have meant writing a grammar nearly as extensive as the chunk grammar itself. For the evaluation of chunks, the first 1,000 sentences of the test set were manually annotated with chunks, and this manually annotated test set was mapped to the chunk output of the shallow parser[2] using IOB-tags as described above. The test set for chunks contains 18,093 tokens and 7,185 chunks. The chunks which were evaluated subsume the chunks which make up the left part of the sentence bracket, i.e. LK (Linke Klammer) and C (C-Feld), and the right part of the sentence bracket (Verbkomplex; VC). Furthermore, the phrasal targets for the annotation of GFs, noun chunks (NCs), prepositional chunks (PCs) and adjectival chunks (ACs) are evaluated.

Only maximal chunks are evaluated. This has to be kept in mind when evaluation results are interpreted and/or compared to the results of other approaches: If, e.g., two ACs are coordinated (cf. figure 176) and the parser fails to recognize this coordination, we count 0% precision and 0% recall. If, however, one considers all the recognized chunks, then a parser would score 66.66% precision and 66.66% recall provided that it fails to recognize the coordinated chunk but succeeds to recognize the basic chunks. This distinction can yield considerable differences in evaluation results if a text contains some complex chunks.

(176)	[$_{AC}$ [$_{AC}$	recht	klein]	aber	[$_{AC}$	sehr	schön]].
		rather	small	but		very	beautiful

'rather small but very beautiful'

The first 1,000 sentences of the test set are a very heterogeneous section of the test section and contain text types as varied as newspaper reports, cinema reviews, letters to the editor or public announcements (about, e.g.,

[2]N.B.: The evaluation of the chunks includes the evaluation of all chunks annotated by the parser including complex chunks as defined in chapter 6.

elections or traffic obstructions). While newspaper reports provide problems like very complex chunks, the cinema reviews contain a lot of foreign languages material and creative writing; the latter is also true of letters to the editor which are not written by professional writers and sometimes in a colloquial style. Furthermore, the public announcements contain addresses as an unexpected type of linguistic phenomenon.

Topological fields and clauses are evaluated on the full test section. The elements of the sentence bracket are non-recursive and can thus be evaluated without any problems. The topological fields LV, VF, MF and NF, however, may be recursive events since, e.g., a clause may be embedded into an MF. We do, however, only evaluate maximal topological fields because it can be assumed, if the maximal topological field is annotated correctly, that the embedded field is annotated correctly, too, as otherwise the annotation of the embedding topological field would not have succeeded.

Since subordinate clauses are, by definition, embedded into matrix clauses, the clause structure is recursive. In order to deal with this recursiveness, matrix clauses and subordinate clauses are evaluated separately. For the evaluation, we only distinguish between verb-first (V1) and verb-second (V2) clauses, which are mainly matrix clauses, on the one hand, and verb-last (VL) clauses, which are almost exclusively subordinate clauses, on the other hand. Since the relation between two V1/V2 clauses, which may be one of matrix clause $\leftrightarrow$ subordinate clause is not subject of shallow annotation but subject of the GF annotation, it is not evaluated. Thus, two V1/V2 clauses are always annotated and evaluated as can be seen in sentence 177. The same is true of subsequent VL clauses, in which the relation is not annotated in the shallow structure and, thus, not evaluated (cf. sentence 178).

(177) [$_{\text{V1/V2}}$ Er sagte,] [$_{\text{V1/V2}}$ er wolle kommen].
He said, he want come.

'He said (that) he wanted to come.'

(178) Er möchte, [$_{\text{VL}}$ daß Du Dich meldest], [$_{\text{VL}}$ wenn Du wieder da bist].
He would like, that you yourself report, when you again there are.

'He would like you to call on him when you are back again.'

The shallow parser is evaluated both on gold PoS tags, i.e. PoS tags which were manually assigned independently by two different annotators, with differences resolved by a third, and on automatically annotated PoS tags. For the automatically assigned PoS tags, we used the TnT PoS tagger presented in Brants (2000). The tagger was trained on material of various text types from the Bild and taz newspapers (including the TüBa-D/Z), and from letters and novels. On the whole, the training and testing material consisted of about 629,000 tokens. The training of TnT on this material achieves an accuracy of 96.36% on the whole TüBa-D/Z with roughly 10% unknown tokens. For the evaluation, ten-fold cross validation was used. In our case, this means that the TüBa-D/Z was split into ten sections of about 26,600 tokens and each of these sections was evaluated separately and the results averaged. Training and test set were disjoint, i.e. the training material for each of the ten sections consisted of the 629,000 tokens except the respective test section of about 26,600 tokens (i.e. 602,400 tokens).

The accuracy of 96.36% compares well with the results obtained in Brants (2000) who uses the same method of evaluation and reports an accuracy of 96.7% for the TnT tagger trained on 320,000 tokens of the NEGRA corpus with about 12% unknown tokens. The accuracy of the TnT tagger on the test set for our shallow parser (i.e. May 6th 1999 of the taz) is 95.72%, which can be seen as an indication that this section of the corpus contains slighty different structures than the rest of the TüBa-D/Z corpus.

10.3 Results

The following sections present the results of the shallow parser. They will be discussed according to the linguistic phenomena of chunks, topological fields and clauses. Examples of typical errors will be given.

10.3.1 Chunks

Table 10.2 and figure 10.6 show the results of the evaluation of the chunk annotation on gold PoS tags. The results are satisfactory since overall precision and recall are both over 96%. The best results are for LK and VC. This is not astonishing since these chunks are the least complex chunks. However, their annotation is not trivial since the obvious approach to just chunk all verbal elements together (including coordinators) does not work in all cases. Sentence 179 shows an example in which a truncated token is coordinated with a verb, and sentence 180 gives an example in which two adjacent verbs

	precision	recall	$F_{\beta=1}$
overall	96.08%	96.10%	96.09
AC	94.87%	95.42%	95.14
C	99.73%	99.18%	99.46
LK	99.90%	99.14%	99.52
NC	94.10%	94.39%	94.24
PC	95.58%	94.86%	95.22
VC	98.29%	99.35%	98.82

Table 10.2: Quantitative evaluation results for chunks

do not belong to one verb chunk. This is even the more interesting since, in the case at hand, this sequence of tokens would represent a single VC in a different context. However, in this case, it is an inversion of an LK and a VC.

(179) Denn Bretschneider kennt noch mehr Kinder, deren Anspruch auf
For Bretscheider knows even more children, whose claim to
Assistenzen [$_{\text{VC}}$ hin-/TRUNC und/KON hergeschoben/VVPP
assistance to- and fropushed
wird/VAFIN].
is.

'For Bretscheider knows even more children whose claim to assistance is being shuffled from one authority to the other – and back.'

(180) [$_{\text{VC}}$ Zu/PTKZU sehen/VVINF] [$_{\text{LK}}$ sind/VAFIN] künstlerische und
To see are artistic and
historische Dokumente [...]
historical documents [...]

'Artistic and historical documents are on display.'

A very striking results as regards LKs and VCs is that only 0.66% of all LKs are mistakenly annotated as VCs although every verb chunk just containing a finite verb could be either LK or VC without taking into consideration contextual information. (There were no mistakes for the inverse case.) This is even the more striking as we only take into consideration immediate context for this disambiguation. However, since this distinction is

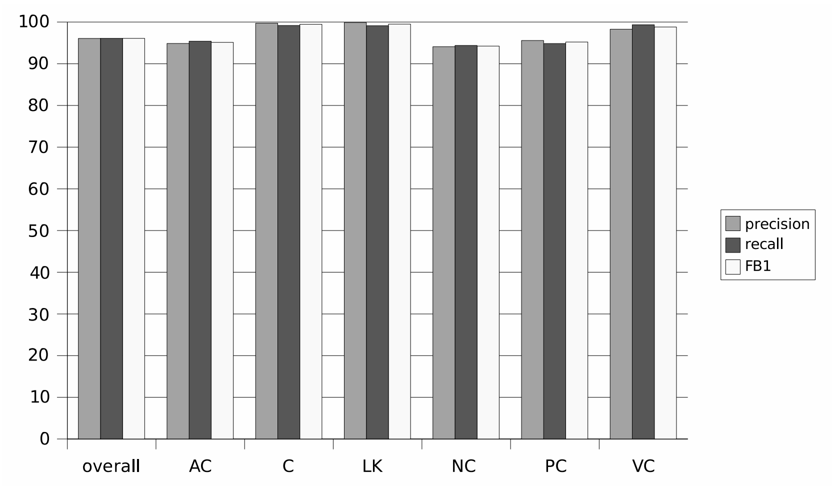


Figure 10.6: Quantitative evaluation results for chunks

crucial for the further annotation of topological fields which itself is crucial for the further annotation of the whole clause including GF annotation, we have put great effort into this distinction and even allowed for the revision of this decision in further steps of annotation. Obviously, this mechanism was successful. Errors which still occurred include the one in sentence 181 which shows two LKs, the second of which occurs in brackets and before a question mark. In this case, our parser assigned the label VC to the second LK. On the whole, there was no consistent error pattern for LKs and VCs except that one could generally say that a lot of errors were caused by unexpected or underspecified punctuation, the latter includes, e.g., hyphens or slashes instead of full stops or commas.

(181) Bei Funny Bones – Regisseur Peter Chelsom [$_{LK}$ darf] (oder [$_{LK}$
With Funny Bones – director Peter Chelsom may (or
muß]?) sich die ZuschauerIn wie ein Kleinkind fühlen.
must?) her-/himself the viewer like an infant feel.

'With Funny Bones director Peter Chelsom, the viewer is allowed to (or has to?) feel like a child.'

Like the chunks LK and VC, the chunk C is both a chunk and a topological field. It is part of the sentence bracket and, thus, crucial for the annotation of the other topological fields like VF, MF or NF. The field C consists of a complementizer, which may be realized by a general subordinator, or of a chunk containing a relative pronoun which may be an NC or a PC.[3] Only the latter case may cause larger problems because the chunk may be complex. Furthermore, there may be chunks containing interrogative pronouns (PWAV) or determiners (PWAT) which may occur in C fields if these interrogatives are used in indirect questions or as relative pronouns (but which may also occur as NCs in VFs). These C fields may also be complex as sentence 182 shows. Generally speaking, the few errors which occurred for C fields were caused by the problem of determining whether a PWAV or PWAT was used as an interrogative or as a complementizer.

(182)

Es	muß	jetzt	auch	um	die	inhaltliche	Frage	gehen,	[$_C$ unter
It	must	now	also	with	the	content	question	dealt,	under

welchen	inhaltlichen	Kriterien]	die	kulturelle	Wirtschaftsförderung
which	content	criteria	the	cultural	business development

gemacht	wird.
made	is.

‘Furthermore, the content question has to be dealt with under which content criteria cultural business development is made.’

The chunks which serve as the phrasal targets for GF annotation received a little lower results. Precision and recall scored fairly equal, which yielded an $F_{\beta=1}$ of 95.14 for ACs, 94.24 for NCs and 95.22 for PCs. It has to be kept in mind that the chunk annotation also includes the complex chunks. Complex chunks were introduced to keep the number of phrasal targets for GF annotation as low as possible in order to contain ambiguity. Sentences 183 and 184 from the test set illustrate the advantage of this approach: Sentence 183 shows the relevant output of the shallow parser. There are four NCs in the sentence, i.e. there are four potential GFs. However, if complex chunks were not annotated, there would be six NCs and, thus, six potential GFs. This would make GF annotation more complex and, thus, more error-prone.

(183)

Zunächst	übernimmt	[$_{NC}$ die	DaimlerChrysler-Tochter	debis]	[$_{NC}$
Initially	takes over	the	DaimlerChrysler daughter	debis	

25,1	Prozent]	[$_{NC}$ des	ehemaligen	Rechenzentrums]	[$_{NC}$ der
25.1	percent	of the	former	data processing centre	of the

[3]In the case that an NC or a PC functions as a C field, it is only evaluated in its C function.

Hansestadt	Bremen].
Hanse town	Bremen.

'Initially, the DaimlerChrysler daughter debis takes over 25.1 percent of the former data processing centre of the Hanse town Bremen.'

(184) Zunächst übernimmt $[_{\text{NC}}$ die DaimlerChrysler-Tochter] $[_{\text{NC}}$ debis] $[_{\text{NC}}$ 25,1 Prozent] $[_{\text{NC}}$ des ehemaligen Rechenzentrums] $[_{\text{NC}}$ der Hansestadt] $[_{\text{NC}}$ Bremen].

However, the annotation of complex chunks has also influence on the evaluation results of the NCs and PCs. Sentence 185 gives an example: The chunk 'Eintracht Frankfurt' (a football club) is a complex chunk (i.e. an apposition). However, since the noun 'Eintracht' did not occur as a noun triggering an apposition in the development set of the shallow parser, the shallow parser fails to recognize the PC 'gegen Eintracht Frankfurt' as a PC but annotates 'gegen Eintracht' as a PC and 'Frankfurt' as an NC. This is one recall error for PCs and one precision error each for PCs and NCs in the evaluation results. Other errors as regards appositions are caused by spelling errors like the one in example 186,[4] in which the noun 'Regisseur' is misspelled.

(185)

$[_{\text{NC}}$ Der Werder-Kapitän]	droht	morgen	$[_{\text{PC}}$ im	wichtigen	
The Werder captain	threatens	tomorrow	in the	important	
Heimspiel]	$[_{\text{PC}}$ gegen	Eintracht	Frankfurt]	auszufallen.	
home match	against	Eintracht	Frankfurt	to drop out.	

'The captain of Werder (Bremen) threatens to drop out in the important home match against Eintracht Frankfurt, tomorrow.'

(186)

$[_{\text{NC}}$ Regiseur	Kirk Jones]
director	Kirk Jones

'film director Kirk Jones'

Further frequent error types are caused by all types of NEs which are realized by sentences (e.g. names of organizations, titles of films and books). In sentence 187, the NP 'des Vereins "Kinder haben Rechte" ' contains a sentence. Since we do not mark these NEs and since we want to annotate the inner structure of such sentences, all chunks are annotated. However, in

[4]Precisely this error occurs twice in the test set.

cases like the one in sentence 187, this leads to errors since the shallow parser recognizes the sequence ‘des Vereins Kinder’ as an NC.[5] Another problem is caused by the frequent occurrence of foreign words mainly from English and Latin sometimes even as whole sentences. If these words are capitalized and occur in typical noun environments, they can be integrated into NCs. In other cases, however, the shallow parser fails like in the case of sentence 188 in which ‘killing fields’ was not integrated into a PC.

(187)

Westerholt	ist	zugleich	erster	Vorsitzender	des	[$_{\text{NC}}$
Westerholt	is	at the same time	first	chairman	of the	

Vereins]	“[$_{\text{NC}}$ Kinder]	[$_{\text{LK}}$ haben]	[$_{\text{NC}}$ Rechte]”.
association	“children	have	rights”.

‘At the same time, Westerholt, is the chairman of the association “children have rights”.’

(188)

In	Albanien	ankommende	Flüchtlinge	hätten	[$_{\text{PC}}$ von	wahren	killing
In	Albania	arriving	refugees	have	of	real	killing

fields]	rund um	Djakovica	gesprochen.
fields	around	Djakovica	spoken.

‘Refugees arriving in Djakovica were said to have spoken of real killing fields.’

Other phenomena which cause problems are wrong segmentations of sentences, which are mainly caused by headlines. In these cases, structures which belong to two different sentences can be included in one chunk and, thus, cause errors. The occurrence of data which has an idiosyncratic structure and precedes or follows the actual text is another problem for the shallow parser since these data are unexpected input. Example 189 shows a piece of data which occurs after film reviews and obviously names the cinemas in which the film is shown. As regards errors occurring in PCs, one can generally that they are caused by problems in recognizing the NC part of the PC and that, thus, NC recognition is the main problem in annotating NCs and PCs.

[5]Quotation marks the way they are used in the TüBa-D/Z corpus did not prove to be useful as cues for annotation and were, thus, not used for it.

(189) (Der Spiegel) CinemaxX, Solitaire (Westerstede), Wall-Kino
(a newspaper) a cinema, a cinema (a town), a cinema
(Ol)
(abbrev. of town)
'no translation'

ACs also serve as targets for GF annotation. The main problems of annotation of ACs are modification phenomena in conjunction with coordination. Although we do not annotate adverbs (ADVs) modifying ACs, we annotate predicative adjectives/adverbs (ADJDs) modifying ACs, since this also reduces the number of potential GF targets and, thus, contains ambiguity. This phenomenon causes few problems since most of the times an ADJD preceding another ADJDs modifies it. Sentence 190 shows an exception which caused an error. The ADJD 'geistig' precedes the ADJD 'kurz' but does not modify it. These errors occur rarely as the evaluation results show. Cases of coordination cause problems because we sometimes have to include modifying structures in order to cover coordination. This sometimes fails in complex cases like the one in sentence 191, in which in the second and third AC the second coordinate is modified by an adverb which is itself modified.

(190) Dann tritt das Kind [$_{AC}$ geistig] [$_{AC}$ kurz] weg und bekommt
Then steps the child mentally briefly aside and gets
vom Unterricht nichts mehr mit.
of the lesson nothing more with.
'The child then briefly drops out mentally and is not able to follow the lesson any more.'

(191) Dafür wäre schon die Schrift [$_{AC}$ zu klein], der Satz [$_{AC}$ zu
For that were already the font too small, the sentence too
lang und zu wenig griffig], das Foto [$_{AC}$ zu ungünstig und zu
long and too little handy, the photo too awkward and too
wenig farbig] und überhaupt alles [$_{AC}$ zu klein].
little colourful and anyway everyhing too small.
'The font would be too small for that, the sentence too long and not handy enough, the Photo too awkward and not colourful enough, and anyway everything too small.'

Table 10.3 and figure 10.7 show the results of the chunks evaluation if TnT tags are used for shallow annotation. The results are satisfactory since

	precision	recall	$F_{\beta=1}$
overall	90.17%	90.86%	90.51
AC	81.10%	79.94%	80.52
C	93.65%	92.12%	92.88
LK	97.39%	96.08%	96.73
NC	87.42%	88.47%	87.94
PC	92.02%	92.44%	92.23
VC	90.95%	94.49%	92.68

Table 10.3: Quantitative evaluation results for chunks with TnT PoS tags

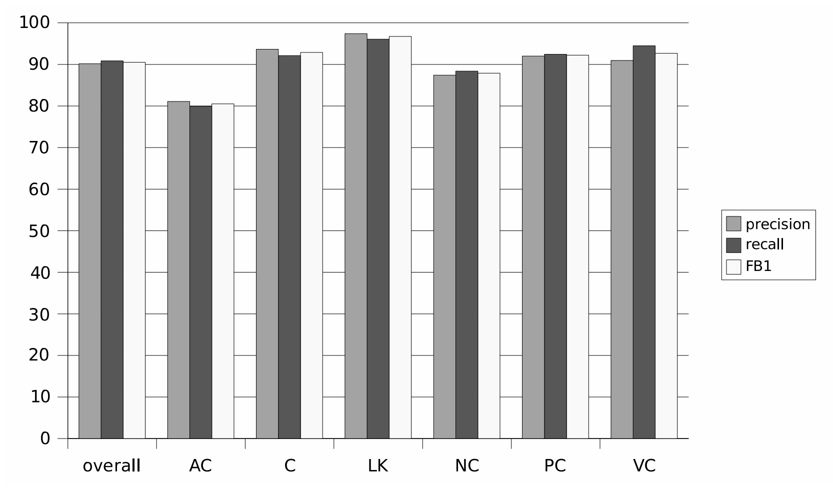


Figure 10.7: Quantitative evaluation results for chunks with TnT PoS tags

	precision	recall	$F_{\beta=1}$
C	97.82%	97.95%	97.89
LK	99.28%	99.08%	99.18
VC	97.09%	98.45%	97.77
VF	94.41%	91.47%	92.92
MF	93.14%	89.52%	91.29

Table 10.4: Quantitative evaluation results for topological fields

the overall $F_{\beta=1}$ is still over 90 (i.e. 90.51). Compared to the $F_{\beta=1}$ of the annotation with gold PoS tags, $F_{\beta=1}$ has dropped by 5.58 points. This is quite moderate if one considers that the tagging accuracy is 95.72% on the test set, i.e. a decrease of 4.28 percentage points as compared to gold tags, and that the wrong annotation of just one PoS tags can prevent the correct annotation of a whole chunk.

10.3.2 Topological Fields

Table 10.4 and figure 10.8 show the quantitative evaluation results for topological fields on the test set (4,174 sentences). The parts of the sentence bracket, i.e. C, LK and VC, have already been evaluated on the test section for chunks. The figures for this much larger test section confirm the good results for these linguistic phenomena. The lower figures for C are due to the fact that in the gold standard the field C also comprises modifying structures (e.g. PPs) in cases in which a C field consists of an NP.

In the test section, there are 3986 VFs. Our parser has annotated 3862, 3646 of which were correct. This yields a precision of 94.41%, a recall of 91.47 and an $F_{\beta=1}$ of 92.92. There were 6343 MFs in the test section. Our parser has annotated 6096 MFs, 5678 were correct. This yielded a precision of 93.14%, a recall of 89.52% and an $F_{\beta=1}$ of 91.29. LVs are not included in table 10.4 because there are just 33 LVs in the corpus which is just a small fraction of all fields. LVs will be discussed later in this section. NFs are not included because their annotation involves the resolvement of grammatical relations. This will become clear when NFs are discussed later on in this section.

The figures for both VFs and MFs are satisfactory since, in both cases, $F_{\beta=1}$ is over 90. As stated before, we only count full matches as correct; but

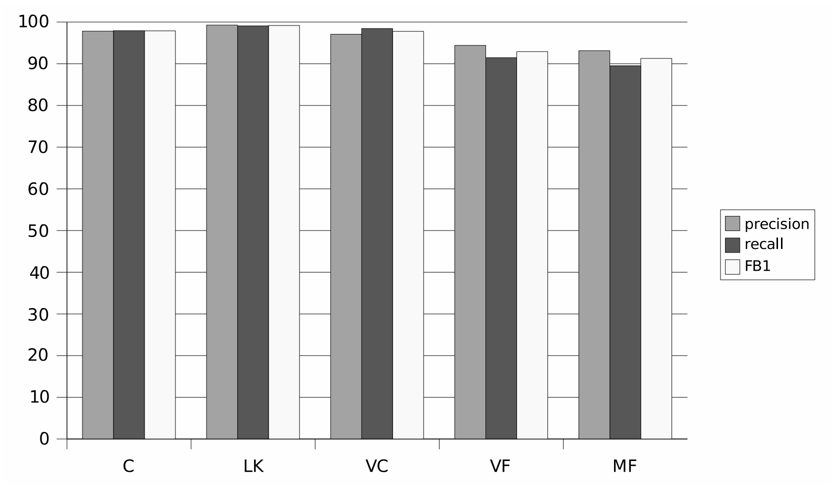
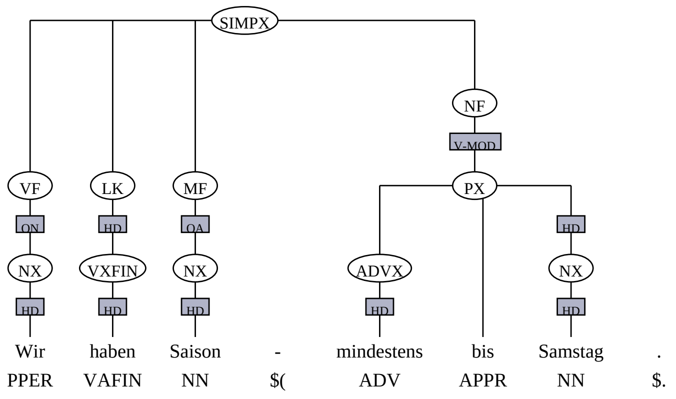


Figure 10.8: Quantitative evaluation results for topological fields

the failure of a full match does not necessarily mean that further annotation is blocked. Sometimes, it is not even influenced and the wrong annotation of the field structure is the only error in the annotation. The sentence in figure 10.9 gives an example. The PP 'mindestens bis Samstag' is in the NF of the clause which is manifested by its exposed position which is underlined by the punctuation (here: the hyphen). However, since such exposed structures without a clear marker (e.g. the right part of the sentence bracket) cannot be detected without a full parse of the tree, the NF is not annotated by the shallow parser. Instead, the shallow parser annotates the MF until the end of the sentence. However, annotation of GFs is not prevented by this error.

Sentence 10.10, which is the second conjunct of a sentence coordination, gives another example: The predicative adjective acts as the right part of the sentence bracket. The following material of the matrix clause is in the NF (here: 'für vertrauliche Gespräche'). The relative clause at the end of the sentence is in the NF, in all cases. However, since at the point of shallow annotation, it is not clear whether the ADJD 'förderlich' is the predicative adjective of maybe an adverb, it cannot be taken as a clear marker for the end of the MF and the beginning of the NF, and, thus, annotation fails. However, further annotation is still possible.

Similar cases like these occur at the beginning of VFs when material which is not part of the sentence in the syntactic sense is segmented as part of the



Wir haben Saison – mindestens bis Samstag.
We have season – at least until Saturday.

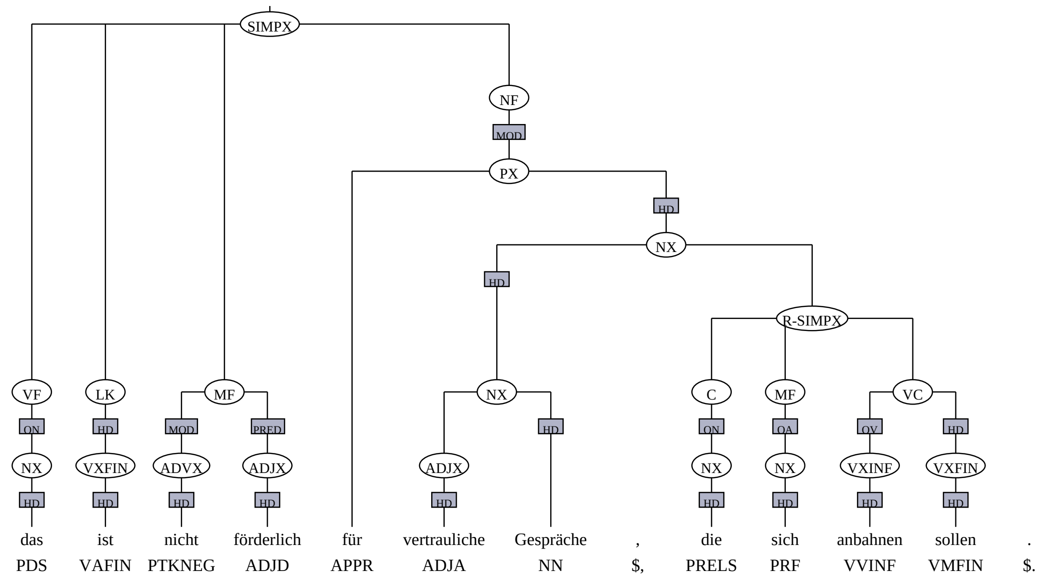
'It's (vegetable) season – at least until Saturday.'

Figure 10.9: NF without clear marker

sentence as a unit of segmentation. These cases may, in fact, obstruct further annotation because the first chunk in a VF may typically act as a GF and, in those case, the first chunk is a chunk which is actually outside the sentence. The sentence in figure 10.11 gives an example: The chunk 'Bürgermeister Runde im Gespräch über die Nutzung seiner alten Hafenreviere' occurs before the actual sentence. It appears to be a pre-headline, and the following part appears to be the actual headline; but, since it is at the beginning of the sentence, the shallow parser annotates it as part of the VF. For the GF annotation, this would be the ideal candidate of a subject. However, the chunk is not even part of the sentence. The hyphen, which separates it from the sentence, is not a reliable marker for separation. This is precisely the reason that the sentence segmenter has included this chunk in the sentence.

(192) Julianne Köhler aber, Theaterbesuchern ohnehin ein Begriff, ist als
Julianne Köhler but, theatre goers anyway a term, is as
sture, treue Musterdeutsche eine Entdeckung.
stubborn, loyal model German a discovery.

'But Julianne Köhler – well known to theatre goers anyway – who plays as stubborn and loyal model German woman is a true discovery.'

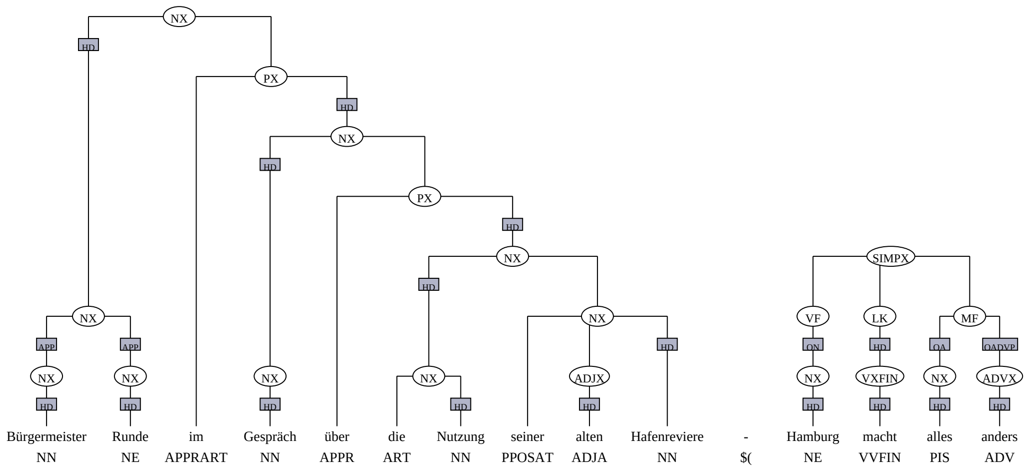


Das	ist	nicht	förderlich	für	vertrauliche	Gespräche,	die	sich
This	is	not	beneficial,	for	confidential	talks,	which	themselves

anbahnen	sollen.
initiate	should.

‘This is not beneficial for confidential talks which are to be initiated’

Figure 10.10: NF after predicative adjective



Bürgermeister	Runde	im	Gespräch	über	die	Nutzung	seiner	alten
Major	Runde	in the	talk	about	the	usage	of his	old

Hafenreviere	–	Hamburg	macht	alles	anders
port area	–	Hamburg	does	everything	differently

‘Major Runde in an interview about the usage of his old port area – Hamburg does everything in a different way’

Figure 10.11: Material occurring before VF

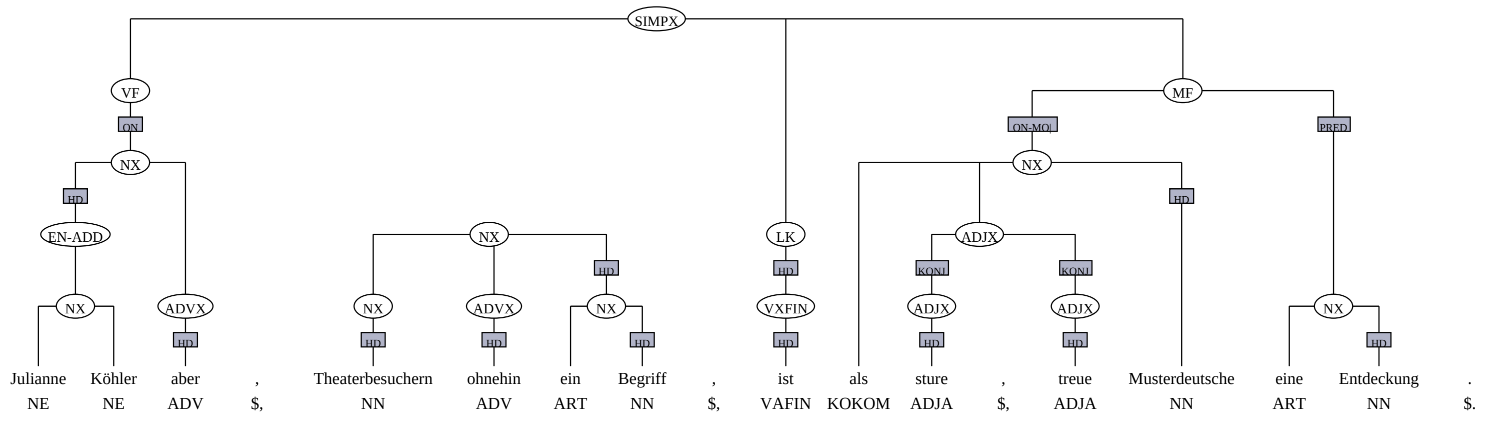
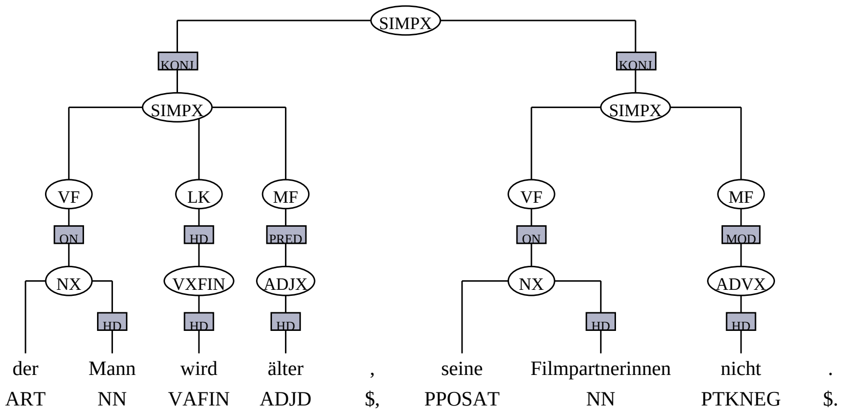


Figure 10.12: Parenthesis after VF



Der	Mann	wird	älter,	seine	Filmpartnerinnen	nicht.
The	man	gets	older,	his	female film partners	not.

'This man gets older but his female film partners do not.'

Figure 10.13:

A further problem to topological field annotation are parentheses like the one in sentence 192, the TüBa-D/Z annotation of which is shown in figure 10.12. Since parentheses typically cannot be detected without the semantic analysis of the sentence, they cannot be annotated by the shallow parser. Thus, they are included in the topological field to which they are the parenthesis. Cases like these may cause problems in further annotation if they occur in the MF because they may be annotated as a certain GF although they are not even part of the respective clause.

While the problems discussed until now do not make further annotation impossible, the following cases certainly lead to problems in further annotation. The sentence in figure 10.13 shows the coordination of two sentences. In the second conjunct the verb is left out. The shallow parser is, thus, not able to determine any field structure and the whole second conjunct is annotated as part of the MF of the preceding clause. Coordination is, in general, a phenomenon which may lead to topological field annotation errors. This is especially true of asyndetic coordinations, i.e. coordinations without a coordinator.

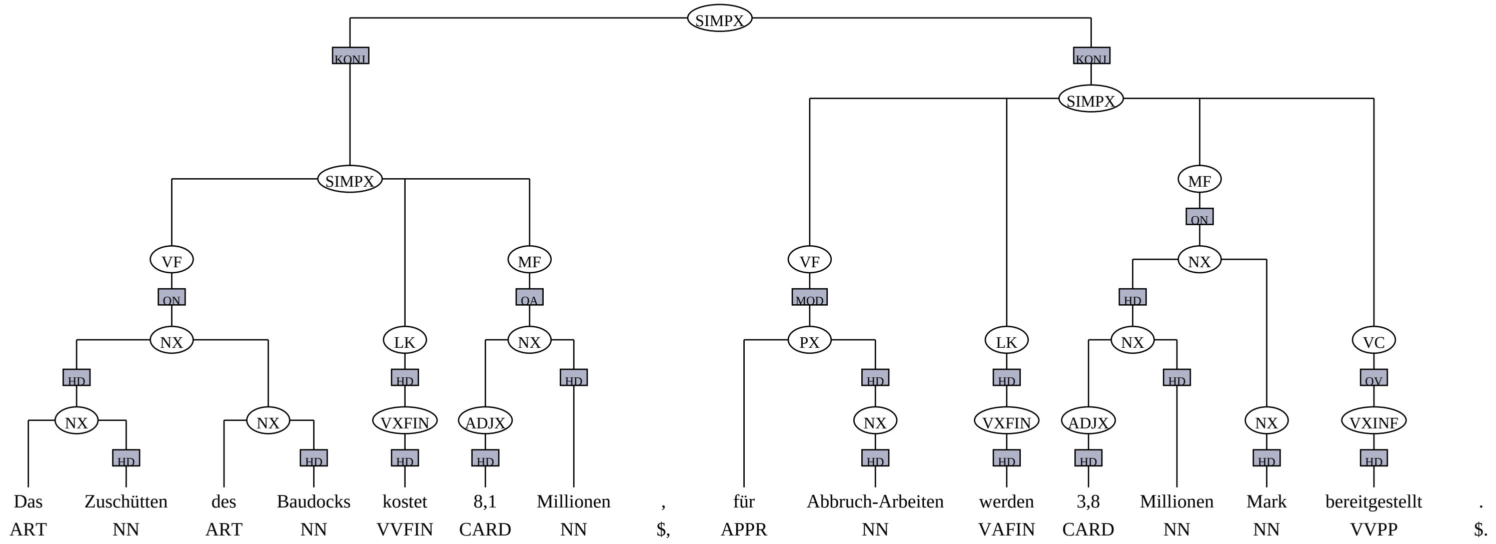

SIMPX
KONJ
KONJ
SIMPX
SIMPX
VF
ON
NX
HD
NX
HD
NX
HD
LK
HD
VXFIN
HD
MF
OA
NX
ADJX
HD
HD
VF
MOD
PX
HD
NX
HD
LK
HD
VXFIN
HD
MF
ON
NX
HD
NX
ADJX
HD
HD
NX
HD
VC
OV
VXINF
HD
Das Zuschütten des Baudocks kostet 8,1 Millionen , für Abbruch-Arbeiten werden 3,8 Millionen Mark bereitgestellt .
ART NN ART NN VVFIN CARD NN $, APPR NN VAFIN CARD NN NN VVPP $.

Figure 10.14: Asyndetic coordination

(193)

Das Zuschütten	des	Baudocks	kostet	8,1	Millionen,	für
The filling-up	of the	construction dock	costs	8.1	millions,	for

Abbruch-Arbeiten	werden	3,8	Millionen	Mark	bereitgestellt.
demolition-workings	are	3.8	millions	marks	mobilized.

‘The filling-up of the construction dock costs 8.1 millions, 3.8 millions marks are mobilized for demolition.’

Sentence 193 gives an example. The annotation is shown in figure 10.14. Since the first clause has no right sentence bracket and since there is no coordinator, there is no clear indication of the end of the first sentence and the beginning of the second. While problems like this may be handled in less complex sentences like this one, it certainly causes problems in more complex ones. In those cases topological field annotation and, consequently clause annotation fails. In general, one can say that problems were caused when no clear markers of the borders of topological fields occurred such as the clear beginning or end of the clause, or the sentence bracket. Further problems were caused by foreign material and NEs which were realized by sentences. Since these problems are more relevant for clause annotation than for topological field annotation, they are discussed in section 10.3.3.

The results for the annotation of the 33 LVs in the test section are rather poor. Only 10 were annotated correctly. One error which appeared more than once is that in a *wenn–dann* construction the wenn-clause was annotated as the VF of the following clause which really is the matrix clause of the whole sentence, a phenomenon similar to the one discussed in section 10.3.3 and illustrated by figure 10.18, there. The main reason for the poor results for LVs is that it is a rare but also quite diverse linguistic phenomenon, which cannot easily be captured.

(194)

[$_{\text{LV}}$	Wenn	man	so	hochwertig	verkaufen	will,]	erklärte	Runde,	[$_{\text{VF}}$
	If	one	so	high-value	sell	want,	explained	Runde,	

dann]	muß	man	”Unverträglichkeiten	ausschalten”.
then	must	one	“incompatibilities	eliminate”.

‘If we want to sell at a high price, explained Runde, we have to “eliminate incompatibilities”.’

There are 1548 NFs in the test section. The shallow parser annotates 1142 but only 818 comply with the ones in the gold standard. This yields a precision of 71.63% and a recall of 52.84% ($F_{\beta=1} = 60.82$). This result can

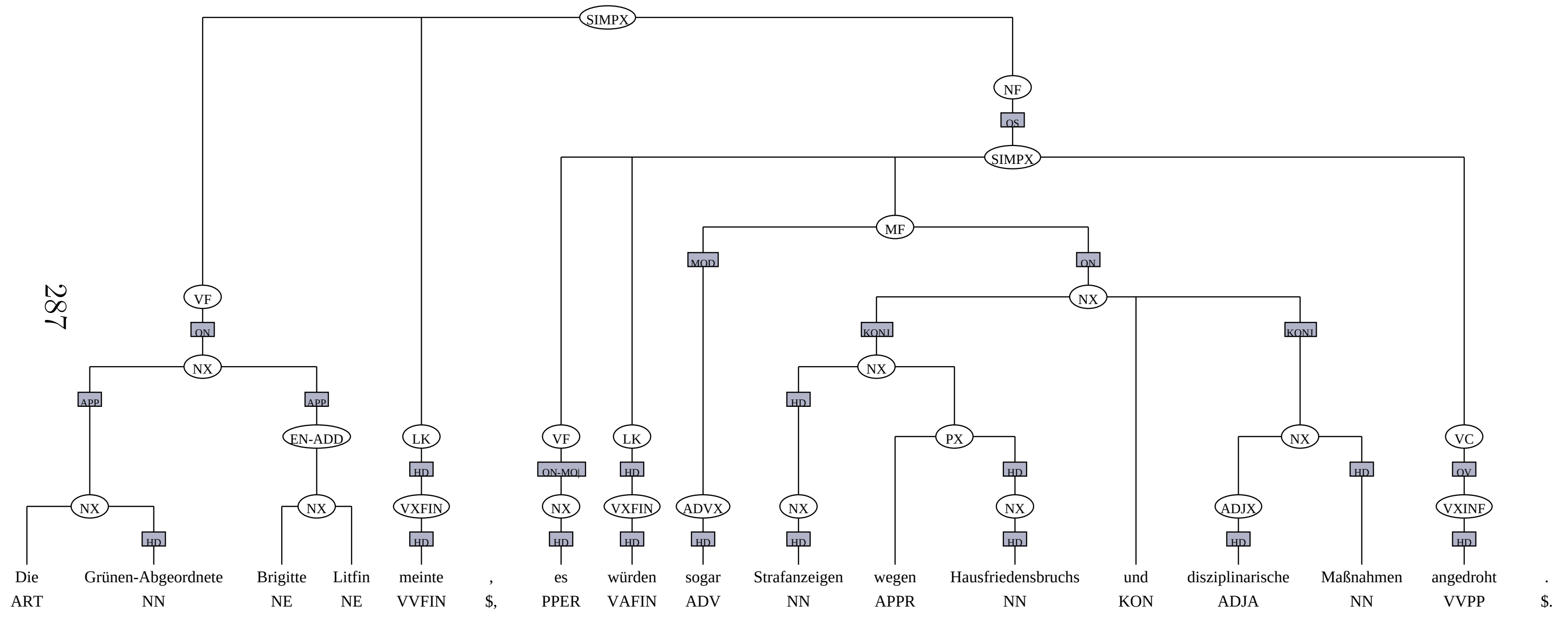


Figure 10.15: V2 clause in NF of V2 clause

be explained with the nature of the shallow annotation in which grammatical relations are not annotated. Sentence 195, which is illustrated in figure 10.15, gives an example: There is a sequence of two V2 clauses in the sentence. Since, at the point of shallow annotation, it cannot be resolved whether the second V2 clause is a conjunct in a coordination or whether it is a the clausal object of the verb in the first V2 clause, this decision is left open. Clausal objects are annotated by the GF annotation component. However, if the second clause is the clausal object of the verb in the first one, then it is in its NF. This information cannot be detected by the shallow parser which is the reason for many of the recall errors for NFs.

(195)

Die	Grünen-Abgeordnete	meinte,	es	würden	sogar
The	Greens-representative	reckoned,	it	would	even

Strafanzeigen	wegen	Hausfriedensbruchs	und	disziplinarische
legal proceedings	because of	trespassing	and	disciplinary

Maßnahmen	angedroht.
measures	threatenend with.

'The representative of the Green Party said that people were even threatened with legal proceedings because of trespassing and with disciplinary measures.'

A second problem concerns modification. This problem affects especially sequences of two or more subordinate clauses. Sentences 196 and 197 give examples: In sentence 196, the second subordinate clause gives the reason for the event in the first subordinate clause; in sentence 197, the second subordinate clause gives the reason for the event in the matrix clause. Consequently, in sentence 196, the second subordinate clause is in the NF of the first subordinate clause and, in sentence 197, the second subordinate clause is in the NF of the matrix clause – together with the first subordinate clause. Structures like these cannot be distinguished by the shallow parser because they can only be distinguished using semantic information or even world knowlegde. Each of these structures may cause precision and recall errors. Since grammatical relations cannot be annotated by a shallow parser, the results for NFs are as poor as they are.

(196)

Klaus	behauptete,	$[_{\text{NF}}$ daß	er	nicht	kommen	konnte,	weil	der
Klaus	claimed,	that	he	not	come	could,	because	the

Zug	Verspätung	hatte].
train	delay	had.

'Klaus claimed that he was not able to come because the train was late.'

	precision	recall	$F_{\beta=1}$
C	93.13%	93.78%	93.45
LK	96.90%	96.44%	96.67
VC	90.52%	94.46%	92.45
VF	93.42%	88.28%	90.78
MF	87.46%	82.59%	84.96

Table 10.5: Evaluation results for topological fields with TnT PoS tags

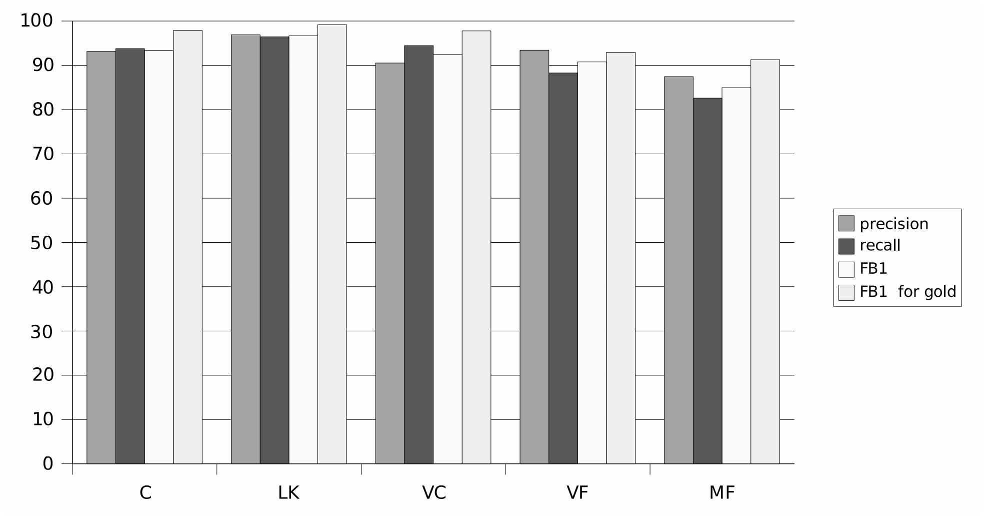


Figure 10.16: Evaluation results for topological fields with TnT PoS tags

(197) Klaus behauptete, $[_{NF}$ daß er nicht kommen konnte, $[_{NF}$ weil
Klaus claimed, that he not come could, because
ihm nichts besseres einfiel]].
him nothing better chimed in.

'Klaus claimed that he was not able to come because he could not come up with anything better.'

Table 10.5 and figure 10.16 give the results for the annotation of topological fields with TnT PoS tags. As regards the elements of the sentence bracket, the decrease in $F_{\beta=1}$ has been moderate for the LK if we keep in mind that tagging accuracy of TnT is 4.28 percentage points below gold standard (cf. figure 10.16). Since the VF is determined by the beginning of a clause and the LK, the drop of $F_{\beta=1}$ for VFs is moderate, too. The other two elements of the sentence bracket, C and VC, have, however, a larger drop in $F_{\beta=1}$. This results in a larger decline of the numbers for the MF with TnT tags, as well.

10.3.3 Clauses

The 4,174 sentences in the test section contain 4,125 matrix clauses and 1,607 subordinate clauses. The number of matrix clauses is even lower than the number of overall sentences although some sentences contain more than one matrix clause. The reason for this is that the definition of a sentence for the sentence segmentation is not purely a linguistic one because there are also some strings of tokens which belong together but do not constitute a sentence in the syntactic sense because they do not contain a verb. These strings are typically headers of or comments to a text as sentence 198 shows.

(198) Aus Paris Dorothea Hahn
From Paris Dorothea Hahn
'From Paris Dorothea Hahn'

Our shallow parser annotates 3,961 V1/V2 clauses in the test section. 3,493 of these are correct. This yields a precision of 88.18% and a recall of 84.68% as shown in table 10.6 and figure 10.17. 1,565 subordinate clauses are recognized and 1,461 of them are correct. This yields a precision of 93.35% and a recall of 90.91% (cf. table 10.6). While the figures for VL clauses are well over 90%, the figures for V1/V2 clauses are not satisfactory, on first sight. However, since only exact matches are evaluated, some of the mismatches which caused errors are no complete failures. If the shallow parser has annotated a structure as a clause which contains just one chunk more or less as the gold corpus, then this yields one precision error and one recall error but these structures may still be useful for further annotation.

Sentence 199 shows an example in which a chunk comes at the end of the segmented sentence but is not annotated as part of neither the matrix clause nor the subordinate clause in the gold standard TüBa-D/Z. In the case at hand, this yields one precision and one recall error each for V1/V2 clauses and for VL clauses because our parser has included the chunk 'ganz im Gegenteil' in the subordinate clause which itself is part of the matrix

	precision	recall	$F_{\beta=1}$
V1/V2 clauses	88.18%	84.68%	86.40
VL clauses	93.35%	90.91%	92.12

Table 10.6: Quantitative evaluation results for clauses

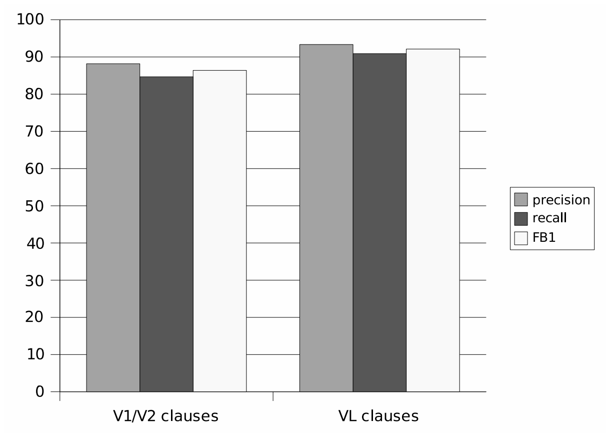


Figure 10.17: Quantitative evaluation results for clauses

clause. The same phenomenon occurs with structures preceding the actual sentence. Although structures like these produce errors in the evaluation, the annotated sentence can still be used for further annotation.

(199)

[$_{\text{V2}}$ Doch	das	bedeutet	keinesfalls,	[$_{\text{VL}}$ daß	Winslets	Wahl	falsch
But	this	means	in no case,	that	Winslet's	choice	wrong

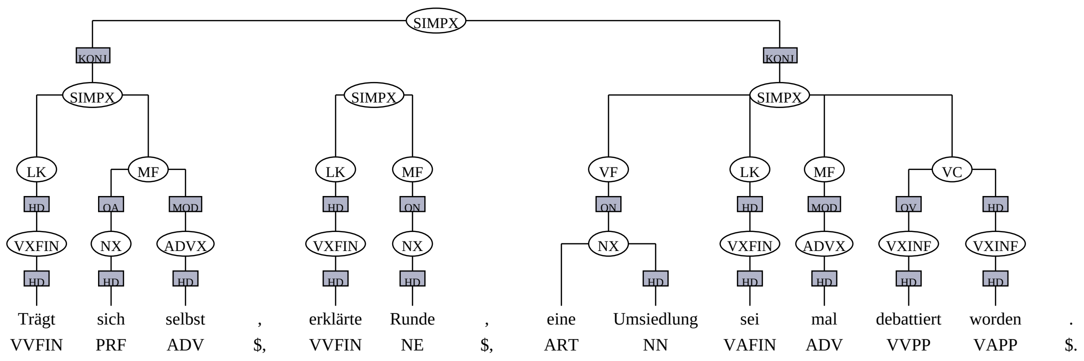
war]]	–	ganz	im	Gegenteil!
was	–	exactly	on	the contrary!

'But this, in no case, means that Winslet's choice was wrong – quite the contrary!'

A further problem causing errors in the evaluation system are sentences of the types as they are shown in figure 10.18. In these sentences, the matrix clause is included in the coordination of the clauses which serve as its clausal object. In the TüBa-D/Z, this relation of the matrix clause to its clausal object is not annotated because this would lead to crossing brackets. These are not allowed in the TüBa-D/Z.[6] Instead, the clausal object is simply left unattached. We do, however, annotate the first V1/V2 clause as part of the second one since it occurs in its VF (Or otherwise the VF would be empty.). This is also done to allow the annotation of at least the first V1/V2 clause as the clausal object of the second. However, as regards the evaluation of shallow clause structures, this leads to one precision and two recall errors each time such a structure occurs.

In the cases in which the annotation failed completely, the main causes of complication were cases of especially asyndetic coordination and cases in which all types of NEs were realized by sentences (e.g. names of organizations and titles of films etc.). Sentence 200 gives an example of the former case and sentence 201 an example of the latter. In sentence 200, the string 'besessen von Verschwörungstheorien' is in the relation of coordination with the preceding 'tief verstört' but there is no coordinator. In fact, even the right part of the sentence bracket intervenes. Structures like these are called *Nachtrag* ('addendum') (cf. Engel, 1996, sec. S 204). Furthermore, the string 'vielleicht sogar einen Bombenanschlag auf ein Justizgebäude in St. Louis geplant und ausgeführt hat' is coordinated with material in the preceding subordinate clause. Again, it is an asyndetic coordination, i.e. there is no coordinator. Since the determination of the scope of coordination is problematic in cases like these, the parser failes to annotate those two coordinations.

[6]This fact also leads to problems for our GF annotation component.



Trägt sich selbst, erklärte Runde, eine Umsiedlung sei mal
Pays itself, declared Runde, a resettlement be once
debattiert worden.
discussed was.

'It pays for itself, declared Runde, a resettlement was once discussed.'

Figure 10.18: Matrix clause included in coordination of subordinate clause

(200) Eine lange, unheilvoll wirkende Einstellung läßt ahnen, [$_{\mathrm{VL}}$
A long, inauspicious seeming setting lets foreshadow,
daß Michael, der Kurse in Terrorismus gibt, tief verstört ist,
that Michael, who courses in terrorism gives, deeply deranged is,
besessen von Verschwörungstheorien], und [$_{\mathrm{VL}}$ daß Olvier eine
obsessed with conspiracy theories, and that Olvier a
zweifelhafte Vergangenheit hat, vielleicht sogar einen Bombenanschlag
dubios past has, possibly even a bomb attack
auf ein Justizgebäude in St. Louis geplant und ausgeführt hat].
on a judiciary building in St. Louis planned and carried out has.

'A setting which seems long and inauspicious forshadows that Michael who gives courses in terrorism is deeply deranged, obsessed with conspiracy theories and that Olvier has a dubious past, possibly has even planned and carried out a bomb attack on a judiciary building in St. Louis.'

(201) [$_{V2}$ [$_{VL}$ Da man die Leiche des Fischmantel-tragenden
Since one the body of the oilskin jacket-wearing
Enterhaken-Killers in [$_{V2}$ "Ich weiß, was Du letzten Sommer
grappling hook-killer in "I know, what you last summer
getan hast"] nie gefunden hat], sind die Alpträume von Julie, der
done has" never found has, are the nightmares of Julie, die
adretten Überlebenden aus Teil I, nicht unbegründet].
preppy survivor from part I, not unsubstantiated.

'Since the body of the grappling hook-killer wearing an oilskin jacket in "I know what you did last summer" was never found, the nightmares of Julie, the preppy survivor from part I, are not unsubstantiated.'

In the case of sentence 201, the annotation has failed because the film title "Ich weiß, was Du letzten Sommer getan hast" is a sentence but acts as a noun. Consequently, this structure does not fit into the grammar of the parser and annotation fails. The only way to avoid problems like these is to annotate NEs like these beforehand. We have, however, not integrated a fully developed NE recognizer into our parser. The initially easy solution to this problem, i.e. to annotate all strings in quotation marks as units, does not prove to be useful since quotation marks, by no means, just include strings which make up constituents but also all types of other material as experience with the TüBa-D/Z has shown. Still, sentence 202, in which the whole clause structure was annotated correctly, shows that the parser was able to annotate even quite complex clauses.

(202) [$_{V2}$ Auch fliegen die Leginonäre (sic!) nach Ohrfeigen und
Also fly the legionnaires after slaps in the face and
Kinnhaken ungefähr so durch die Luft, [$_{VL}$ wie man
hooks to the chin approximately so through the air, as one
sich das bei der Comic-Lektüre immer ausgemalt hatte]] ...
oneself that during the comic-reading always envisioned has ...
[$_{V2}$ aber genau da, bei den Special effects, muß die Mäkelei
but exactly there, with the special effects, must the fault-finding
einsetzen], [$_{V2}$ denn so manche Tricks – etwa der mit dem
begin, for many a trick – for instance the one with the
Elefanten in der Arena – sehen wirklich zu hausbacken aus], [$_{V2}$ da
elephant in the arena – look really too dowdy like, there

	precision	recall	$F_{\beta=1}$
V1/V2 clauses	84.63%	79.27%	81.86
VL clauses	89.25%	84.75%	86.95

Table 10.7: Quantitative evaluation results for clauses with TnT PoS tags

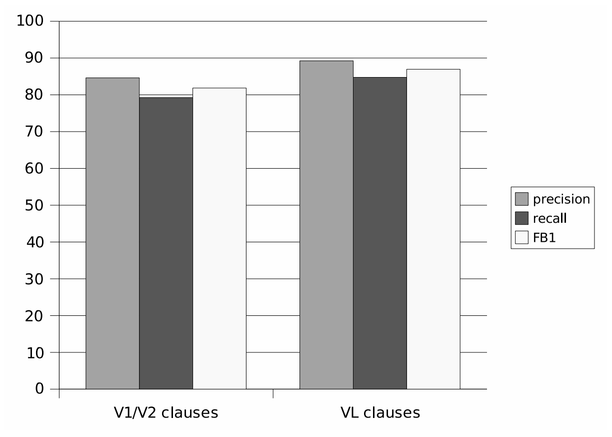


Figure 10.19: Quantitative evaluation results for clauses with TnT PoS tags

erwartet	der	verwöhnte	Kinogänger	Ende	der	90er	Jahre	von
expects	the	spoilt	filmgoer	at the end of the 90's				of

einer	internationalen	Großproduktion	deutlich	bessere
an	international	large-scale production	considerably	better

Effekte,	[$_{\mathrm{VL}}$ zudem	es	am	Geld	offenbar	nicht	gefehlt	
effects,	moreover	it	as regards money		obviously	not	lacked	

hat]].
has.

'The legionnaires also fly through the air after slaps and hooks in a way in which we have always imagined it when reading the comics ... but exactly at that point of the special effects, we have to start carping because many an animation – e.g. the one with the elephant in the arena – really look too dowdy, at points like this, the spoilt filmgoer expects considerably better special effects in an international large-scale production at the end of the 90's, especially because there was obviously no lack of money.'

Table 10.7 and figure 10.19 show the results for the evaluation of V1/V2 clauses and VL clauses with PoS tags assigned automatically by the TnT tagger. Compared to the $F_{\beta=1}$ of the annotation with gold PoS tags, $F_{\beta=1}$ has dropped by 4.54 points for V1/V2 clauses and by 5.17 for VL clauses. Again, as for chunks, this decline can be characterized as moderate since the tagging accuracy for TnT PoS Tags is 95.72% on the test set, i.e. a decrease of 4.28 percentage points as compared to gold PoS tags. As with chunks and topological fields, one tagging error may result in the wrong annotation of a whole clause.

Chapter 11

Grammatical Function Annotation

ABSTRACT: Chapter 11 describes the evaluation of the GF parser. Section 11.1 gives an overview of the evaluation system like the coding of GFs in the gold standard corpus, the conversion of the corpus into a common representation and the methods of evaluation. Section 11.2 explains the concrete procedure of the evaluation, i.e. the test setting. The results of the GF parser evaluation are discussed in section 11.3. Section 11.4 shows how much every component of the GF parser has contributed to the overall results.

11.1 The Evaluation System

11.1.1 Task Description

The general requirements for an evaluation system of a GF parser do not differ from the ones for a shallow annotation evaluation system as described in section 10.1.1: The results of the evaluation system have to be meaningful thus serving as a diagnosis tools for the detection of errors in the parser and for the comparsion with other parsers. In order to achieve its goal, a GF evaluation system needs a gold standard corpus containing GFs, a method to convert the GFs in the gold standard corpus and the GFs annotated by the parser into a common representation and a method for the quantitative and qualitative analysis of errors.

11.1.2 GFs in the TüBa-D/Z Gold Corpus

In the TüBa-D/Z, GFs are marked by edge labels so that they are essentially attributes of the constituents from which the arcs originate. The GFs could not have been annotated as edges themselves because this would have resulted in crossing edges since GFs can occur across topological field boundaries, and the TüBa-D/Z does not allow crossing edges. Figure 11.1 gives an example of a tree in the TüBa-D/Z containing GFs. The tree consists of a V2 clause into which a VL clause is embedded. Both clauses are labeled 'SIMPX'. The embedding SIMPX is introduced by a coordinator field 'KOORD'; and the left part of the sentence bracket 'LK' and the right part 'VC' divide the sentence into the VF, the MF and the NF. Punctuation is not attached.

This topological field description is surface-oriented in that it does not show any of the grammatical relations in the sentence. These relations are shown by the edge labels. The SIMPX in the NF is, for instance marked as 'MOD' which means that it is either an ambiguous modifier or a modifier of the whole sentence. In the case that, e.g., the SIMPX was a relative clause, the modified element would, in addition, be marked by the element which it modifies, e.g. OAMOD for modifier of accusative object. The scope of the MODs is confined by the structures in which they occur. Thus, in the embedded SIMPX, the ADVX (adverbial chunk or phrase) 'dann', which serves as a MOD, can only be a modifier of either the whole embedded SIMPX or any elements within it.

In the whole sentence in figure 11.1, the GFs ON and OA occur twice. They are also marked by edge labels. However, GFs are defined as relations **between** constituents and these relations are not explicitly marked in the syntactic tree. Still, the occurrence of the related lexical items of the GFs is limited by the topological fields and the clause structure in the sentence in figure 11.1. The NX 'Uwe Beckmeyer' is assigned the functional label ON. It is, in fact, the ON sub-categorized by the verb 'aufweichen' in the VXINF. This is not explicitly marked but inherent in the constituent structure since one can infer the relation between the ON and the lexical item it is sub-categorized by from the constituent information because the ON is always sub-categorized by the lexical verb in the clause if there is any. In those cases in which there is a predicative adjective in the clause (e.g. *Er ist des Wartens müde.*), this adjective is the sub-categorizing lexical item. Thus, although the sentence in figure 11.1 contains two ONs, there is no ambiguity or underspecification in the syntactic structure.

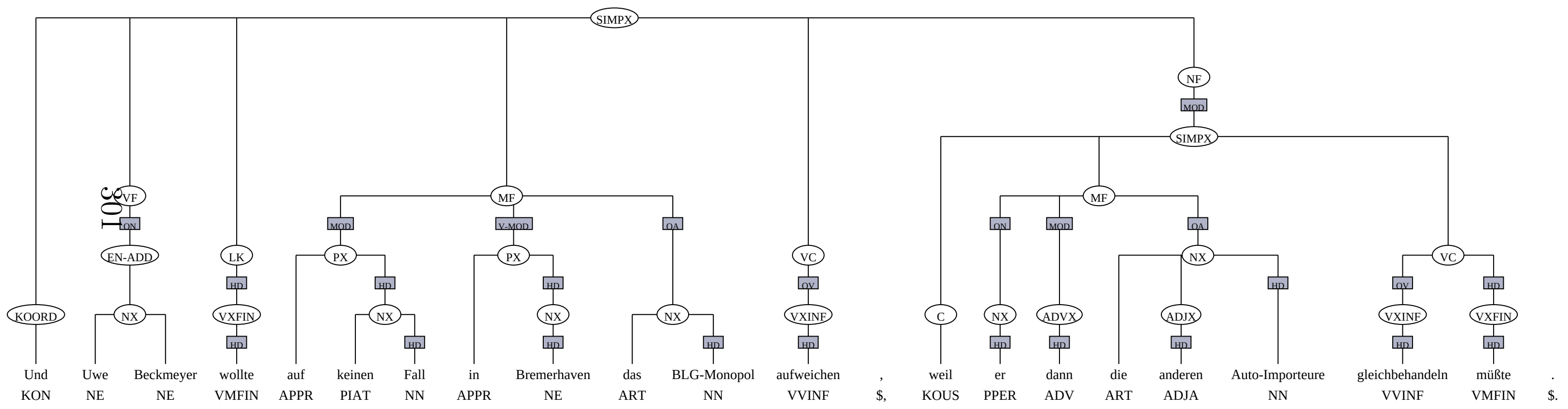


Figure 11.1: Syntactically annotated tree from the TüBa-D/Z

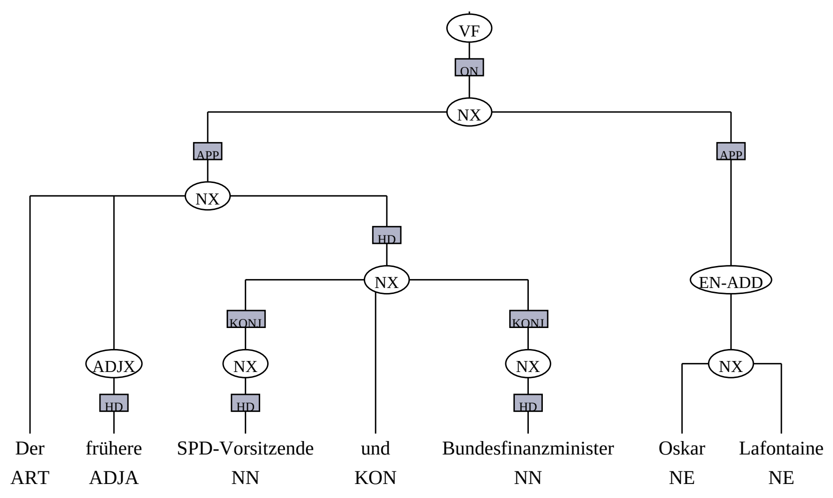


Figure 11.2: Complex phrase with grammatical function

11.1.3 Conversion into Common Representation

Just like they are annotated in the TüBa-D/Z, we evaluate GFs as attributes of constituents and not as relations between them. The sites of their attachment are sufficiently disambiguated by the existing shallow parsing structure, which shows the topological fields and clause boundaries. In order to compare the output of the parser and the gold data, we convert both such that they just contain the GF information. As the constituents carrying the GF information, we choose the heads of the phrases or chunks which represent a certain GF. For OPPs we decided not to choose the preposition but the head noun of the NX dominated by it.

The evaluation of the GFs is, thus, head-based. There remains the question of how to exactly convert the GF information in the TüBa-D/Z into a head-based representation such that it can be matched with the converted head-based representation of the output of the parser. Furthermore, there is the question which representation can be used for comparing these data. As regards the conversion of the data from the TüBa-D/Z, we first transform the TüBa-D/Z into a dependency format. Kübler and Telljohann (2002) show that this is possible. They use the head/non-head distinction which is coded in the TüBa-D/Z to convert the constituency structure into dependency relations; and they use the functional labels to convert the GFs into dependency relations. In only a few cases in which heads are not present (e.g. coordinations and appositives), specific operations have to be used to determine the dependencies within a constituent. Using this information, it is possible to allocate the correct GF to the respective head of a phrase.

The process of finding the head of a phrase can be illustrated by figure 11.2 in more detail: Figure 11.2 shows that, sometimes, the GF constituents are relatively complex and conversion is non-trivial. The phrase 'der frühere SPD-Vorsitzende und Bundesfinanzminister Oskar Lafontaine' in figure 11.2 is, for instance, a multi-embedded structure. The whole phrase serves as an ON. However, on the whole, it consists of six NXs. In order to determine the head of the whole phrase, the information about the head/non-head distinction can be used but it is not sufficient. For those cases in which there is no such distinction, precisely defined operations are applied to select one noun as the head.

The top NX has the edge label ON. The two NXs it dominates do not have any HD label but both have the label APP. In those cases, the first of the two is chosen and the process of finding the head proceeds with the first of the dominated NXs. This NX dominates just one further NX which carries the label HD and is, thus, chosen for further processing. This NX dominates two coordinated NXs. In those cases, both NXs receive the label KONJ in the TüBa-D/Z. We, again, choose the first one to continue processing. This preterminal NX contains just one word which is also the head of the NX and is, consequently, chosen as the head of the whole phrase and assigned the function label ON. In general, in those cases in which there is no clear head information, we choose the first feasible constituent for further processing.

In order to compare gold standard annotation and automatic annotation, the representation of both data sources has to be changed such that a comparison is possible. Since the heads are non-recursive events (i.e. no head whatsoever is contained in any other head), it is possible to represent the evaluation as a tagging problem, just like for the evaluation of the shallow structures. Thus, gold data and parser output are converted such that the heads of the (potential) GFs receive the respective GF labels. Then the two data sets are mapped and can be compared in order to yield various evaluation measures. Furthermore, errors in the corpus can now be located and made subject to a qualitative evaluation. Figure 11.3 shows a sample sentence in the evaluation format. The heads of the GFs, i.e. 'Papst-Besuch' as ON and 'Rolle' as OA, are tagged with their GF tags and the other tokens are tagged with 'O' for 'outside'.

`Der O O`	`the`
`Papst-Besuch I-ON I-ON`	`pope-visit`
`in O O`	`to`
`Bukarest O O`	`Bukarest`
`spielt O O`	`plays`
`sowohl O O`	`both`
`außenpolitisch O O`	`foreign-policy-wise`
`als O O`	`and`
`auch O O`	`also`
`für O O`	`for`
`Rumänien O O`	`Romania`
`selbst O O`	`itself`
`eine O O`	`a`
`große O O`	`big`
`Rolle I-OA I-OA`	`role`
`. O O`	

Figure 11.3: Sample of evaluation representation

11.1.4 Methods of Evaluation

Just like for the annotation of shallow structures, we want to know precision, recall and F-Measure for the annotation of GFs. Counting the errors is achieved using the data in the representation as shown in figure 11.3, which is the same data-format as for the evaluation of the shallow structures. Thus, again, the problem can be viewed as a tagging problem. Table 11.1 shows the results of the evaluation. These results are one important diagnosis tool for the developer and user of the GF parser. In the development phase of the parser, these results enable the grammar constructor to make inferences and work on the crucial problems in the annotation algorithm.

The results of the quantitative evaluation as presented in table 11.1 contain a lot of useful information about the performance of the parser. There are, however, also areas which are not dealt with. The results in table 11.1 do, for instance, not present the absolute figures for errors. This is especially striking for the figures for OG which score $F_{\beta=1}$=100.00. In fact, there is only one OG in the corresponding test section. There is also no way to tell which categories are mixed up with each other. In order to evaluate these

	precision	recall	$F_{\beta=1}$
overall	85.52%	80.02%	82.68
OA	82.94%	82.77%	82.86
OD	83.66%	59.40%	69.47
OG	100.00%	7.69%	14.29
ON	90.93%	91.27%	91.10
OPP	72.06%	48.50%	57.98
OS	76.92%	60.67%	67.83
PRED	78.53%	72.87%	75.59

Table 11.1: Quantitative evaluation results for GFs

problems, there is a need to find out about the type of errors.

In order to find out more about the types of errors which the parser produces, we apply two further evaluation methods which are a first step towards a qualitative evaluation in the sense that they indicate the reasons for certain errors. This can be important for the user in order to know the specific problems in the parser output. It is even more important for the developer in order to improve the parser. If we know, for instance, that the precision of ON is 90%, we need to know for the improvement of the parser what kind of errors are the remaining 10%. It is important to know whether these errors occurred because the parser annotated constituents as ONs which are no GFs at all or whether the parser annotated constituents as ONs which are other GFs. In the latter case, it is essential to know with which GFs ON has been confused. Figure 11.4 shows the output of the diagnosis tool for the detection of such types of errors. Furthermore, it is important to ensure that errors are eliminated without introducing new ones. Thus, it is also necessary to have an evaluation which points to the relevant changes after the system has been worked on. That way, it can be controlled which kind of overall effect a change to the parser has. Figure 11.5 shows the output of the tool which points to errors in the parser output.

Figure 11.4 gives the number of correctly and incorrectly annotated GFs in decreasing order. The categories in the left column (marked with a '$\star$') represent the target category and the categories in the right column (marked with a '+') represent the selected category. In the ideal case, the two categories match (i.e. the system has selected the correct target). These are

```
     1103    *ON     +ON        <-- true positives
      463    *OA     +OA
X     286    *ON                <-- false negatives
      218    *PRED   +PRED
X     186    *OPP
X     158    *OA
      149    *OPP    +OPP
X     103    *PRED
       82    *OS     +OS
X      61            +OPP       <-- false positives
       45    *OD     +OD
X      42            +ON
X      40            +OA
X      39    *OD
X      32    *OS
X      31            +PRED
X      25    *OA     +ON        <-- false negatives/false positives
X      24            +OS
X      20    *ON     +OA
X      18    *OD     +OA
X      13            +OD
X       7    *ON     +PRED
X       6    *ON     +OS
X       5    *PRED   +ON
X       5    *OD     +ON
X       1    *OPP    +OA
X       1    *OG
```

Figure 11.4: Output of the diagnosis tool for the types of errors – gold standard categories left, automatic categories right – errors are marked with an 'X' – one example for each category is marked with an arrow

the true positives. In some cases, the system does not return any category for the target category (false negatives). This has an unfavourable effect on recall. In other cases, the system selects a category where there is none (false positives). This has an unfavourable effect on precision. In still other cases, the system mixes up categories. In those cases, it depends on the perspective what type of errors they are. For instance: If an OA is annotated as an ON, then this is a false negative from the perspective of the OA since it has not been annotated. Seen from the perspective of the ON, it is a false positive since the selected ON does not match the correct GF. Errors like these are disadvantageous for precision **and** recall. True negatives (i.e. tokens without category which were correctly not annotated) are not taken into consideration in this tool since they are not relevant for the evaluation.

With the tool at hand, it is possible to take a closer look at the types of errors of the parser, especially at the absolute numbers of errors. The figures in the sample output show that, at this point of development, most errors were caused by false negatives and that false positives are only relatively frequent for OPPs as compared to the absolute number of OPPs. What is striking is that the confusion of GFs of different categories which has a bad effect on both precision and recall is not very frequent as compared to other negative effects. For instance, 1103 ONs are annotated correctly but only 20 are confused with OAs. Findings like these show the grammar constructor where to put the emphasis of the work on the system.

In order to investigate the cause of different error types, one can query the development corpus in the vertical format as it is shown in figure 11.5. As can be seen, this representation is a simplification of the representation in figure 11.3 and additionally contains the sentence identification numbers. This makes working with it more user-friendly. Figure 11.5 shows the vertical format for a sentence in which ON and OA have been mixed up. In this case, the cause of the error is clear. The morphology of both noun chunks is ambiguous and, in such a case, the annotation system decides in favour of the unmarked constituent order 'ON OA'. In other cases, the cause of the error is less clear and it is necessary to have a look at the vertical bracketed format as shown in figure 11.6.

Figure 11.6 shows a case in which an ON was not correctly assigned because of a failure of the chunking system which is part of the shallow parsing system. This error was discovered using the format as in figure 11.5. If no obvious cause can be identified in the simple vertical format, the sentence identification number allows finding the respective sentence in the vertical

```
-------------
sentence 67
-------------
So                                          so
sieht                                       sees
es      *OA     +ON                         it
auch                                        also
die                                         the
Staatsanwaltschaft      *ON     +OA         public prosecutor
in                                          in
Freiburg                                    Freiburg
:
```

‘This is also the opinion of the public prosecutor in Freiburg’

Figure 11.5: Spotting errors in parser output using simple vertical format

bracketed format. In the case at hand, the chunking system did not correctly annotate ‘anderen’ as an elliptical noun chunk followed by a noun chunk just containing proper nouns (NE) (Vilma Nunez). This happened because adjectives (ADJA) can in fact modify proper names. Here, the correct head was integrated into a prepositional chunk and, thus, the annotation system failed assigning the correct ON. Instead the apposition ‘Präsidentin’ was annotated as ON. This component of the diagnosis tool helps identifying causes of errors and, thus, eliminating them.

11.2 The Test-Setting for Evaluation

For the evaluation of the GF annotation, we have used the test section of the TüBa-D/Z (cf. section 10.2). The test section was not used for the development of the parser but exclusively for its evaluation thus avoiding tuning the parser to the test set. In order to judge the quality of a parser, it is important to know how well it does on unseen data and not how well it does on the data which has been used for its development. This is important both for machine-learning approaches and manual approaches. During its development, the parser was evaluated repeatedly on the development set with the methods described in section 11.1.4, and occurring errors were mended.

```
-------------
sentence 41
-------------
(VSEC
    {VF
        [PC -PRED-
            .APPR    Zu                                    as
            [NC
                .NN      Gast ] ] }                        guest
    [VCLAF
        .VAFIN   ist ]                                     is
    {MF
        [PC
            .APPR    unter                                 amongst
            [NC
                [AJAC
                    .ADJA    anderen ]                     others
                .NE      Vilma                             Vilma
                .NE      Nunez ] ]                         Nunez
        .$,      ,
        [NC -ON-
            .NN      Präsidentin ]                         president
        [NC
            [NC
                .ART     der                               of the
                [AJAC
                    .ADJA    nicaraguanischen ]            Nicaraguan
                .NN      Menschenrechtsorganisation ]      human rights
            [NC                                            organization
                .NE      CENIDH ] ] }                      CENIDH
    .$.      . )
```

'Amongst others, Vilma Nunez, president of the Nicaraguan human rights organization CENIDH, is a guest.'

Figure 11.6: Spotting errors in parser output using vertical bracketed format

However, the parser was not altered after the beginning of the evaluation on the test set.

For the evaluation in the following sections, the data was parsed by our parser under various preconditions in order to test their influence on parsing performance. We tested, e.g., the importance of morphological information for GF assignment by applying the parser without any meaningful morphological annotation (i.e. just with default morphological ambiguity classes), by applying it with the morphological ambiguity class assigned by DMOR and also by applying it with the morphological ambiguity class assigned by DMOR and reduced by our own morphological reduction component (as described in chapter 8).

Furthermore, we tested the influence of the PoS tags on the performance of the parser by applying the parser once with the gold (i.e. hand-annotated) PoS tags and once with the PoS tags as annotated by the statistical PoS tagger TnT (*Trigrams'n'Tags*; Brants, 2000). In addition, we tested the contribution of the component not using sub-categorization (SC) information to the parsing results. This component is applied on top of the component using SC information. It is needed because not all verbs can be assigned SC frames or because they do not contain the respective SC frame as realized in the sentence in question.

11.3 Discussion of Overall Evaluation Results

Table 11.2 shows the final results of the evaluation of the GF parser. The results are illustrated in figure 11.7. Overall precision is 85.52% and overall recall 80.02%. This yields an $F_{\beta=1}$ of 82.68. The results are, however, varying with respect to the different GFs and this fact is one of the main issues of this section. The differences are caused by factors which can be located in the German language itself and by factors which are connected with the lexicon used for annotation and the corpus used for evaluation. Besides the quantitative evaluation of the GF annotation of the various GFs, we will present a qualitative evaluation of errors.

The results for ON are best among the results for GFs: All values are over 90% (precision: 90.93%; recall: 91.27%; $F_{\beta=1}$: 91.10). Figure 11.8 shows the ratio of the 9.07% of cases in which the parser wrongly assigned the GF ON (i.e. the precision of ON). In nearly half of the cases (48.82%), the parser assigns ON where there is no GF which we annotate with our parser, at all.

	precision	recall	$F_{\beta=1}$
overall	85.52%	80.02%	82.68
OA	82.94%	82.77%	82.86
OD	83.66%	59.40%	69.47
OG	100.00%	7.69%	14.29
ON	90.93%	91.27%	91.10
OPP	72.06%	48.50%	57.98
OS	76.92%	60.67%	67.83
PRED	78.53%	72.87%	75.59

Table 11.2: Final results of evaluation of GF annotation

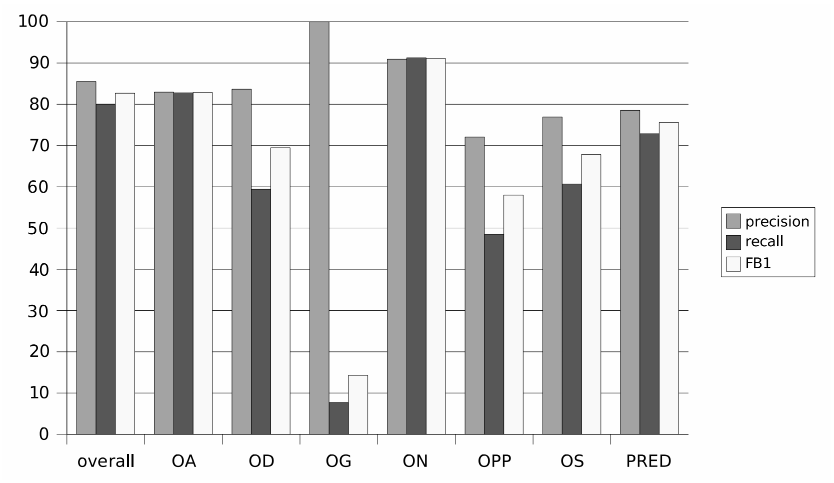
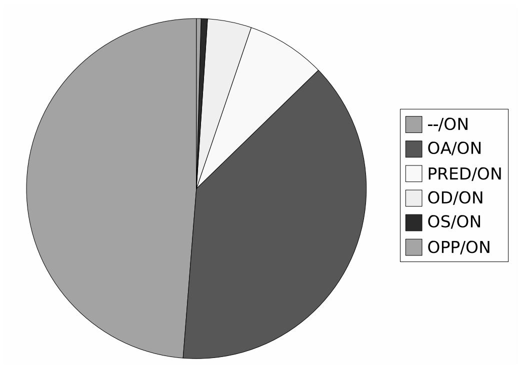


Figure 11.7: Final results of evaluation of GF annotation



(a) ratio of errors

error type	number	percentage
–/ON	249	48.82%
OA/ON	196	38.43%
PRED/ON	39	7.65%
OD/ON	21	4.12%
OS/ON	3	0.59%
OPP/ON	2	0.39%
	510	

(b) number and percentage of errors

Figure 11.8: Ratio of errors for ON precision

The main reason for this is that modifying structures are assigned the label ON. Frequent examples of this case are dummy 'es' and genitive modifiers. In sentence 203 the dummy 'es' is the modifier of the ON 'ein vierter Anbieter' (hence ON-MOD). However, the parser assigned 'es' the label ON and 'ein vierter Anbieter' the label PRED. Sentence 204 shows a case in which the ON consists of a common noun modfied by a genitive proper noun (i.e. 'Winslets Wahl'). Since the proper noun 'Winslets' was not unambiguously assigned the case feature *genitive*, we were not able to recognize it as a modifier to 'Wahl' and, thus, 'Winslets' was incorrectly assigned the label ON.

(203) Denn [$_{\text{ON-MOD}}$ es] ist [$_{\text{ON}}$ ein vierter Anbieter] [$_{\text{PRED}}$ am Markt].
For it is a fourth supplier at the market.

'For there is a fourth supplier at the market.'

(204) [$_{\text{ON}}$ Das] bedeutet keinesfalls, [$_{\text{OS}}$ daß [$_{\text{ON}}$ Winslets Wahl] [$_{\text{PRED}}$
That means not at all that Winslets choice
falsch] war].
wrong was.

'That does not mean at all that Winslet's choice was wrong.'

(205) Fünf Tage und Nächte trommeln [$_{\text{OD}}$ sich] [$_{\text{ON}}$ mehr als 300
Five days and nights drum themselves more than 300

Schlagzeuger] [$_{\text{OA}}$ die Finger] wund.
percussionists the fingers sore.

'For five days and nights, more than 300 percussionists will play the drums until their fingers get sore.'

(206) Unter dem Titel queer-studies veranstaltet [$_{\text{ON}}$ die
Under the title queer studies organized the
Arbeitsgemeinschaft LesBiSchwule Studien] [$_{\text{OA}}$ einen offenen Tag]
study group queer studies an open day
[...]

'The study group queer studies organized an open day entitled 'queer studies'.'

Adverbial NCs are a further problem for the annotation of ONs, especially if they occur in positions in which ONs occur in the unmarked GF order. Sentence 205 gives an example: The adverbial NC 'fünf Tage und Nächte' occurs in the VF of the sentence. Since it also has the feature *nominative* in its morphological ambiguity class, it is annotated as ON. The correct ON is, however, the third NC in the sentence (i.e. 'mehr als 300 Schlagzeuger'). It is possible to mark NCs as temporal expressions using lexical information. However, since lexical items expressing temporal expressions can also be part of GFs as sentence 206 shows, annotation of temporal expressions is a non-trivial task.

With 38.43% of all precision errors, the annotation of OAs as ONs comes second. The reason for this is that 90% of GFs in the TüBa-D/Z are either ON or OA (cf. figure 11.26 in section 11.4.1) and because the majority of morphologically ambiguous chunks has the features *nominative* **and** *accusative* in its ambiguity class (cf. figure 11.25 in section 11.4.1). Sentence 207 shows a typical example: If two NCs are **both** ambiguous as regards the features *nominative* and *accusative*, the parser decides in favour of the unmarked GF order, which is 'ON precedes OA'. This decision proves to be right in most cases but wrong in sentence 207. The sentence can easily be understood by humans because of the semantic selectional restrictions of the verb (i.e. people (especially spokespersons) affirm facts and not the other way round). In order to correctly annotate sentences like these, semantic information would have to be integrated into the parser because there are no other cues than the verb as to which NC is the ON and which the OA. This can be illustrated by sentence 208 in which the verb is replaced with another one which only allows the reading ‘ON precedes OA’ since events affect people and not the other way round.

(207)

[$_{\text{OA}}$ Das]	bestätigte	[$_{\text{ON}}$ Firmensprecher	Peter Schmidt]	
That	affirmed	company spokesman	Peter Schmidt	
gestern	gegenüber	der	taz.	
yesterday	towards	the	taz.	

‘Company spokesman Peter Schmidt affirmed this fact in a talk with our newspaper, yesterday.’

(208)

[$_{\text{ON}}$ Das]	betraf	[$_{\text{ON}}$ Firmensprecher	Peter Schmidt].
That	affected	company spokesman	Peter Schmidt.

‘This event affected company spokesman Peter Schmidt.’

The third most frequent precision error for ONs is annotation of PREDs as ONs (7.65%). The main reason for this is that there are some cases in which difference between ON and PRED could only be determined by semantic information. Typically, the ON precedes the PRED. Only if the first nominative NC does not agree in number with the verb, the second is annotated as the ON in TüBa-D/Z. However, there are also cases like the one in sentence 209 in which the order is different in TüBa-D/Z. In those cases, our parser confuses ON and PRED. A less frequent error is the confusion of OD with ON (4.12%). Since ODs occur a lot rarer than ONs, they are annotated quite conservatively by contrast to ONs. Thus, in some cases, if morphologically ambiguous and/or in typical ON position, an OD may be confused with an ON. Typical cases are chunks containing or consisting of the tokens ‘der’ and ‘ihr’.

Sentence 210 gives an example: Since ‘der’ may be either *nominative* or *dative*[1] and since ‘BBB’ is an acronym which is morphologically unspecified, ‘der BBB’ may be both the OD or the ON of the sentence. However, we do not annotate ODs if they also contain *nominative* in their morphological ambiguity class since this has lead to too many annotation errors. Thus, the parse in the case of sentence 210 failed to annotate the OD and annotated it as ON. The two remaining groups of mistakes each account for less than 1%. OS/ON mistakes are the equivalent of OA/ON mistakes on the level of clauses and the OPP/ON confusion is caused by the fact that it was not

[1]We ignore the case of *genitive* here since the verb does not sub-categorize for an OG.

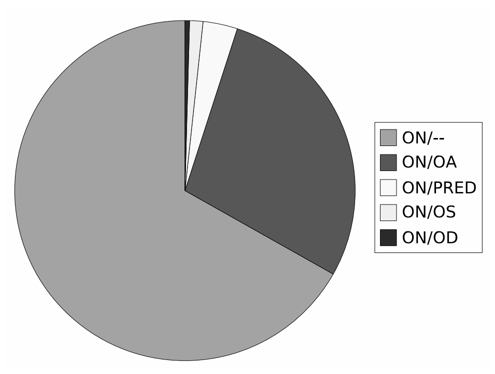
taken into account that the pronoun 'worum' is a pro-form of a PP and not an NP.

(209) [...] bei einem Artikel, [$_{\text{PRED}}$ dessen Autor] [$_{\text{ON}}$ Rudolf Sharping]
with an article, whose author Rudolf Scharping
ist [...]
is
'with an article the author of which is Rudolf Scharping'

(210) Erst nach der Verschrottung der alten Anlage sei [$_{\text{OD}}$ der
Only after the scrapping of the old construction be the
BBB] aufgefallen, [$_{\text{ON}}$ daß die notwendigen Sanierungsarbeiten
BBB struck, that the necessary reconstruction
insgesamt rund 2 Millionen Mark kosten].
in total about 2 million Marks costs.
'Only after the scrapping of the old construction, it struck the BBB that the necessary reconstruction would cost about 2 million Marks, in total.'

Figure 11.9 shows the recall errors for ON. In many ways, the errors in recall are the mirror image of the errors in precision because, if another chunk than the ON is annotated as the ON (error in precision), then the correct ON is not annotated (error in recall). In 66.94% of the recall errors, the correct ON was not annotated with any GF. In a large number of cases, the reason for this would be the same as was the case with the most frequent error in precison, i.e. an adverbial chunk or clause received the label ON. Furthermore, another GF could wrongly have been assigned the label ON. Only in very few cases, the reason for the fact that the ON is not annotated at all is that no ON whatsoever is annotated in the clause because we try to assign an ON in every finite clause.

The second most frequent error for recall (28.23%) is that the ON is annotated as OA. This is the mirror image of the precision error in which an OA is annotated as ON. The reason for the fact that the number of cases in which an ON is not annotated because it is assigned the label OA is lower than the other way round (140 compared to 196) is that we assign an ON in every clause and the number of annotated ONs is simply higher than the number of annotated OAs. The third most frequent error is the confusion of ONs with PREDs (3.23%). Again, fewer ONs are annotated as PREDs than the other way round (16 compared to 39) as with the ON/OA errors.



(a) ratio of errors

error type	number	percentage
ON/–	332	66.94%
ON/OA	140	28.23%
ON/PRED	16	3.23%
ON/OS	6	1.21%
ON/OD	2	0.40%
	496	

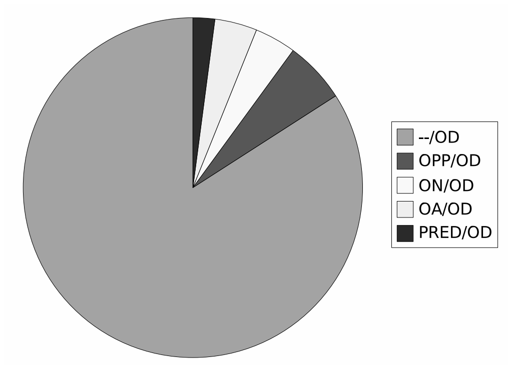
(b) number and percentage of errors

Figure 11.9: Ratio of errors for ON recall

The reason is the same: PREDs are annotated more conservatively than ONs. ON/OS mistakes are the equivalent of ON/OA mistakes on the level of clauses. They occur rarely (1,21% of all ON recall errors). Only two ON/OD errors occur in the test section as compared to 21 OD/ON errors. The reason is, again, that ODs are annotated conservatively as compared to ONs.

OGs are a very special case in the annotation of GFs since there are only 13 of them in the test section, which is 0.14% of all nominal GFs. Only one of them is annotated correctly. The others are not annotated, at all. Consequently, OG has a precision of 100% and a recall of 7.69%. The reason for this result is that we only assign the label OG if the respective verb in the sentence sub-categorizes for an OG. While ODs can also be annotated purely on the grounds of their morphological features because every dative is assigned the label OD in the TüBa-D/Z, this is not possible for OGs since the vast majority of genitive chunks are modifying chunks and not GFs. In the cases in which the OGs were not annotated, the SC frames of the respective verbs were not contained in the lexicon. The only verb sub-categorizing for an OG which was in the lexicon and the test section was 'entledigen' which sub-categorizes for an ON, a reflexive OA and an OG.

For the GF OD, there is a precision of 83.66% and a recall of 59.40%. This yields an $F_{\beta=1}$ of 69.47%. The reason for the comparatively low recall can be found in two facts which are related with the SC frames of verbs: On the one hand, *free datives* (cf. section 7.3 for a definition) are annotated as ODs in the TüBa-D/Z. However, free datives are not sub-categorized for by the verb and



(a) ratio of errors

error type	number	percentage
–/OD	42	84%
OPP/OD	3	6%
ON/OD	2	4%
OA/OD	2	4%
PRED/OD	1	2%
	50	

(b) number and percentage of errors

Figure 11.10: Ratio of errors for OD precision

are, thus, not contained in the SC frame. On the other hand, a SC lexicon is never exhaustive such that not all ODs, even if sub-categorized for by the verb, are contained in its SC frame. The figures presented and discussed in section 11.4.3 which deals with the effect of the non-lexical component (i.e. the component not using any SC frame information) show that of the 59.40% of recall for OD only 35.73% percentage points were annotated using SC frame information, i.e. only 60.15% of all ODs annotated by the parser were annotated using SC frames, the by far lowest ratio for all GFs.

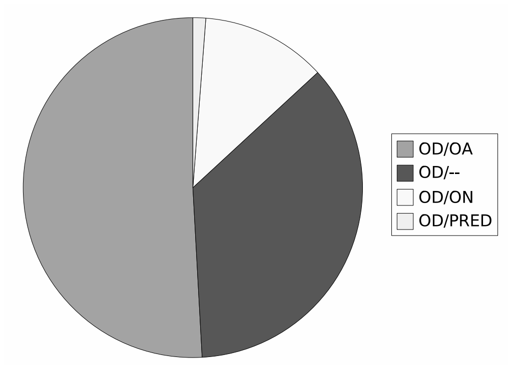
The comparatively low recall for ODs is, thus, caused by the fact that, for the majority of occurrences, OD is not listed in the SC frame of the respective verb. On the other hand, however, if an OD is listed in the SC frame, this does not necessarily mean that it is realized in the sentence in which the verb occurs. In fact, it seems to be optional in many cases, and, in those cases, the parser annotates chunks which are no GF at all, as OD as can be seen in figure 11.10. This group of errors is by far the largest (84%). Like with ONs, the main reason for this was that a modifying chunk was annotated as OD. In the case of ODs, this modifying chunk is almost exclusively a genitive modifier. The reason for this is that ODs are only annotated if the target chunk does not contain the features *nominative* or *accusative* in its morphological ambiguity class in order to avoid the wrong annotation of the far more frequent ONs or OAs. Thus, confusion of ON and OA occurs rarely as can be seen in figure 11.10.

However, a strategy in which ODs were only annotated in the case they were unambiguously dative did not prove to be effective. Thus, ODs were annotated even if they contained the feature *genitive*. In order to keep confusion of ODs with modifying genitives as low as possible, they were annotated quite conservatively, i.e. either if they were in the SC frame of the verb (for the lexical component) or if they did not occur in positions in which genitive modifiers could occur (for the non-lexical component). The majority of cases in which genitive modifiers were annotated as ODs were, thus, annotated by the lexical component of the parser. The ambiguity in these sentences could only be resolved by semantic information as can be seen in sentence 211 in which our parser incorrectly assigned the post-modifying genitive phrase 'der Niederlassung' (*the branch*) the label OD. The verb 'sagen' does, in fact, sub-categorize for an ON, an OS and an OD and the chunk 'der Niederlassung' does contain the feature *dative*. Human readers realize that 'der Niederlassung' is not the OD because branches cannot be talked to.

(211) [$_{\text{OS}}$ Bei der einwöchigen Abgabe eines Probewagens der
With the one-week disposal of a test car of the
A-Klasse an die Ehefrau von Broemme habe [$_{\text{ON}}$ es] [$_{\text{OA}}$ sich] [$_{\text{OPP}}$
A-Class to the wife of Broemme have it itself
um einen normalen Vorgang] gehandelt], sagte [$_{\text{ON}}$ der Leiter]
about a normal procedure concerned said the manager
der Niederlassung Berlin gestern.
of the branch Berlin yesterday.

'The one-week disposal of an A-Class test car to the wife of Mr Broemme has been a normal procedure said the manager of the Berlin branch, yesterday.'

Figure 11.11 shows the detailed results for the evaluation of OD recall: By contrast to ON and also to OA, the major error category for recall is not that the GF is not annotated, at all, but that it is assigned the wrong label. 50.86% of all recall erros were caused by the annotation of OD as OA. The reason for this is that OA is far more frequent than OD (6.7 times; cf. figure 11.26). Thus, in cases of doubt, OA was assigned rather than OD. This especially concerned the tokens 'sich' and 'uns' which can be both dative or accusative. Sentence 212 gives an example: The token 'uns' serves as the OD in the sentence. It was, however, assigned the label OA by our parser. OD is far less often confused with ON (12%) because pronouns with identical form are not so frequent. The second most frequent error category, i.e. that ODs are not annotated at all, is caused by the fact that ODs are not in the SC



(a) ratio of errors

error type	number	percentage
OD/OA	89	50.86%
OD/–	63	36.00%
OD/ON	21	12.00%
OD/PRED	2	1.14%
	175	

(b) number and percentage of errors

Figure 11.11: Ratio of errors for OD recall

frame of the respective verb and that the non-lexical component annotates ODs quite conservatively for the reason mentioned above. Sentence 213 gives an example: the verb 'angleichen' does not contain the SC frame for an ON, a reflexive OA and an OD and, thus, OD is not annotated. The non-lexical component fails to assign OD because 'den Privaten' also contains the feature *accusative*.

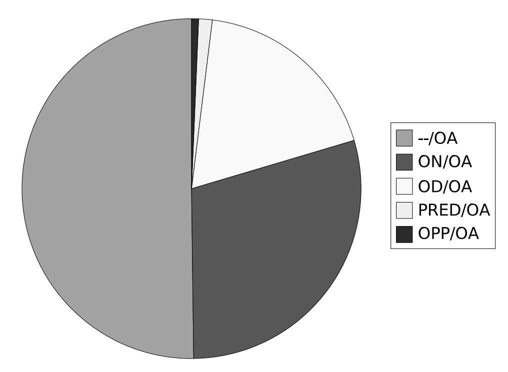
(212) $[_{ON}$ Sie] wollen $[_{OD}$ uns] weismachen, $[_{OS}$ daß $[_{ON}$ es] $[_{OA}$ keine
They want us make believe, that it no
Unterdrückung] im Kososvo gibt] [...]
oppression in the Kosovo exists

'They want to make us believe that there is no oppression in Kosovo.'

(213) $[_{ON}$ Matschke] bedauerte, $[_{OS}$ daß $[_{OA}$ sich] $[_{ON}$ der
Matschke regretted, that itself the
öffentlich rechtliche Rundfunk] in seinen Sendeformen
public-law broadcast in its broadcasting formats
immer mehr $[_{OD}$ den Privaten] angleiche].
more and more the privates adapt.

'Matschke regretted that the broadcast under public law more and more adapts to the private broadcast in its broadcasting formats.'

For the GF OA, precision and recall are nearly equal. Precision scores 82.94% and recall 82.77%. This yields an $F_{\beta=1}$ of 82.86 which is closest to



(a) ratio of errors

error type	number	percentage
–/OA	240	50.21%
ON/OA	140	29.29%
OD/OA	89	18.62%
PRED/OA	6	1.26%
OPP/OA	3	0.63%
	478	

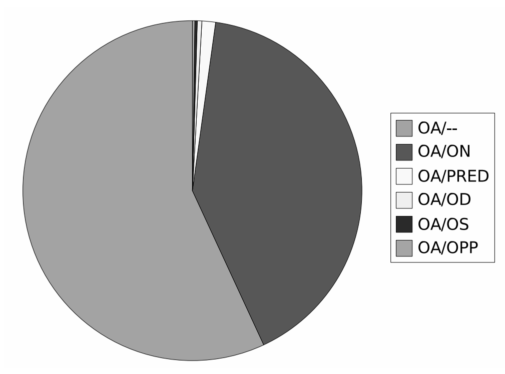
(b) number and percentage of errors

Figure 11.12: Ratio of errors for OA precision

the overall $F_{\beta=1}$ of 82.68. As regards evaluation results, OA is, thus, the most average of all GFs. Figure 11.12 shows that there are three main error types for OA precision. The most frequent is that OAs were annotated where there was no GF, at all, which accounts for half of the mistakes (50.21%). The second is that ONs were confused with OAs which accounted for nearly a third of the errors in OA precision (29.29%). Third comes the annotation of ODs as OAs (18.62%). As regards recall, the are just two main error types: In over half of the cases (56.81%), the OA was not annotated, at all; and in 41.09% of the cases it was annotated as ON. Most of the errors have already been discussed above since they also concerned the GFs ON and OD.

The two most frequent error types for OA precision have corresponding error types in the ON precision. For ONs and OAs, the number of cases if which no grammatical function is annotated is about half the number of errors. Like with ONs, most of the errors are a result of modifying structures being annotated as OA. Sentence 214 gives an example: The first potential NC serving as the OA is 'ein einziges Mal', which is annotated as OA by our parser. However, in fact, the NC 'knackigen Jeanshintern' is the OA. Cases like this, could only be avoided by using larger lexica for temporal expressions.

(214) Allerdings: [$_{\text{ON}}$ Sharon Stone] ist auch als zu Sorgenrunzeln
However, Sharon Stone is even as to worry wrinkles



(a) ratio of errors

error type	number	percentage
OA/–	271	56.81%
OA/ON	196	41.09%
OA/PRED	6	1.26%
OA/OD	2	0.42%
OA/OS	1	0.21%
OA/OPP	1	0.21%
	477	

(b) number and percentage of errors

Figure 11.13: Ratio of errors for OA recall

genötigte Mutter, [$_{\text{ON}}$ die] nur ein einziges Mal [$_{\text{OA}}$ knackigen
coerced mother, who only one single time firm
Jeanshintern] zeigen darf, [$_{\text{PRED}}$ sehr sexy].
jeans backside show allowed is, very sexy.

'However, Sharon Stone is very sexy even as mother forced to make a worry-wrinkled face who is only once allowed to show her her firm backside in jeans.'

The second most frequent error type is the confusion of ONs with OAs. This is the mirror image of the confusion of OAs with ONs. However, in absolute numbers, the pair ON/OA (140 instances) occurs less than the pair OA/ON (196). The reason for this is that, if OA is confused with ON in a clause, ON is not necessarily confused with OA in the same clause. Since not all verbs sub-categorize for an OA but nearly all sub-categorize for an ON, ON is annotated more often (and it does, in fact occur more often in the corpus) and, thus, it is more often confused with OA than vice versa. The third most frequent error for OA precison (18.62%) is the one the caused half the errors in OD recall. If one also considers that OA precision is over 80% and OD recall less than 60%, this fact illustrates that a smaller error for a more frequent GF can have a large effect for a less frequent GF.

Figure 11.13 shows the dominance of two main errors for OA recall. OA is either not annotated, at all, or it is annotated as ON. We have already extensively discussed the confusion of OA and ON. The reason for the failure to annotate OA is caused by the same fact as for ONs, i.e. that a modifying chunk was annotated as OA. However, since not all verbs sub-categorize for OA, it it is also caused by the fact that the lexicon is not exhaustive. The non-lexical component works as a back-up, then. However, it only annotates OAs if they do not contain the features *genitive* or *dative* in their morphological ambiguity class in order to keep precision high. This leads to some recall errors like the one in sentence 215 which is caused by the fact that 'geben' has no lexical entry for an ON, an OA and an OPP with 'für' as is realized in the sentence 215.[2] Thus, the non-lexical component would have to annotate the OA. However, 'Anlaß' also contains the feature *dative* and, thus, OA is not annotated. If we did allow the annotation of OAs in cases in which it also contained the feature *dative*, this would rise OA recall but also lower OA precision.

(215) [$_{\text{ON}}$ Ihr Schwachpunkt] gibt [$_{\text{OA}}$ Anlaß] [$_{\text{OPP}}$ für die gute
Her weak point gives cause for the positive
Botschaft] des Films.
message of the film.

'Her weak point brings out the positive message of the film.'

Prepositional objects (OPPs) are a special case of GFs because the main decisive feature in them is the preposition and not the morphological features as with nominal GFs. In the TüBa-D/Z, two different kinds of prepositional objects, which have to be distinguished from free adjuncts (V-MOD), are defined:

1. *A PP is called* **OPP** *within a sentence if the sentence were ungrammatical without the OPP (or if there was at least a very noticeable change of meaning). For instance,* Sie gehen [OPP gegen die Faschisten] vor./ Das Gesetz ist [OPP in Kraft] getreten.
2. *A PP is called* **FOPP** *if it can be left out of this specific sentence without causing ungrammaticality (or a very noticeable change of meaning) and if its preposition is selected by this specific verb. For instance,* Insgesamt berichtet die

[2]In fact, this is the support verb 'Anlaß geben für etwas' (to give cause for sth., i.e. to cause sth.). However, the entry is also missing in the support verb lexicon.

Polizei [FOPP von 19 Festnahmen und 98 Ingewahrsamnahmen]./Später würden wir [FOPP über Auswandern] nachdenken. *Here, the prepositions select these specific verbs and the PPs cannot be added to any arbitrary verb (which is possible for free adjuncts). In addition, in passive clauses, the subject of the original active clause, which has the form of a prepositional phrase, is marked as FOPP* (Sie wurden [FOPP von Autonomen] umringt.).

3. *A PP is called* **V-MOD** *if its preposition is not selected by this specific verb, i.e., it can be exchanged by any other modifying PP, and similarly, this PP can occur with arbitrary verbs* (Nur [V-MOD im griechischen Lager] gab es Probleme). *Typical V-MODs are temporal or local adjuncts specifying time and location of the action, event, or state expressed by the verb.*

(TüBa-D/Z; Telljohann, Hinrichs, and Kübler, 2003, p. 123)

We do not take over the distinction between OPPs and FOPPs and annotate both of them as OPPs because, in the IMSLex, there is no distinction between optional and obligatory prepositional objects. The definition of prepositional objects for the TüBa-D/Z as quoted above confirms what was said about OPPs in section 7.3, i.e. that the distinction between the adverbial and the object function of the PP is not a clear-cut one and that, thus, the definition necessarily remains vague, to a certain extent. There are relatively precise alternation rules for obligatory prepositional objects, i.e. that the PP cannot be deleted without the sentence becoming ungrammatical, unless there is a 'noticeable change of meaning'. However, there remains the problem of the definition of 'noticeable' and the problem of obligatory adverbial PPs.

But what is more, is that there are no clear-cut alternation rules for optional prepositional objects. It is just stated that the verb selects the preposition, and the PP in question 'cannot be added to any arbitrary verb (which is possible for free adjuncts)'. However, Eisenberg (1999), p. 294–295, gives an appropriate counterexample: In sentence 216, the PP 'am Gartentor' is a locative adverbial PP which can be left out. However, in sentence 217, it is an obligatory prepositional object, at least in the meaning of 'working on sth.'. The preposition is now semantically empty just like the preposition 'on' in the English verb, which has lost its locative meaning. It is, thus, the semantic change which plays an important role in the definition of prepositional objects.

(216)	$[_{\text{ON}}$ Harald]	unterhält	$[_{\text{OA}}$ sich]	am	Gartentor.
	Harald	converses	himself	at the	garden door.

'Harald is chatting at the garden door.'

(217)	$[_{\text{ON}}$ Harald]	arbeitet	$[_{\text{OPP}}$ am	Gartentor].
	Harald	works	at the	garden door.

'Harald is working on the garden door.'

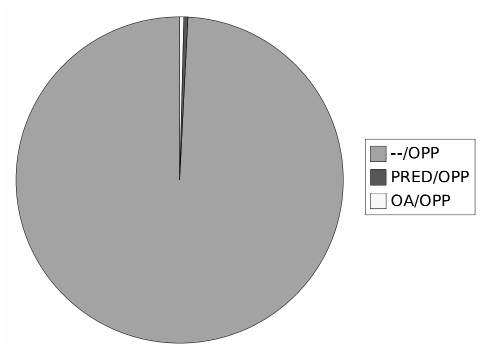
(218)	$[_{\text{ON}}$ Harald]	arbeitet	im	Haus.
	Harald	works	in the	house.

'Harald is working in the house.'

Sentence 218, furthermore, shows that the PP 'am Gartentor' can, in fact, be replaced by another PP, but not in the sense of the verb 'working on'. The distinction between obligatory **and** optional prepositional objects, on the one hand, and adverbial PPs, on the other hand, thus, relies on both syntactic and semantic tests. While the syntactic replacement or deletion test is relatively clear-cut, the decision whether a 'noticeable' semantic change has taken place is not clear-cut because the change is gradual. It may be obvious in sentences 217 and 218 but the discussion is far more difficult in real-world language data. Thus, the problem concerning the definition of prepositional objects as stated by Eisenberg (1999), which was already quoted in section 7.3 remains:

> *Tatsächlich ist es noch niemandem gelungen, die Objekts- und Adverbialfunktion der PrGr* [Präpositionalgruppe; our remark] *durchgängig zu trennen.* (Eisenberg, 1999, p. 295)
>
> *In fact, no one ever succeeded in universally separating object and adverbial functions of the prepositional group.*

The results for both precision and recall show that the task of annotating OPPs is harder than annotating ONs or OAs: OPP precision is at 72.06% and OPP recall at 48.50%. One of the main problems with annotating OPPs is the definition of OPPs discussed above. On the one hand, this definition



(a) ratio of errors

error type	number	percentage
–/OPP	262	99.24%
PRED/OPP	1	0.38%
OA/OPP	1	0.38%
	264	

(b) number and percentage of errors

Figure 11.14: Ratio of errors for OPP precision

is not a clear-cut one even within one approach and, on the other hand, the definition of the IMSLex, which we use for annotation, and the TüBa-D/Z, which we use as a gold standard, differ. Secondly, PCs which serve as post-modifying chunks or as adverbial chunks may be confused with OPPs if they have the same preposition as an OPP occurring in the SC frame of the respective verb in the sentence. This is especially a problem for precision. Since OPPs are realized by PCs and (except for some cases of PREDs) no other GF which we evaluate is realized by PCs nearly all errors (99.24%) are caused by the annotation of chunks which are no GFs, at all, as figure 11.14 shows.

(219) Auf elf Jahre in Forschung und Lehre kann [$_{ON}$ die 55jährige]
On eleven years in research and teaching can the 55-year-old
zurückblicken.
back look.

‘The 55-year-old can look back on eleven years in research and teaching.’

(220) Maria blickt zurück (auf das Meer / durch den Nebel / in den
Maria looks back (on the sea / through the fog / in the
Wald).
woods).

‘Maria looks back (onto the sea/through the fog/into the woods).’

Sentences 219–221 give examples for OPP precision errors caused by definition problems of prepositional objects: In sentence 219, ‘die 55jährige’ is annotated as the ON in the TüBa-D/Z. No other GFs are annotated. According to the definition used in TüBa-D/Z, ‘auf elf Jahre in Forschung und Lehre’ is no OPP because the PP can be deleted withoput causing ungrammaticality (or noticeabbly changing the meaning) and because the preposition is not selected by the verb, i.e. ‘zurückblicken’ can occur freely with other prepositions. This is in fact the case as can be seen in sentence 220. However, while ‘zurückblicken’ is clearly used figuratively in sentence 219 since eleven years are no physical object to be regarded, it is clearly not used figuratively in sentence 220. As regards the TüBa-D/Z definition, there remains the question in how much the figurative and the non-figurative reading of the verb differ noticeable in meaning.

> *Ein P-Objekt liegt vor, wenn die PP bzw. deren Proform übertragen gebraucht wird, d.h. wenn sich die Bedeutung der PP im übertragenen Gebrauch direkt von einer adverbialen Bedeutung ableitet.* (IMSLex; Eckle-Kohler, 1999, p. 128)
>
> *A P-Object* [equivalent to OPP; our remark] *is existent if the PP or its pro form is used figuratively, i.e. if the meaning of the PP in its figurative use can be directly derived from an adverbial meaning.*

As regards the definition of prepositional objects in Eckle-Kohler (1999), p. 128 (quoted above), at least this case is clear since ‘zurückblicken’ is used figuratively and the PP is, consequently, seen as an OPP. Thus, the PP receives the label OPP, which yields one of the 264 precision errors in the test section. In this case, the error was caused by a different definition in TüBa-D/Z and IMSLex. Although the case is arguable, the error is incontestable because we use the TüBa-D/Z as our gold standard. Sentence 221 gives another example: ‘Bremen’ is annotated as the ON and ‘sich’ as the OA. However, in the IMSLex ‘verschulden’ also sub-categorizes for an OPP with the preposition ‘mit’. Thus, ‘mit 200 Millionen Mark’ is annotated as an OPP by our parser. In this, case it is arguable whether the PP is used figuratively. It can, however, not be replaced by a PP with any other preposition.

(221)

$[_{ON}$ Bremen]	will	$[_{OA}$ sich]	mit	200 Millionen	Mark	neu
Bremen	wants	itself	with	200 million	Marks	new

verschulden,	um	$[_{OA}$ seinen	Großmarkt]	umzusiedeln.
indebt,	in order to	its	superstore	relocate.

‘Bremen wants to incur new debts of 200 million Marks in order to relocate its superstore.'

The examples above have shown the problems caused by the definition of OPP. Since this definition is not easy to apply, one can assume that the inter-annotator agreement would not be as high as with, e.g., ONs. This means that even one single annotator would not be able to reach the gold standard.[3] Although this has not been quantified, the baseline for OPPs would, thus, be lower than for ONs or OAs. There are, however, also errors caused by the fact that PCs may occur which have the same preposition as a potential OPP for which the verb in question sub-categorizes but which are post-modifying or adverbial chunks. Sentence 222 gives an example: In this sentence, 'der Bund der Steuerzahler' is annotated as the ON and 'weitere Einsparungen' as the OA. Our parser has, however, also annotated the PC 'zur Konsolidierung' as an OPP because there is also a SC frame which demands an ON, an OA and an OPP with 'zu' in the SC frame of 'fordern'. But in sentence 222, 'zur Konsolidierung' does not serve as an OPP. In fact, it is ambiguous: It either modifies the preceding OA or the verb. Hence, it is annotated as MOD (ambiguous modifier) in the TüBa-D/Z. Sentence 223 gives an example of an instantiation of the SC frame which demands an ON, an OA and an OPP with 'zu' for the verb 'fordern'.[4]

(222) $[_{ON}$ Der	Bund	der	Steuerzahler	in Hamburg]	forderte	
The	Union	of the	tax payers	in Hamburg	demanded	
derweil	$[_{OA}$ weitere	Einsparungen]	zur	Konsolidierung	des	
meanwhile	further	savings	for the	consolidation	of the	
Haushalts	in der Hansestadt.					
budget	in the Hanse town.					

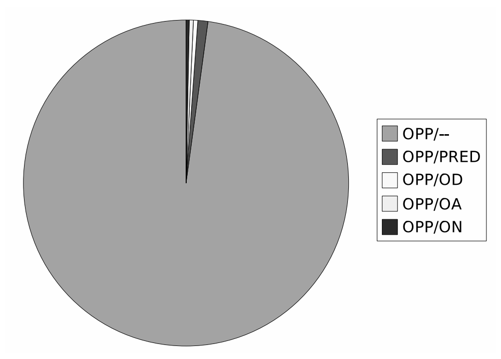
'Meanwhile, the tax payers union in Hamburg demanded further savings for the consolidation of the budget in the Hanse town.'

(223) $[_{ON}$ Er]	fordert	$[_{OA}$ seinen	Gegner]	$[_{OPP}$ zum	Duell].
He	challenges	his	rival	to a	duel.

'He challenges his rival to a duel.'

[3]The TüBa-D/Z has been annotated by at least two annotators and, in case of disagreement, the problesm were discussed.

[4]N.B.: Although 'fordern zu' just occurs with few nouns, it is no support verb since the verb 'fordern' has not lost its lexical content.



(a) ratio of errors

error type	number	percentage
OPP/–	708	97.93%
OPP/PRED	7	0.97%
OPP/OD	3	0.41%
OPP/OA	3	0.41%
OPP/ON	2	0.28%
	723	

(b) number and percentage of errors

Figure 11.15: Ratio of errors for OPP recall

Figure 11.15 shows the ratio of errors which occurred for OPP recall. The vast majority of errors (nearly 98%) occurred because the OPP was not annotated, at all. Again, the problem of the definition of prepositional objects plays a crucial role in the error analysis like for OPP precision, except that the case of recall shows the mirror image of the case of precision, i.e. prepositional objects recognized by the TüBa-D/Z but not contained in the IMSLex. However, sentence 224 gives an example in which the PC would most likely be regarded as an OPP by both approaches but is annotated as a prepositional object in TüBa-D/Z but not contained in the IMSLex and, thus, not annotated by our parser. The PC 'um das Thema' is selected by the verb 'ergänzen' and it cannot be replaced by a PC with any other preposition. Futhermore, an OPP can be assumed by method of exclusion: 'um das Thema' does not modify the preceding pronoun and it does not act as any adverbial expression.

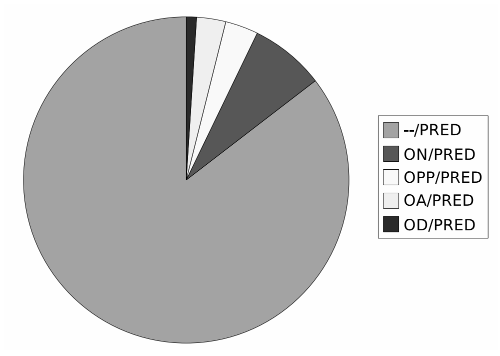
(224)

$[_{ON}$ Ich]	würde	$[_{OA}$ das]	$[_{OPP}$ um	das	Thema]	ergänzen,	wie	$[_{ON}$
I	would	that	with	the	issue	complete,	how	

man]	in	den	Stadtteilen	wirklich	$[_{OA}$ Kristallisationspunkte]
one	in	the	town quarters	really	focal points

schaffen	kann.
create	can.

'I would complete that with the issue how you could really create (cultural) centres in the different quarters in town.'



(a) ratio of errors

error type	number	percentage
–/PRED	181	85.38%
ON/PRED	16	7.55%
OPP/PRED	7	3.30%
OA/PRED	6	2.83%
OD/PRED	2	0.94%
	212	

(b) number and percentage of errors

Figure 11.16: Ratio of errors for PRED precision

The GF PRED is annotated with a precision of 78.53% and a recall of 72.87%, which yields an $F_{\beta=1}$ of 75.59. PRED has a special status among the GFs because it can be realized by an NC with the feature *nominative* or with the *Vergleichspartikel* (particle of comparison; KOKOM) 'als', a PC, an AJVC and an AVC. This makes finding the correct PRED among these various candidates a challenging task. The main error for PRED precision is that PREDs are annotated where there are no GFs, at all (85.38%). There are mainly two reasons for that. On the one hand, there are cases in which there is a PRED in the clause but the parser annotated the wrong chunk as the PRED. On the other hand, there are cases in which the parser annotated a PRED in clause in which none occurred. The latter case mainly concerns chunks with 'als'.

For the first case, sentence 225 gives an example: In it, the adjective chunk 'zu klein' is the PRED. However, since predicative adjectives (PoS-tagged ADJD) can also serve as adverbs, we have chunk-labelled them as AJVC for either adjective **or** adverb chunk. Hence, it is by no means clear whether 'zu klein' is the PRED. In the case at hand, our parser has annotated the PC 'dafür' at the beginning of the sentence as the PRED, and 'dafür' could, in fact be the PRED with different lexical material in the sentence. This example illustrates the special status of PREDs among the GFs. There are various candidates for the annotation of the PRED.

(225) Dafür wäre schon $[_{\text{ON}}$ die Schrift] $[_{\text{PRED}}$ zu klein].
For that were already the font too small.

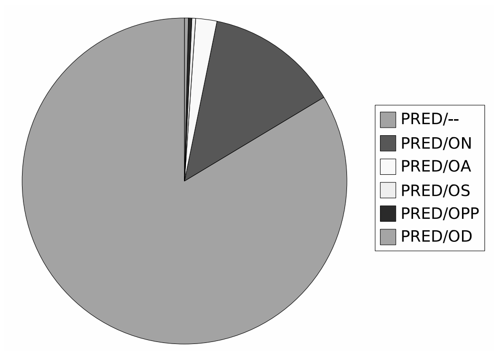
'The font would be too small for that, in the first place.'

Another peculiarity of PREDs is that they are just sub-categorized for by copula verbs like 'sein', 'werden', 'bleiben', 'scheinen' und 'heißen'. An exception to this is PRED with 'als'. This type of PRED can be sub-categorized for by various verbs as in 'agieren als' (*to act as*) or 'gelten als' (*to be regarded as*). Since, however, NCs with 'als' can also have other functions than PRED, their annotation is one of the main problems for PRED precision. Sentence 226 gives an example: The NC 'eine 60köpfige Versammlung' is the ON in the subordinate clause of the sentence and the NC 'als das schottische Parlament' is a post-modifier to the preceding NC. Since, however, the verb 'wählen' sub-categorizes for a PRED with 'als', it is annotated as the PRED. This reading would, in fact, be possible with the same syntactic structure but with different lexical material.

(226) In Wales, wo heute $[_{\text{ON}}$ eine 60köpfige Versammlung] mit weit
In Wales, where today a 60-member assembly with a lot
weniger Befugnissen als das schottische Parlament gewählt wird,
fewer competences than the Scottish parliament elected is,
hat $[_{\text{ON}}$ Labour] $[_{\text{OA}}$ weniger Sorgen].
has Labour less problems.

'In Wales where an assembly of 60 members with a lot fewer competences than the Scottish parliament is elected today, Labour has less problems.'

The second most frequent error for PRED precision which is also relatively prevalent is the confusion of ONs with PREDs (7.55%). The main reason for this error is that ON and PRED have the same case features. Although the ON typically precedes the PRED, there are still errors possible as the case of sentence 227 shows: There are two NCs which contain the feature *nominative* in the sentence: 'es' and 'ein vierter Anbieter'. Hence, our parser annotated the first as the ON and the second as the PRED. However, 'es' is a modifier to the ON (ON-MOD) and the PRED is realized by the PC 'am Markt'. This error shows, again, the problems occurring because of the fact that PREDs are realized by various constituents.



(a) ratio of errors

error type	number	percentage
PRED/–	243	83.51%
PRED/ON	39	13.40%
PRED/OA	6	2.06%
PRED/OS	1	0.34%
PRED/OPP	1	0.34%
PRED/OD	1	0.34%
	291	

(b) number and percentage of errors

Figure 11.17: Ratio of errors for PRED recall

(227)

Denn	[$_{\text{ON}-\text{MOD}}$ es]	ist	[$_{\text{ON}}$ ein	vierter	Anbieter]	[$_{\text{PRED}}$ am	Markt].
For	it	is	a	fourth	supplier	at the	market.

‘For there is a fourth supplier at the market.’

Figure 11.17 shows the ratio of the errors for PRED recall. The most frequent error is that PREDs are not annotated, at all (83.51%). The main reason for this is that no adequate candidate for the annotation of PRED could be found. This is different from other GFs because in their cases, the GF might just have been missing in the SC frame of the relevant verb. For PREDs, however, there are just the few copula verbs. Thus, this problem does not occur. Sentence 228 gives an example of a case where no adequate candidate was found. A PRED typically occurs to the end of the sentence. It may be realized by various constituents. In the case at hand, the final NC in the clause does not have the feature *nominative* and the PC ‘auf dem Stand’ which actually is the PRED does not occur to the end of the sentence. Thus, our parser failed to assign this PRED. There are many combinations like this in which we rather do not annotate a PRED than risking to spoil the recall of PRED.

(228)

[$_{\text{ON}}$ Er]	ist	[$_{\text{PRED}}$ auf	dem	Stand]	eines	Kleinkindes,	sagt	[$_{\text{ON}}$ sein	Mutter].
He	is	on	the	level	of a	toddler,	says	his	mother.

‘He is on the developmental level of a toddler says his mother.’

The second most frequent error is that PREDs are confused with ONs (13.40%). The reason for this is that both ON and PRED are marked by the feature *nominative*. Typically ON precedes PRED. However, if the first potential ON with nominative case does not agree in number with the verb, the second potential ON ist taken if it agrees in number with the verb. Still, for our parser, it is, in some cases, not possible to unambiguously determine the feature *number* of an NC. This is especially true of coordinated NCs. In the NC 'der SPD-Vorsitzende und Bundeskanzler Schröder' it is not clear whether there are one or two persons. Hence, for coordinated NCs, we assume for our parser that they are either singular or plural (except if all constituents are plural). The problems caused by this decision can be seen in sentence 229. Since the second NC in the second clause is plural and the verb is supposedly plural, too,[5] the second NC is annotated as the PRED and the first as ON. Our parser, however, annotates the first as ON and the second as PRED.

(229)

Doch	$[_{\text{OA}}$ alle	fünf	Altrocker]	plagt	$[_{\text{ON}}$ ein	existenzielles
But	all	five	senior rockers	afflict	an	existential

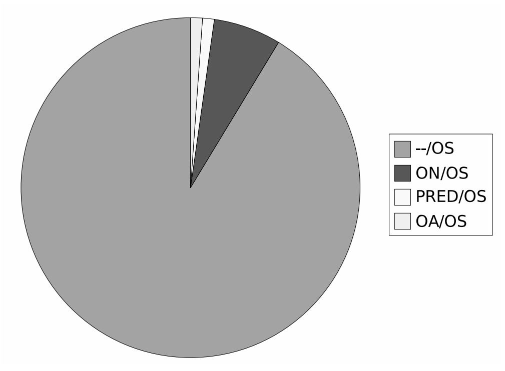
Zipperlein],	sein	$[_{\text{PRED}}$ es]	$[_{\text{ON}}$ der	Suff,	Schulden,	eine	kaputte
gout,	be	it	the	boozing,	debts,	a	broken

Ehe	oder	die	(inzwischen	historische)	große	Liebe].
marriage	or	the	(meanwhile	historical)	great	love.

'But all five senior rockers are afflicted by an existential gout, be it boozing, debts, a marriage in pieces or the (meanwhile historical) great love.'

The GF OS is annotated with a precision of 76.92% and a recall of 60.67%. This yields an $F_{\beta=1}$ of 67.83. 91.40% of the cases of precision errors are caused by annotating clauses which did not have any GF label, at all, and 6.45% by annotating clauses which actually were labelled ON. For the first case of precision errors, there were mainly two reasons: On the one hand, there are errors caused by an annotational problem in the TüBa-D/Z and, on the other hand, there are modifying clauses wrongly annotated as OS. Sentence 230 gives an example of the problem which occurs if parts of a subordinate clause both precede and follow the matrix clause. In sentence 230, 'erklärte Runde' is the matrix clause and the sentential object of this clause surrounds the matrix clause. Thus, it is impossible to annotate the subor-

[5]In fact, the verb is misspelled and the annotators in the TüBa-D/Z had to determine whether the author intended to use the singular (i.e. 'sei') or the plural reading (i.e. 'seien') of the verb. They decided for the latter.



(a) ratio of errors

error type	number	percentage
–/OS	85	91.40%
ON/OS	6	6.45%
PRED/OS	1	1.08%
OA/OS	1	1.08%
	93	

(b) number and percentage of errors

Figure 11.18: Ratio of errors for OS precision

dinate clause as the OS of the matrix clause without crossing edges (which are not allowed in TüBa-D/Z) as can be seen in figure 11.19. In our annotation system, we cannot annotate surrounding subordinate clauses, either. However, since we also cannot determine whether there is a surrounding OS clause,[6] we annotate one part of it, which yields an OS precision error.

(230)

Wenn	$[_{\text{ON}}$ man]	so	hochwertig	verkaufen	will,	erklärte	$[_{\text{ON}}$ Runde],
If	one	so	high-value	sell	wants,	declared	Runde,

dann	muss	$[_{\text{ON}}$ man]	$[_{\text{OA}}$ Unverträglichkeiten]	ausschalten.
then	must	one	incompatibilties	eliminate.

‘If you want to sell at a high price, declared Runde, you have to eliminate incompatibilities.’

(231)

$[_{\text{ON}}$ $[_{\text{ON}}$ Wer]	$[_{\text{OA}}$ sich]	so	rasch	$[_{\text{OPP}}$ von	seinen	pazifistischen
Who	himself	so	quickly	of	his	pacifistic

Positionen]	verabschiedet	und	statt dessen	$[_{\text{OD}}$ der scheinbaren
positions	say good-bye	and	instead of that	the imaginary

Kriegslogik]	$[_{\text{OA}}$ das Wort]	redet,]	sollte	$[_{\text{OA}}$ sich]	$[_{\text{OPP}}$ davor]
war logic	the word	talks,	should	himself	of that

hüten,	$[_{\text{OPP-MOD}}$ $[_{\text{OD}}$ anderen]	$[_{\text{OA}}$ nationalistische	Motive	und
beware,	others	nationalistic	motives	and

[6]In sentence 230, this would mean that we had to determine whether the *wenn*-clause is a modifier to the matrix clause or the OS clause, which we cannot.

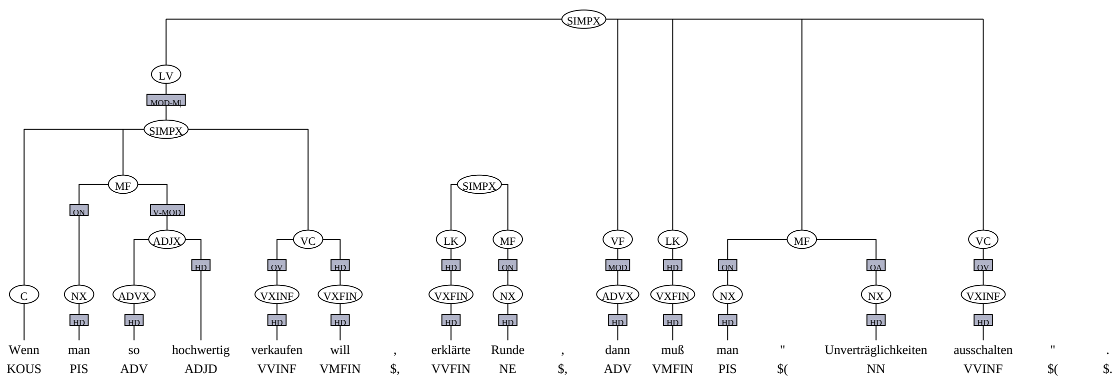

SIMPX
LV
MOD-MI
SIMPX
MF
ON
V-MOD
ADJX
HD
VC
OV
HD
SIMPX
LK
MF
HD
ON
VF
LK
MF
VC
MOD
HD
ON
OA
OV
C
NX
ADVX
VXINF
VXFIN
VXFIN
NX
ADVX
VXFIN
NX
NX
VXINF
HD
Wenn man so hochwertig verkaufen will , erklärte Runde , dann muß man " Unverträglichkeiten ausschalten " .
KOUS PIS ADV ADJD VVINF VMFIN $, VVFIN NE $, ADV VMFIN PIS $(NN VVINF $($.

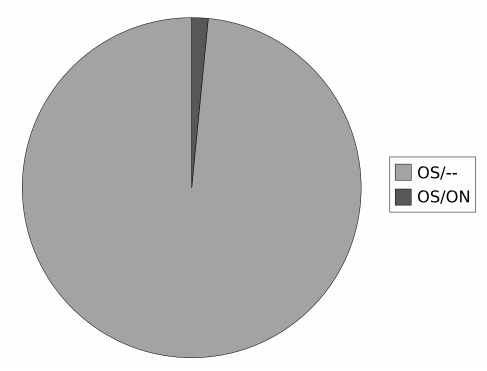
Figure 11.19: Subordinate Clause surrounding matrix clause in TüBa-D/Z

Gedanken] zu unterstellen].
thoughts to allege.

'Whoever gets rid of his pacifistic positions so quickly and talks in favour of the imaginary logic of war should beware of alleging that others have nationalistic motives and thoughts.'

Another problem is to distinguish between OS clauses and modifying clauses. In sentence 231, the infinitive clause 'anderen nationalistische Motive und Gedanken zu unterstellen' is a modifier to the OPP 'davor'. Thus, the verb is 'sich davor hüten zu'. However, this reading is missing in the SC frame list of the verb 'hüten', we annotated the other reading 'sich hüten zu'. This yielded an OS precision error. The second most frequent OS precision error (6.45%) is annotating ON clauses as OS clauses. Sentence 232 gives an example: The daß-clause is the ON of the sentence, and the 'es' in the matrix clause is the modifier of the ON (ONMOD). Since this reading is, however, not coded in the SC frame of the verb in the lexicon, the annotation of the correct reading has failed and the parser has matched the structure into the incorrect reading of an existing SC frame with ON and OS.

(232) [$_{\text{ONMOD}}$ Es] ist nicht auszuschließen, [$_{\text{ON}}$ daß er seinen trüben
It is not to be ruled out that he his cheerless
Obskurantismus auch in seine Unterrichtseinheiten einfließen
obscurantism also into his teaching units incorporate



(a) ratio of errors

error type	number	percentage
OS/–	198	98.51%
OS/ON	3	1.49%
	201	

(b) number and percentage of errors

Figure 11.20: Ratio of errors for OS recall

läßt].
lets.

'It cannot be ruled out that he incorporates his cheerless obscurantism into his lessons, as well.'

As regards recall errors for OSs, most of the error were caused by not annotating the respective sentence, at all (98.51%). Only in 1.49% of the cases, wrong annotation lead to an OS recall error (cf. figure 11.20). The main reason for these recall errors were missing SC frames. The main cause for missing SC frames can be seen in the newspaper style in the corpus which allows many verbs to sub-categorize for an OS which would typically not sub-categorize for an OS. Sentence 233 gives an example: In the sentence, three V2 clauses are coordinated asyndetically and this coordination acts as the OS of the matrix clause. While this structure does not cause major problems to the parser, the lexical verb of the matrix clause 'rechnen' does not have the SC frame for an ON and an OS in its SC list; and, in fact, the use of 'rechnen' with these GFs is unlikely outside newspaper style. The only way to handle structures like these is to deal with them in the non-lexical component. This approach did, however, not succeed.

(233) [$_{\text{OS}}$ Die staatliche Bremer Investionsgesellschaft (BIG) soll das
The state-run Bremian investment society (BIG) should the
Projektmanagement übernehmen, dafür bekommt sie ein Prozent
project management take over, for that receives it one percent

	precision	recall	$F_{\beta=1}$
overall	83.49%	75.40%	79.24
OA	80.85%	77.09%	78.92
OD	81.73%	57.08%	67.21
OG	100.00%	7.69%	14.29
ON	88.62%	87.84%	88.23
OPP	71.66%	44.63%	55.00
OS	72.96%	55.97%	63.34
PRED	75.70%	62.72%	68.60

Table 11.3: Final results of evaluation of GF annotation with TnT PoS tags

des	"geschätzten	Auftragsvolumens",	das	seien	2,58 Millionen
of the	"estimated	size of the order",	this	be	2.58 million

Mark,]	rechnet	[$_{\text{ON}}$ das Wirtschaftsressort].
Marks,	calculates	the economics department.

'The calculations of the department of economics are as follows: The state-run investment society of Bremen (BIG) is supposed to take over the project, it will receive one percent of the "estimated size of the order" for this, this would be 2.58 million Marks.'

Table 11.3 shows the final results of the evaluation of the GF parser on TnT PoS tags.[7] The results are illustrated in figure 11.21. Overall precision is 83.49% (gold PoS tags 85.52%; –2.03 percentage points) and overall recall 75.40% (gold PoS tags 80.02%; –4.62 percentage points). This yields an $F_{\beta=1}$ of 79.24 (gold PoS tags 82.68; –3.44 points). These results show that the GF parser works robustly because the PoS tag quality has dropped by 4.28 percentage points. The decrease in precision is very moderate if compared to the decrease of the tagging accuracy. The decrease in recall shows that the tagging accuracy rather affects recall than precision (cf. tables 11.22 and 11.23), which means that, due to the tagging error, there was no longer a parsable sequence which the parser could match. The decrease in performance for TnT tags is distributed quite evenly both for precision and recall. The only exception is the drop of recall for PREDs (– 10.15 percentage points) which is due to the 14.62 drop of $F_{\beta=1}$ on TnT tags for adjective chunks which are

[7]The PoS tag annotation with TnT for the test set is described is section 10.2.

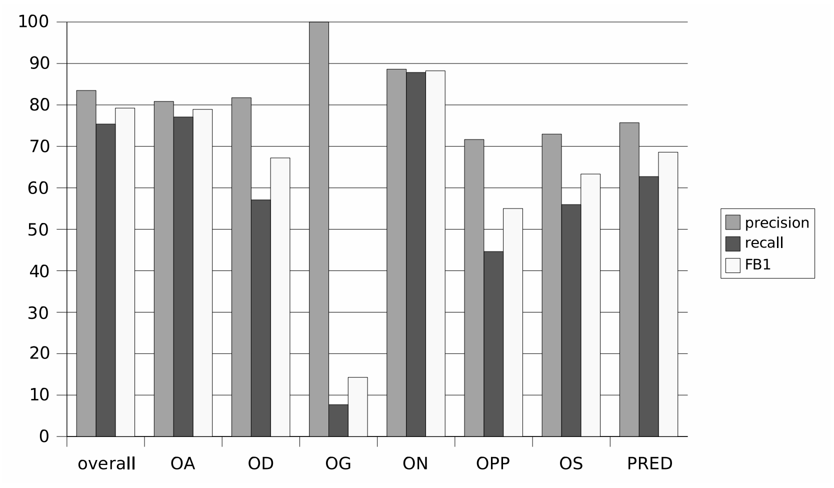


Figure 11.21: Final results of evaluation of GF annotation with TnT tags

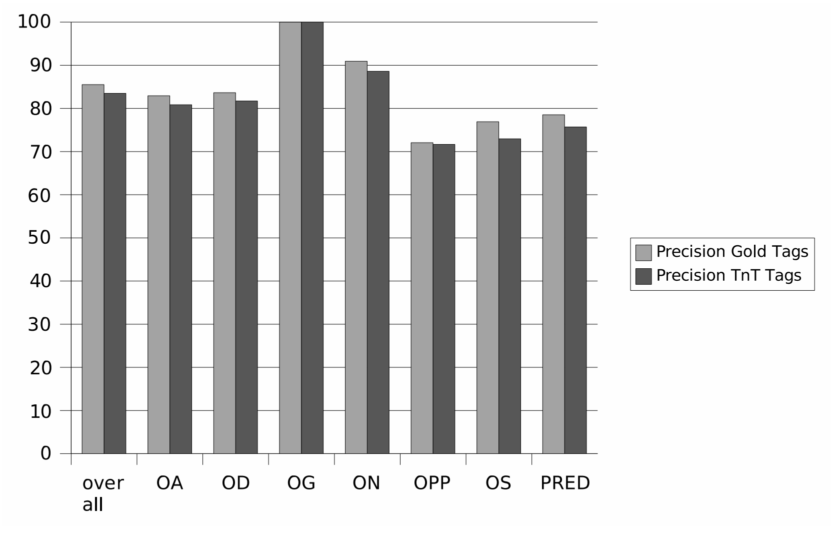


Figure 11.22: Impact of PoS tag quality on precision

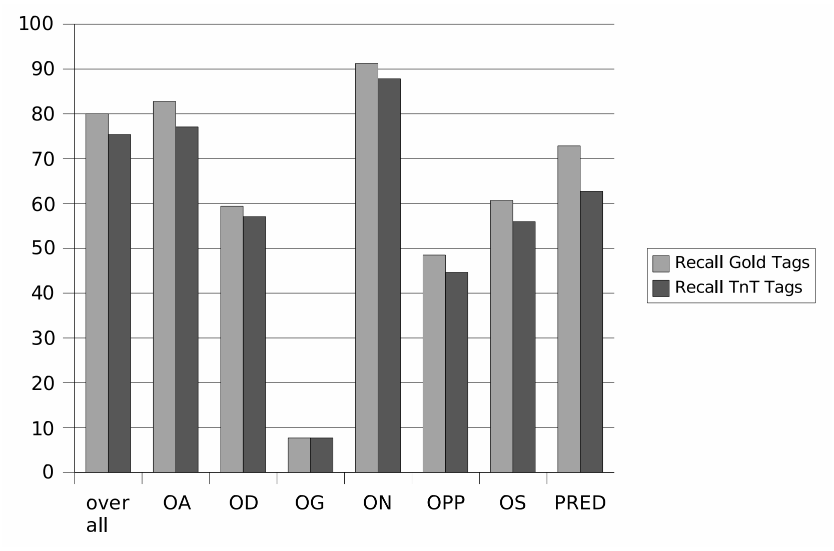


Figure 11.23: Impact of PoS tag quality on recall

in part the target chunks for PREDs.

11.4 Impact of Different Components on Annotation

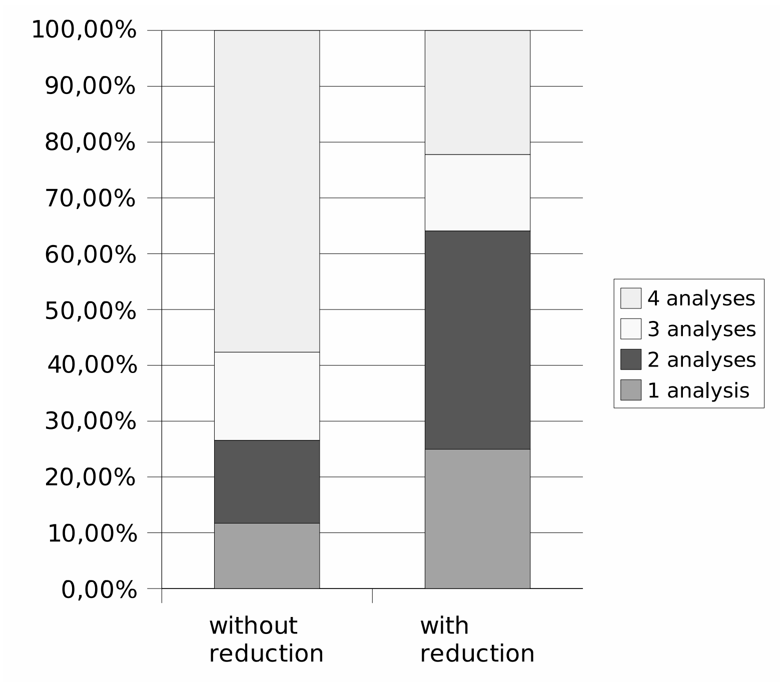
This section points out how much a certain technique or knowledge source contributes to the final evaluation results discussed in section 11.3. That way, the importance of a certain technique or knowledge source either for overall success or for a specific GF can be made clear. Section 11.4.1 presents the results of the component reducing the number of readings in the morphological ambiguity class of a chunk, and section 11.4.2 shows how this reduction affects the results of the annotation of GFs.

11.4.1 Evaluating Morphological Ambiguity Reduction

The first component which is applied after the shallow parsing component is the morphological ambiguity class reduction component described in chapter 8. On the basis of the shallow parsing structure and evidence about the constraints of morphological features occurring, e.g., within one chunk, it reduces the number of morphological features in chunks. Thereby, it reduces the potential GFs which a chunk can have since nominal GFs (i.e. GFs realized by a noun chunk) are closely connected with case features. Figure 11.24 shows the ratio and number of case features occurring in the morphological ambiguity class of a chunk after the morphological annotation by the tool DMOR and in how far they are reduced after application of the morphological ambiguity class reduction component. Case features before reduction were calculated by taking into account the ambiguity classes of all tokens in a chunk. Case features after reduction were calculated by taking into account the reduced morphological ambiguity class assigned to the respective chunk. Only *maximal chunks* (i.e. chunks not contained in any other chunk) were taken into account because only they can serve as GFs. NCs contained in PCs are, thus, not included in the count since, for the annotation of OPPs, only the feature *preposition* is used.

The maximal ambiguity of a chunk is that it has the case features of all four cases. In the ideal case, a chunk has just one case feature. In this case, the GF can almost unambiguously be assigned (except for genitives).[8] Figure 11.24 shows that before the application of the reduction component, the majority of tokens, i.e. 57.60%, has four analyses, i.e. it is completely ambiguous. With the help of the reduction component it is possible to reduce this proportion by 35.39 percentage points to 22.24% of all chunks. On the other end of the scale, it is possible to raise the share of the chunks which are unambiguous from 11.70% to 24.95%. This is even more important if it is taken into account that one unambiguous potential GF can also disambiguate another one without any assumptions about the linear order of GFs. If, e.g., there are two potential GFs in a clause with a verb which sub-categorizes for an ON and an OA, and one of them is unambiguously nominative and the other one has the potential to be either nominative or accusative, then the fact that one of the two is unambiguously nominative reveals that the other one is unambiguously accusative as the simple sentence 234 shows.

[8]Only few NCs do not have any morphological case features since we assign default ambiguity classes. However, reflexive pronouns (STTS tag: PRF) and cardinal numbers (STTS tag: CARD) are not assigned case features by DMOR. We do not assign any default morphological ambiguity class to PRFs since their respective GFs are assigned on the basis of their form. Cardinals only receive a default ambiguity class when they are **not** dates, telephone numbers, etc.



(a) ratio of different analyses in test section

	without reduction		with reduction		difference in percentage points
1 analysis	1446	11.70%	3084	24.95%	+13.25
2 analyses	1836	14.86%	4834	39.11%	+24.25
3 analyses	1954	15.81%	1692	13.69%	–02.12
4 analyses	7123	57.63%	2749	22.24%	–35.39

(b) total numbers and percentage of different analyses in test section

Figure 11.24: Effect of morphological ambiguity class reduction

(234)	$[_{\text{NC}}$ Die Tochter]{nom,acc}	liebt	$[_{\text{NC}}$ der Vater]{nom}.
	The daughter	loves	the father.

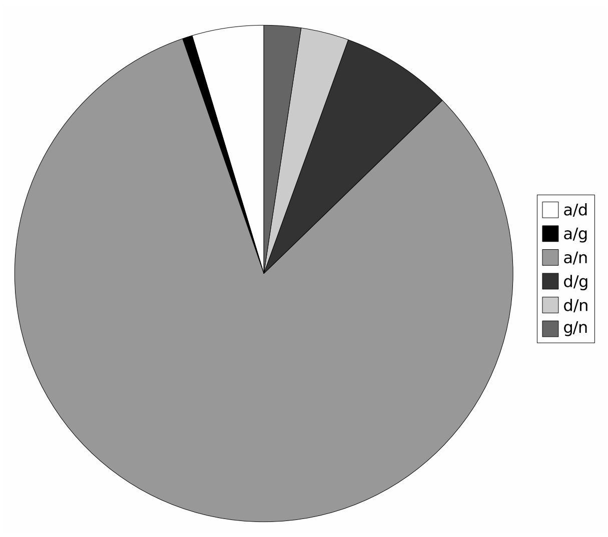
'The father loves the daughter.'

Figure 11.24 furthermore shows that there has been a significant rise in the number of chunks which have only two analyses left (+24.25 percentage points) and that the number of chunks which still have three analyses has been slightly reduced (–2.12 percentage points). Especially the rise of the number of chunks which just have two analyses left could suggest, on the first sight, that annotation of GFs could now be considerably easier since the annotation also relies on SC frame information of the verbs. However, if one takes a closer look at the distribution of the case features among the chunks with two analyses as shown in figure 11.25, it becomes clear that there are still problems.

As can be seen in figure 11.25, among the ambiguity classes which comprise two case features, the ambiguity class which covers the vast majority of chunks is the one with the case features *accusative* and *nominative* (82.11%). Since, however, these two case features are strongly connected with the three most frequent nominal GFs ON, OA and PRED which together comprise about 95.26% of all nominal GFs (cf. figure 11.26[9]), this class does not contribute as much to the disambiguation of GFs as it appears on the first sight. On the whole, 87.46% of the chunks with more than one analysis have both nominative and accusative in its ambiguity class. But, although the reduction might not lead to a large overall improvement, one can predict that it will be important for the lower frequency GFs OG and OD. Furthermore, it has to be pointed out that the fact that a chunk is nominative or accusative also prevents it from being regarded as a genitive modifier which is not counted among the GFs, at all.

On the whole, one can say that, although the figures presented in this section give evidence about the effectiveness of the morphological ambiguity class reduction component itself, they do not straightforwardly show how these results will affect the annotation of GFs by the following GF annotation component. In order to check this, we show how the performance of the morphological ambiguity class reduction component effects the annotation of GFs in section 11.4.2.

[9]Only the nominal realizations of the GFs ON, OA and PRED were counted in figure 11.26.

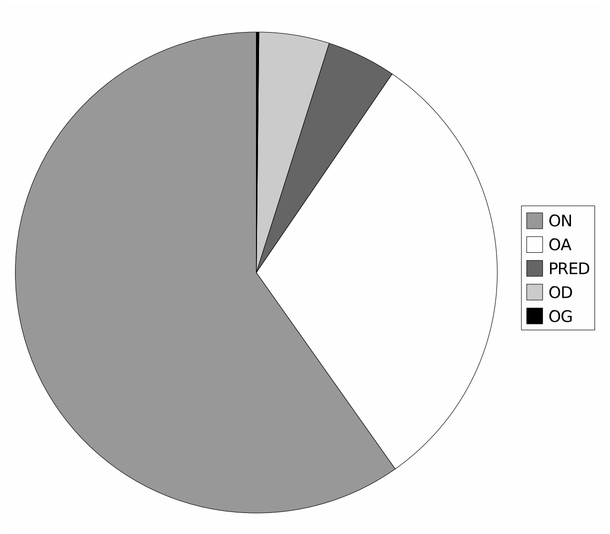


(a) ratio of two-case-features combinations

case features	total number	percentage
a/d	220	4.55%
a/g	31	0.64%
a/n	3969	82.11%
d/g	354	7.32%
d/n	149	3.08%
g/n	111	2.30%
	4834	

(b) total numbers and percentage of two-case-features combinations

Figure 11.25: Ratio and numbers of two-case-features combinations



(a) ratio of nominal GFs in the test set

GF	total number	percentage
ON	5507	59.74%
OA	2839	30.80%
PRED	435	4.72%
OD	425	4.61%
OG	13	0.14%
	9219	

(b) total numbers and percentage of nominal GFs in the test set

Figure 11.26: Ratio and numbers of nominal GFs in the test set

	precision	recall	$F_{\beta=1}$
overall	85.52%	80.02%	82.68
OA	82.94%	82.77%	82.86
OD	83.66%	59.40%	69.47
OG	100.00%	7.69%	14.29
ON	90.93%	91.27%	91.10
OPP	72.06%	48.50%	57.98
OS	76.92%	60.67%	67.83
PRED	78.53%	72.87%	75.59

Table 11.4: GF annotation with morphological ambiguity reduction

11.4.2 Impact of Morphological Ambiguity Class Reduction

In order to explore the influence of the morphological ambiguity class reduction component on the annotation of GFs, we annotated the test set once with this component and once without it. In the case in which we annotated the test set without the morphological ambiguity class reduction component, we took all the morphological ambiguity classes of all tokens in a chunk as the ambiguity class of this chunk. Since it is also of interest in how far morphology contributes to the annotation of GFs at all, we also annotated the test set with just default morphological ambiguity classes, i.e. we assigned all four cases to all relevant chunks.

Figure 11.27 illustrates the effect of the morphological ambiguity class reduction component on the precision of GF annotation and figure 11.28 shows the same effect for recall. The exact numbers are presented in tables 11.4–11.6. Figure 11.27 shows that, as expected, the precision of the annotation of nominal GFs falls when the morphological ambiguity class reduction component is not applied. This is even more the case when no morphological annotation is applied, at all. For ONs, precision falls from 90.93% to 88.88% without reduction component and to 84.72% without any morphological annotation. For OAs, the fall is higher from 82.94% to 76.79% to 71.14%. Although this decrease is significant, it is not dramatic. This supports the assumption that there is an unmarked linear order of GFs because, if morphology is not present, the parser solely relies on the assumption of unmarked GF orders.

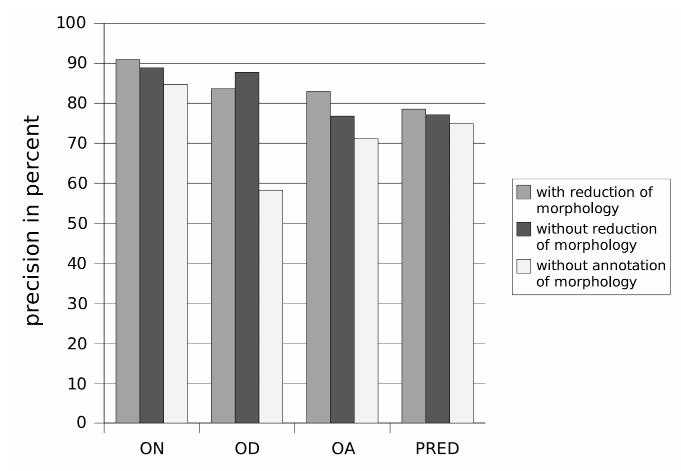


Figure 11.27: Effect on precision

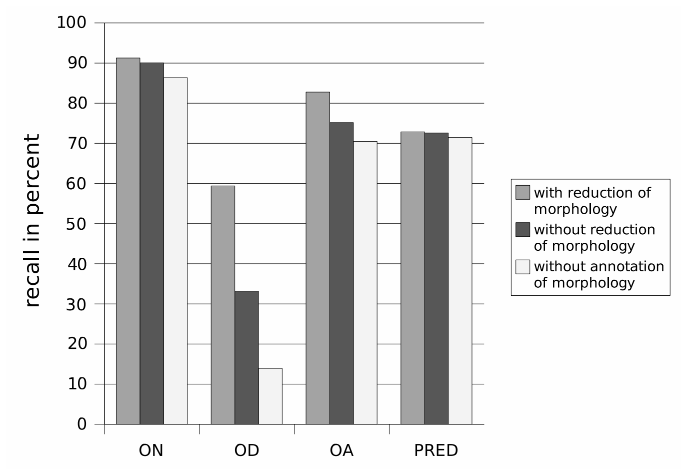


Figure 11.28: Effect on recall

	precision	recall	$F_{\beta=1}$
overall	82.86%	76.79%	79.71
OA	76.79%	75.17%	75.97
OD	87.73%	33.18%	48.15
OG	0.00%	0.00%	0.00
ON	88.88%	90.06%	89.47
OPP	71.72%	49.50%	58.58
OS	76.92%	60.67%	67.83
PRED	77.14%	72.59%	74.80

Table 11.5: GF annotation without morphological ambiguity reduction

	precision	recall	$F_{\beta=1}$
overall	78.65%	73.13%	75.79
OA	71.14%	70.48%	70.81
OD	58.25%	13.92%	22.47
OG	0.00%	0.00%	0.00
ON	84.72%	86.38%	85.54
OPP	70.76%	49.29%	58.10
OS	76.85%	61.06%	68.05
PRED	74.90%	71.47%	73.15

Table 11.6: GF annotation with default ambiguity classes

diesen	{dpm, dpf, dpn, asm}	asm
Schluß	{nsm, dsm, asm}	
ein	{nsm, nsn, asn}	nsm
Bundeswehr-Arzt	{nsm, dsm, asm}	

Table 11.7: Example of morphological ambiguity class reduction

Figure 11.28 shows that the effect on recall is similar to that on precision. However, as regards the difference in decrease as compared between the different GFs, a general tendency becomes obvious. For those GFs which are (mainly) selected by lexical verbs, i.e. ON, OD and OA, the decrease is generally the higher the lesser they occur. The slight increase in precision for OD when annotated without reduction component (+4.07%) is accompanied by a dramatic decrease in recall (-26.22%). Thus, one can say that, generally, morphological information is most important for those GFs which occur rarer. This shows how important morphological information is for annotation, since the parser is intended to annotate corpora for linguistic research and for this purpose it is important that all linguistic phenomena including the less frequent ones are annotated sufficiently.

Sentence 235 shows an example from the test corpus in which the morphological ambiguity class reduction component disambiguates two NCs, which are targets for GF annotation, such that a wrong annotation is avoided. Both NCs have nominative, dative and accusative in its ambiguity class (as can be seen in table 11.7). After the morphological ambiguity class reduction, both NCs are fully disambiguated such that the annotation of the unmarked GF order, which would require the ON to precede the OA, is avoided and the correct annotation of 'diesen Schluß' as the OA and 'ein Bundeswehr-Arzt' as an ON is enforced.

(235)	$[_{\text{OA}}$ Diesen	Schluß]	zog	am	Mittwoch	$[_{\text{ON}}$ ein
	This	conclusion	drew	on	Wednesday	a

Bundeswehr-Arzt]	im	makedonischen	Camp Cegrane.
German Army doctor	in the	Macedonian	Camp Cegrane.

'A medical officer of the German Army drew this conclusion in the Macedonian Camp Cegrane on Wednesday.'

11.4.3 Impact of non-lexical Component

The GF annotation component crucially relies on SC frame information. Each verb in the corpus is assigned its SC frame from the IMSLex. However, there are, inevitably, some verbs which are not assigned SC frames or which are not assigned the SC frame which they sub-categorize for in the corpus. There are various reasons for this: One the one hand, a SC frame lexicon is not exhaustive. Thus, it does not cover the whole vocabulary of a language. From the test corpus, verbs like 'aufpeppen' (to spice sth. up), 'residieren' (to reside) or 'zerbröseln' (to crumble) were not contained in the IMSLex. On the other hand, language is creative such that a writer might create or use a neologism like 'e-melden' (to get in touch via e-mail), which occured in the test corpus.

Furthermore, there are spelling errors like 'zusamenstellen' instead of 'zusammenstellen' (to compile) or 'beachtc' instead of 'beachte' (to consider), and unexpected input due to the electronic processing of the text. For instance, German 'ck' is hyphenated as 'k-k'. This lead to verbs like 'pikken' or 'drükken' instead of 'picken' (to pick) or 'drücken' (to push). We did not change this unexpected input because it is already contained in the orginal corpus. A rarer event are dialectal words in the text like the Swabian 'koscht' for 'kostet' (costs).

In our test section, there are 4,951 lexical verb tokens. On the whole, the verbs which were not assigned a SC frame amounted to just 3.9% of all tokens (193) and to 10.6% of all verb types (157 of 1,475). The verb 'sitzen' (to sit) was by far the most frequent verb without SC frame. It occurred 17 times. The next most frequent verbs occurred 3 times. From the 3.9% of tokens without SC frame, 3% were lemmatized by the tool DMOR but could not be assigned a SC frame, i.e. they were not contained in the IMSLex (cf. the list in appendix D.1). 0.9% of the tokens could not be lemmatized by DMOR and could, thus, also not be assigned a SC frame (cf. the list in appendix D.2). It is important to consider that the non-lexical component does not just have to deal with the 3.9% of verbs which are not assigned any SC frame but also with a certain amount of verbs which are not assigned the SC frame with which they are realized in the test section. Thus, the amount of verbs to be dealt with is higher than 3.9%.

The non-lexical component, which does not use any SC information, considerably enhances the performance of the system as table 11.8 and table 11.9 show. While overall precision just decreases by 2.15%, overall recall rises by

	precision	recall	$F_{\beta=1}$
overall	85.52%	80.02%	82.68
OA	82.94%	82.77%	82.86
OD	83.66%	59.40%	69.47
OG	100.00%	7.69%	14.29
ON	90.93%	91.27%	91.10
OPP	72.06%	48.50%	57.98
OS	76.92%	60.67%	67.83
PRED	78.53%	72.87%	75.59

Table 11.8: Results of evaluation with non-lexical component

10.06%. Consequently, $F_{\beta=1}$ rises by 4.86. These figures show the usefulness of the non-lexical component. Since the non-lexical component uses less information than the lexical component, i.e. it works without SC frame information, there remains the question why there should be a lexical component, at all. Table 11.10, which shows the results of the application of **just** the non-lexical component, gives the answer to this question: First of all, the non-lexical component does not annotate all GFs. OSs are not dealt with and OPPs are virtually not dealt with.[10] And, secondly, the figures in table 11.10 show that the non-lexical component has an acceptably high precision if compared to the lexical component in table 11.9 (cf. figure 11.29 for an illustration) but a considerably low recall except for ONs and ODs (cf. figure 11.30 for an illustration). Still, it is interesting to note that the non-lexical component has such a good performance with just 41 rules, which is about 0.5% of all rules.

The reason for the bias towards precision is that the non-lexical component annotates GFs quite conservatively, i.e. it is tuned to the preceding lexical component. Unlike the lexical component, it only annotates GFs if there is clear morphological indication for them, since there is no indication of the verb as to which GFs may occur. Only as regards ONs, this information suffices as the evaluation results of 89.11%/89.70% (precision/recall) show. This is because our assumption that nearly every sentence needs an

[10]There is a heuristic in the non-lexical component which annotates all PCs containing ‘sich’ as OPPs, which has a precision of 69.23%, which is slightly lower than the 72.06% overall precision for OPPs. This heuristic raises $F_{\beta=1}$ by 0.71.

	precision	recall	$F_{\beta=1}$
overall	87.67%	69.96%	77.82
OA	85.65%	72.30%	78.41
OD	83.70%	35.73%	50.08
OG	100.00%	7.69%	14.29
ON	93.17%	79.02%	85.52
OPP	72.23%	47.44%	57.27
OS	76.92%	60.67%	67.83
PRED	84.61%	64.27%	73.05

Table 11.9: Results of evaluation without non-lexical component

	precision	recall	$F_{\beta=1}$
overall	86.66%	62.75%	72.79
OA	86.11%	58.65%	69.78
OD	90.50%	50.81%	65.08
OG	0.00%	0.00%	0.00
ON	89.11%	89.70%	89.41
OPP	69.23%	1.28%	2.52
OS	0.00%	0.00%	0.00
PRED	68.18%	46.49%	55.28

Table 11.10: Results of evaluation of only non-lexical component

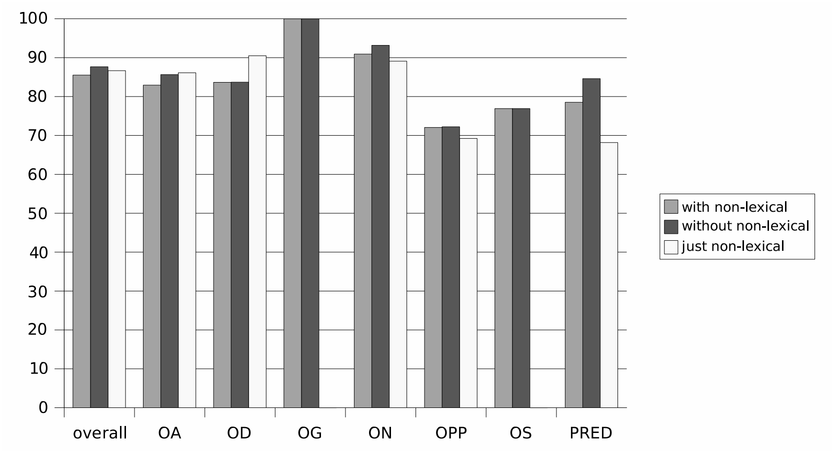


Figure 11.29: Effect on precision

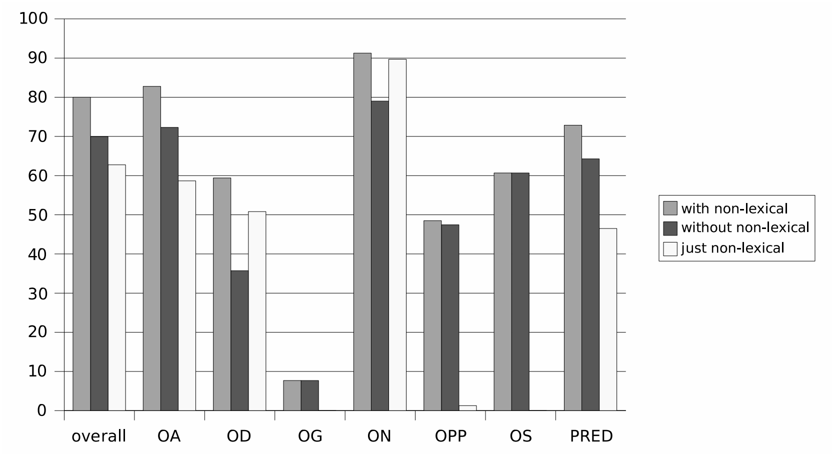


Figure 11.30: Effect on recall

ON seems to have been correct. As regards the other GFs, however, this assumption was not made and it would have been incorrect. While nearly every verb sub-categorizes for an ON, sub-categorization of other GFs is far more restricted, i.e. it depends far more on the verb in question. This is, however, precisely the information we are lacking in the non-lexical component. Thus, we have to rely on morphological information and can only then annotate a GF if this morphological information is very secure. This leads to figures with quite high precision but with lower recall (as compared to the lexical component). If we tried to increase the recall of the system by relaxing the constraints on morphology, we would lower the overall precision of the system. Our intention was, however, to increase recall without considerably reducing precision.

Figure 11.30 also illustrates the special case of ODs: Except for the case of ONs in which we assumed that every verb sub-categorizes for them, only ODs have a higher recall if **only** the non-lexical component is applied compared to the case that **only** the lexical component is applied. The main reason for this is that, in the TüBa-D/Z, all datives are annotated as ODs (including free datives)[11] while, in the IMSLex, only sub-categorizable datives are listed. Since free datives occur rarely, we had to apply a very restricted approach to annotating non-sub-categorizable ODs. Thus, althouh the non-lexical component considerably enhances recall for ODs (by 23.67%), the recall still remains comparatively low (59.40%).

The results of the non-lexical component as shown in table 11.10 are significant for the assessment of our system because they show the importance of the lexical component, which uses a large lexicon to annotate GFs. On the one hand, quite high precision for some GFs is possible just using morphological information and information about linear order but, on the other hand, the SC information from the lexicon is absolutely essential in order to achieve adequate precision **and** recall for **all** GFs. Although the recall of the non-lexical component alone is higher in some cases than the recall of the lexical component alone, the combination of both components excels the recall of both individual components in the case of all GFs (cf. figure 11.30). Thus, both components are indispensable for the parser. On the one hand, some GFs cannot be annotated without using lexical information (OS and OPP, but also to a large extent PRED and OA) but, on the other hand, all

[11]Section 7.3 gives the details of the problem of sub-categorizable and non-sub-categorizable datives.

GFs profit from the addition of the non-lexical component to the parser.

Sentences 236–238 show some examples in which the non-lexical component annotated GFs which the lexical component was not able to annotate. In sentence 236, DMOR was able to analyze the verb ‘vorzubreiten’ since the verbal particle ‘vor’ (before) and the verb ‘breiten’ (to spread) exist. However, the combination of the two does not exist and, thus, a SC frame was not assigned. In fact, the token simply is a spelling mistake because the author actually wanted to use the token ‘vorzubereiten’ (to prepare). The chunk ‘die neue Aufgabe’ is morphologically ambiguous. It can be either ON or OA. Since, however, ONs cannot occur in infinitive clauses, the chunk is correctly assigned the GF OA. The fact that the non-lexical component also works as a backup in cases of spelling mistakes illustrates its importance for the robustness of the whole parser.

(236)

Worauf	[$_{\text{ON}}$ Rangnick]	zurücktrat,	um	in aller	Ruhe
Whereupon	Rangnick	resigned,	in order to	in all	quietness

[$_{\text{OA}}$ die	neue	Aufgabe]	in	Stuttgart	vorzubreiten.
the	new	challenge	in	Stuttgart	prepare.

‘Whereupon Rangnick resigned in order to prepare his new challenge in Stuttgart without ruffle.[12]’

Sentence 237 shows are different problem: The auther puts a part of the verbs in parentheses in order to evoke the same effect the verb has in the English translation, i.e. that there actually two verbs. Since this structure could not be morphologically analyzed, it did not receive any SC frame, either. Still, the parser was able to assign the correct GFs because the non-lexical component worked as a backup: the chunk ‘eine neotribale Struktur’ which could be both ON or OA is assigned the GF ON since there is no other chunk in the sentence which has the feature *nominative*. The reflexive ‘sich’ may be either an OD or an OA. Since the case that ‘sich’ is an OA is more frequent than the case that it is an OD, it is assigned the GF OA provided that there is no other OA in the sentence. The token ‘sich’ is only assigned the GF OD by the lexical component. Sentence 237 shows that the non-lexical component is also able to robustly deal with sentences which lack more than one GF.

[12]What the author seems to mean but does not write is ‘to prepare himself for his new challenge’.

(237) Denn dort hatte [$_{\text{OA}}$ sich] inzwischen mit Haider [$_{\text{ON}}$ eine
For there has itself meanwhile with Haider a
"neotribale Struktur] des öffentlichen Raumes"
"neo-tribal structure of the Public Space"
(wieder-)hergestellt.
(re-)established.

'For, with Haider, a "neo-tribal structure of the Public Space" has been (re-)established.'

In sentence 238 the case is different: Overall, the sentence contains eight GFs. Seven of these GFs are annotated by the lexical component of the parser. However, in the subordinate clause 'das den Stuttgartern in den vergangenen Monaten soviel Hoffnung auf die Zukunft machte' only the ON 'das' and the OA 'soviel Hoffnung' are annotated by the lexical component since the SC frame of the verb 'machen' does not contain the SC frame 'ON OD OA', which would be the correct one to be applied to the sentence. Thus, the rules for the SC frame 'ON OA' match since this SC frame is a subset of the correct one. However, the non-lexical component is able to correctly annotate the chunk 'den Stuttgartern' as the OD because it was identified as an unambiguous *dative.* This shows that the non-lexical component is able to robustly deal with cases in which there is, in fact, a SC frame for a verb but in which this SC frame is incomplete.

(238) Aber vor allem darum, ob [$_{\text{ON}}$ man] [$_{\text{OA}}$ das Konzept], [$_{\text{ON}}$
But most of all about that, whether one the concept,
das] [$_{\text{OD}}$ den Stuttgartern] in den vergangenen Monaten [$_{\text{OA}}$
which the Stuttgarters in the previous months so much
soviel Hoffnung] auf die Zukunft machte und [$_{\text{ON}}$ das] so eng
hope for the future made and which so closely
[$_{\text{OPP}}$ mit dem Namen Rangnick] verknüpft ist, schon vor der
with the name Rangnick associated is, already before the
neuen Saison wieder [$_{\text{OPP}}$ in die Tonne] hauen muß.
new season again into the bin throw has to.

'But, most of all, about the question whether we, already before the new season, have to chuck away the concept which has given so much hope for the future to the Stuttgarters during the previous months and which is so closely connected with Rangnick.'

	precision	recall	$F_{\beta=1}$
overall	85.51%	79.80%	82.55
OA	83.00%	82.71%	82.86
OD	83.71%	59.63%	69.65
OG	100.00%	7.69%	14.29
ON	90.97%	91.16%	91.07
OPP	71.14%	47.54%	57.00
OS	76.60%	60.86%	67.83
PRED	78.82%	72.13%	75.33

Table 11.11: Results of evaluation without support verb component

11.4.4 Impact of Support Verb Component

Table 11.11 shows the results of the evaluation of the GF parser without the support verb component. The results show that overall $F_{\beta=1}$ has dropped by 0.13 points. The contribution of the support verb component to overall annotation success is, thus, very humble, especially if one considers that the spport verb component comprises 31.2% of all the rules in the GF parser. Only the $F_{\beta=1}$ of OPPs is considerably better with the support verb component (+0.98 points). This does not mean, however, that the support verb component is unfit as such. This becomes clear if one takes a look at the results of the support verb component alone which is shown in table 11.12 and illustrated in figure 11.31. Precision for ONs, OAs and OPPs is outstanding and the fact that recall is low is due to the fact that only few verbs are support verbs. Precision for PREDs is by 12.15 percentage points lower than for the GF parser without support verb component but, due to the recall, $F_{\beta=1}$ rises by 0.26 points with the support verb component.

The reason why the support verb component still does not contribute significantly to the overall success of the parser is that the high-frequent GFs ON and OA can also be annotated straightforwardly and at an adequate accuracy rate by the non-lexical component. To a certain extent, this is also true of PREDs. Consequently, the support verb component can only contribute significantly to the annotation of the OPPs because these cannot be treated by the non-lexical component. Sentence 239 gives an example of the instantiation of the support verb 'ums Leben kommen' (lose one's life), which was annotated correctly by the support verb component.

	precision	recall	$F_{\beta=1}$
overall	91.73%	1.02%	2.02
OA	100.00%	0.86%	1.70
OD	0.00%	0.00%	0.00
OG	0.00%	0.00%	0.00
ON	96.77%	1.05%	2.09
OPP	100.00%	1.42%	2.81
OS	0.00%	0.00%	0.00
PRED	66.67%	1.68%	3.27

Table 11.12: Results of evaluation of support verb component

(239) Innerhalb weniger Minuten kommen [$_{\text{ON}}$ die Männer] auf skurrile
Within few minutes come the men in whimsical
Weise [$_{\text{OPP}}$ ums Leben].
manner around life.
'Within few minutes, the men whimsically lose their lives.'

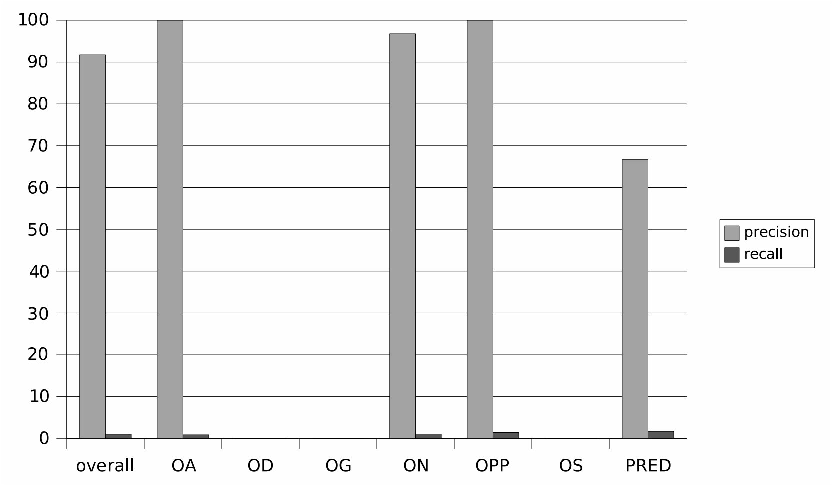


Figure 11.31: Results of evaluation of support verb component

Chapter 12

Comparison with Related Work

ABSTRACT: Chapter 12 compares the results of the parser in the thesis at hand with the results of comparable parsers. Section 12.1 briefly discusses the problems connected with comparing different parsing approaches. Section 12.2 presents the comparison of the evaluation results of the parser at hand with other shallow parsers and section 12.3 does so with the evaluation results of other GF parsers.

12.1 General Remarks

The parsing approaches whose results are discussed here have already been presented in chapter 3 in order to position our parser in the scientific field. In this chapter, we compare the quantitative evaluation results of those parsers with the evaluation results of the parser in the thesis at hand. In order to compare different parsing approaches, one would ideally test them using a *shared-task scenario*. In a shared-task scenario, different parsers are tested under the same conditions, i.e., they are tested on the same data annotating exactly the same categories. The only difference would concern the very approaches themselves, i.e. the actual parsers. Examples of such shared-task scenarios are presented, e.g., in Tjong Kim Sang and Buchholz (2000) for chunking and in Tjong Kim Sang and Déjean (2001) for clause identification.

Unfortunately, there is no such shared-task scenario for shallow parsing or for GF annotation for German. The reason for this may be seen in the comparatively new field of syntactic annotation of German and in the different definitions of syntactic categories which are annotated by the different parsing approaches. These are in part due to the different corpora which are used for training and development, on the one hand, and for testing, on the other hand. As a consequence of this, parsers not only differ in the categories they annotate but also in the data on which they are tested; and even if, for example, only newspaper texts are used, these can vary significantly with respect to the type of language occurring in them. This has to be kept in mind when we compare the figures for the various approaches in the following.

It is important to note that it is not only the language, i.e. the text type or domain, which differs when different language data are used for testing. It is also the degree and quality of **preprocessing** which differs; and this preprocessing is typically not described in detail when parsers are presented. Thus, the "quality" of the input text varies significantly. Before a text is handed over even to a PoS tagger, it is segmented into sentences and tokenized. This task may be achieved manually or (semi-)automatically and the standards of this segmentation may differ. In some data, even complex words may be recognized, which considerably eases the task of parsing. Some texts have headlines and footers removed.

12.2 Shallow Parsing

Tables 12.1 and 12.2 which show the quantitative evaluation results of the chunking component of our parser as described in section 10.3.1 are the same tables as tables 10.2 and 10.3. They are repeated here for convenience. They can be taken as a guideline for the comparison of our evaluation results with those of the other approaches. The results of the different approaches are presented here in the order in which those approaches are introduced in section 3.2.

Braun (1999); Neumann, Braun, and Piskorski (2000); Neumann and Piskorski (2002) give results for verb chunks (LK and VC in our approach) and for NPs and PPs which are defined as phrases without attachment. It does not say that appositions are included in this definition. (We include appositions in our chunk definition.) Verb chunks are tested on a corpus of 43 news reports containing 400 sentences with a total of 6,306 tokens. NPs and PPs are tested on a different corpus of 20,000 tokens of business news from the newspaper *Wirtschaftswoche*.[1] In both cases, tags

[1]The reason for the different corpora which are used can be seen in the fact that verb chunks belong to the topological field structure and are annotated and tested in Braun

are assigned automatically using MORPHIX (Finkler and Neumann, 1988). MORPHIX assigns PoS tag ambiguity classes on the basis of morphological information of the token. Then, constraints are applied to the tokens which reduce PoS tag ambiguity. In that way, 95.37% of all tokens receive a unique reading.

The results for verb chunks are 98.43% precision and 98.10% recall ($F_{\beta=1}$ 98.26). These results are in fact impressive if one considers that, typically, the PoS tagging error rate is about 2–4% depending on tagger and text type and that the correct assignment of verb chunks has to rely on PoS tags. Even if one builds a parser which takes into account second-best PoS tags, one would typically have problems achieving such good results because taking into account second-best tags would also introduce some mistakes. One explanation for these good results for the verb chunks and also for the clause annotation (see below) can be seen in the fact that the results of the PoS tagger are measured on another corpus (i.e. the *Wirtschaftswoche corpus*) and that they are significantly better on the corpus used for verb chunk and clause evaluation, which just contains 43 news reports. This is also supported by the fact that the NP/PP chunking evaluation, which is performed on the larger *Wirtschaftswoche corpus*, is of a different quality (see below.)

The PoS tagger we use has an accuracy of 96.36% on the whole TüBa-D/Z and 95.72% on the test section (cf. section 10.2 for details). Our chunker achieves results of 97.39% precision, 96.08% recall and $F_{\beta=1}$ 96.73 for the left part of the sentence bracket (LK) and 90.95%/94.49% /92.68[2] for the right part of the sentence bracket (VC) on TnT PoS tags. The test section for our chunker contains 18,093 tokens (1,000 sentences/7,185 chunks), i.e. three times the tokens contained in the test section in Braun (1999). The average sentence length is 18.1 as compared to 15.7 Braun (1999). Most of the errors of our chunker are due to tagging mistakes. This can also be seen in the results of our chunker on gold PoS tags. These are 99.90%/99.14%/99.52 for LK and 98.29%/99.35%/98.82 for VC.

The PoS tagging errors may have been caused by the quite heterogeneous composition of the *taz* newspaper which we use. The texts used in Braun (1999) all belong to the text type *news report* even if they are from three

(1999) while other chunks are tested in Neumann and Piskorski (2002). Nonetheless, the parser in Braun (1999) is presented with the same evaluation results as part of the system described in Neumann and Piskorski (2002) by those authors.

[2]With respect to evaluation results of our parser, the three figures in this order and format will henceforth stand for precision/recall/ $F_{\beta=1}$.

	precision	recall	$F_{\beta=1}$
overall	96.08%	96.10%	96.09
AC	94.87%	95.42%	95.14
C	99.73%	99.18%	99.46
LK	99.90%	99.14%	99.52
NC	94.10%	94.39%	94.24
PC	95.58%	94.86%	95.22
VC	98.29%	99.35%	98.82

Table 12.1: Quantitative evaluation results of our parser for chunks

different domains. This might also have had an effect on the chunker. This shows the problems for comparison which occur if no shared-task scenario is applied. Differences in performance may be caused by all types of effects outside the parsing algorithms, and they are, to a large extent, a matter of speculation.

As regards the results for NPs and PPs, the results of our parser are better than the ones in Neumann and Piskorski (2002) . Neumann and Piskorski (2002) give 91.94% precision and 76.11% recall ($F_{\beta=1}$ is 83.27) for NPs/PPs for their 20,000 tokens *Wirtschaftswoche* test section with automatic PoS tags. Our parser yields 87.42% /88.47% /87.94 for NCs and 92.02%/92.44%/92.23 for PCs. While precision for NCs in our approach is 4.5% lower than NP/PP precision in Neumann and Piskorski (2002), all other figures are higher – especially recall, which is considerably higher. Again, we could only speculate about the reasons for that.

Tables 12.3 and 12.4 show the quantitative evaluation results for topological fields as presented and discussed in section 10.3.2. They are repeated here for convenience. Braun (1999); Neumann, Braun, and Piskorski (2000); Neumann and Piskorski (2002) do not give any results for the annotation of topological fields. The reason for this can be seen in the status which topological fields have in their approach. While they are annotated for their own sake in our approach, they just serve the purpose of further annotation in Braun (1999); Neumann, Braun, and Piskorski (2000); Neumann and Piskorski (2002). The further purpose is the annotation of clause structure.

For the annotation of clauses, Braun (1999); Neumann, Braun, and Piskorski (2000); Neumann and Piskorski (2002) have made the distinction be-

	precision	recall	$F_{\beta=1}$
overall	90.17%	90.86%	90.51
AC	81.10%	79.94%	80.52
C	93.65%	92.12%	92.88
LK	97.39%	96.08%	96.73
NC	87.42%	88.47%	87.94
PC	92.02%	92.44%	92.23
VC	90.95%	94.49%	92.68

Table 12.2: Quantitative evaluation results of our parser for chunks with TnT PoS tags

	precision	recall	$F_{\beta=1}$
C	97.82%	97.95%	97.89
VF	94.41%	91.47%	92.92
MF	93.14%	89.52%	91.29

Table 12.3: Quantitative evaluation results for topological fields

	precision	recall	$F_{\beta=1}$
C	93.13%	93.78%	93.45
VF	93.42%	88.28%	90.78
MF	87.46%	82.59%	84.96

Table 12.4: Evaluation results for topological fields with TnT PoS tags

	precision	recall	$F_{\beta=1}$
V1/V2 clauses	88.18%	84.68%	86.40
VL clauses	93.35%	90.91%	92.12

Table 12.5: Quantitative evaluation results for clauses

	precision	recall	$F_{\beta=1}$
V1/V2 clauses	84.63%	79.27%	81.86
VL clauses	89.25%	84.75%	86.95

Table 12.6: Quantitative evaluation results for clauses with TnT PoS tags

tween *main clauses* and *base clauses*. This distinction is basically made in the same way as our distinction between V1/V2 clauses, on the one hand and VL clauses, on the other hand. The categories are, thus, relatively comparable. The results for main clauses are 93.80% precision, 93.08% recall and $F_{\beta=1}$ 93.43. Our approach yields 84.63% /79.27% /81.86. For base clauses, their results are 95.75% precision, 90.25% recall and $F_{\beta=1}$ 92.91. Our approach yields 89.25%/84.75%/86.95. Again, it is true what was said for chunks. Extremely good PoS tagging results make excellent parsing possible. The fact that the ratio of subordinate clauses per main clause is 39% in our test section (i.e. 1,607 subordinate clauses in 4,125 matrix clauses) and 32.5% in the one in Braun (1999); Neumann, Braun, and Piskorski (2000); Neumann and Piskorski (2002) (i.e. 130 subordinate clauses in 400 matrix clauses) can also be a hint that the texts used are of a less complex nature. Still, a more satisfactory comparison cannot be achieved without a shared-task scenario.

Wauschkuhn (1996) gives no comparable evaluation figures. Only the amount of sentences which receive at least one analysis is given (85.7%). There is no evaluation as to whether these sentences are annotated correctly. As can be seen in the tables above, the amount of structures which were annotated **correctly**, i.e. recall, is typically higher for all categories with our parser. Still, it is the merit of Wauschkuhn (1996) to have introduced the idea of topological field parsing.

Schiehlen (2002, 2003a,b) gives results for noun chunks. This includes prepositional chunks in his definition. The gold standard he uses is extracted from the NEGRA treebank (Skut, Krenn, Brants, and Uszkoreit, 1997) which contains 321,000 tokens from which Schiehlen (2002) extracts 100,974 *base noun chunks* and 78,942 *full noun chunks*. Obviously, the whole corpus is taken as test material. The NCs were extracted using “a quite sophisticated Perl script”. Correctness was not checked by hand, however. It has to be mentioned here that it is a problematic approach to extracting categories like chunks from a corpus which are not explicitly coded in it. To do so, one has to apply certain heuristics which in some cases more or less resemble those heuristics which are used to annotate the structures (i.e. chunks) one wants to evaluate. This is even more the case, if the writer of the extraction rules and the writer of the chunk grammar are the same person. Unfortunately, there is no easy way out of this dilemma as long as there is no chunked corpus, but the problem has to be kept in mind.

We do not have the distinction between base noun chunks and full noun chunks in our approach. What Schiehlen (2002) defines as base noun chunks does not come close to what we define as chunks. What he defines as full noun chunks comes quite close to what we define as chunks including complex chunks – only that Schiehlen (2002) also includes genitive modification in the full noun chunk definition. For automatically assigned PoS tags, Schiehlen (2002) gives a precision of 90.83% and a recall of 83.61% ($F_{\beta=1}$ 87.07) without agreement check and 91.11%/87.10%/89.05 with agreement check. We score 89.03%/89.90%/89.46 with a slightly different chunk definition on completely different data. These figures show that the results of the two chunkers are somewhat comparable. What is more important is that they show that in the approach in Schiehlen (2002) agreement check , i.e. the use of morphological information for chunking, is needed to achieve high recall and that this is not the case in our approach.

Kermes and Evert (2002, 2003) annotate PCs and NCs as well as adverbial and adjectival chunks. However, only NCs are evaluated. The NCs which are annotated largely correspond to the ones in the definition of Schiehlen (2002). Kermes and Evert (2002, 2003) also use the NEGRA corpus for evaluation. However, results cannot be compared because they use a different, later version of the NEGRA corpus (containing 355,096 tokens). Furthermore, they do not include prepositional chunks within the NC definition. Kermes and Evert (2002, 2003) also mention the problems related to the automatic extraction of chunks from a corpus in which they are not annotated. They state that it was not always possible to filter out material which “would usually not be subsumed under the category NP”. It is unclear whether this leads to a conversion which makes the evaluation results appear worse or better than would otherwise be the case.

For the evaluation on gold POS tags, Kermes and Evert (2003) give 88.21% precision and 90.03% recall ($F_{\beta=1}$ 89.11) for NCs. Our chunker scores significantly better with 94.10%/94.39%/94.24. However, the reservations to comparisons of that kind have to be kept in mind. For the evaluation with automatically assigned PoS tags, Kermes and Evert (2003) give 82.48%/86.11%/84.26 as the result as compared to 87.42%/88.47%/87.94 for our approach. It is astonishing that, again, the results are not better although agreement information is used. However, for a more detailed comparison, a more detailed evaluation would be needed.

Trushkina (2004) evaluates the annotation of chunks and topological fields (clauses are not included). The results are 95.31% precision, 96.43% recall and $F_{\beta=1}$ 95.87% on gold PoS tags on a 12,020 tokens test corpus generated from the TüBa-D/Z. Our parser yields 96.08%/96.10%/96.09 overall results for chunks. However, no detailed analysis of the results for each single category is given in Trushkina (2004). Hence, it is completely unclear how each category contributes to the overall results. This is even more important since Trushkina (2004) does not use maximal chunks for evaluation. Since the TüBa-D/Z, which she uses as a gold standard, is annotated in detail, it may contain a couple of chunks in one maximal chunk. Nonetheless, these chunks may be rather simple which may make annotation an easier task.

The chunk from TüBa-D/Z in 240 gives an example: On the whole, there are three chunks. But there is only one maximal chunk (i.e. a chunk not contained in any other chunk). In Trushkina (2004), the correct annotation of the prepositional chunk (with the included chunks) would yield three correct chunks while it would only yield one correct chunk if only the maximal chunk is considered. However, it is typically the maximal chunk which is the most complex one and which is the one which is hardest to annotate. In bottom-up systems, the annotation of the maximal chunk presupposes the correct annotation of the included chunks, and, in top-down systems (like ours), the correct annotation of the included chunks is a task which is far easier after the annotation of the maximal chunk.

(240)	[$_{\text{PX}}$ wie für	[$_{\text{NX}}$ den	[$_{\text{ADJX}}$ alkoholkranken]]	Mann]]
	like for	the	alcohol-ill	man

'like for a male alcoholic'

If one tries to compare the evaluation results of maximal chunks with the ones of all chunks, one has, thus, to keep in mind that the latter task is less complex. If, for example, the parser fails to annotate the maximal chunk in 240 (i.e. the prepositional chunk), but succeeds in annotating the other two chunks, then this yields 0% precision and 0% recall if only maximal chunks are considered but 66% precision and 66% recall if all chunks are considered. Still, the main flaw in the evaluation procedure in Trushkina (2004) is that it does not give a detailed overview of the results of each category although these results must exist since the figures which are given refer to **labelled** precision and recall.

Schmid and Schulte im Walde (2000) solely annotate and evaluate noun chunks. The NCs annotated include PCs in which the determiner and the preposition are morphologically combined. It also includes PCs centre-embedded in NCs and appositions. The parser therefore annotates slightly fewer structures than are annotated by our parser. The input to the parser is ambiguously tagged tokens. The parser is evaluated on 378 sentences which subsume 2,140 chunks (7,185 chunks in our test set). The results are 93.06% precision and 92.19% recall as compared to 87.42%/92.02% precision for NCs/PCs and 88.47%/92.44% recall for NCs/PCs in our parser (on automatic PoS tags).

Again, it is hard to compare results because of the different test set (which, in this case, is also more than three times smaller). However, our results compare well with the ones in Schmid and Schulte im Walde (2000) if one considers that our parser also annotates complex chunks (e.g. appositions). This shows that our parser is highly competitive to a hybrid grammar as well, since Schmid and Schulte im Walde (2000) employ a handcrafted context-free grammar augmented with probabilities (i.e. a PCFG). This is even more important to mention if one considers that the NC/PC component of the chunker only contains 72 chunking rules (including those for complex chunks) and is very easy to maintain.

Becker and Frank (2002) use PCFGs like Schmid and Schulte im Walde (2000). However, while Schmid and Schulte im Walde (2000) simply annotate noun chunks, Becker and Frank (2002) simply annotate topological fields and clauses. They use gold PoS tags as the input to their parser. The test section of their corpus contains 1,058 sentences as compared to 4,174 in our test section. Table 12.7 shows the results for the annotation of topological fields and clauses. For the four different types of topological fields which we compare, Becker and Frank (2002) score a higher $F_{\beta=1}$ twice, about the same one once and a lower one once, too. Becker and Frank (2002) also have slightly better results for clauses. Again, we can only stress that this shows

	precision	recall	$F_{\beta=1}$	our $F_{\beta=1}$
C/LK	99.6%	99.4%	99.50	99.46/99.52
RK	96.3%	95.8%	96.05	98.82
VF	96.1%	91.8%	93.90	92.92
MF	93.2%	93.1%	93.15	91.29
Clauses	88.9%	92.2%	90.52	88.01

Table 12.7: Results of Becker and Frank (2002) as compared to our results

that our parser is competitive with respect to the learning algorithm since it cannot be ruled out that the difference in evaluation results is caused by the difference in test sections (the one in Becker and Frank (2002) is also four times smaller than ours).

Klatt (2004a,b) gives evaluation results for chunks and topological fields. For the evaluation of the chunks, he takes the first 500 sentences from the REFD-corpus compiled by the IMS Stuttgart. Our chunk test section contains 1,000 sentences. The results are astonishingly good. Klatt (2004a) gives as results: 98.09% precision and 96.74% recall for NPs; 98.39% precision and 99.00% recall for DPs; and 98.22% precision and 98.00 recall for PPs. These results are much better than all the results of all other approaches. Especially in the light of this, one can see the need for a shared-task scenario. It would be extremely helpful to find out how these good results came about, even more if one considers that the results seem to have been produced with automatically assigned PoS tags.

For the evaluation of topological fields, Klatt (2004b) uses the first 1,225 sentences from the REFD-corpus. We have 4,174 sentences in our test section. Again, the results are exceptionally good. For instance, the $F_{\beta=1}$ for VFs is 98.99 while the $F_{\beta=1}$ for VFs is 93.90 in Becker and Frank (2002) and 92.92 in our approach. This is even more astonishing if one considers that Klatt (2004b) uses automatically assigned PoS tags while Becker and Frank (2002) and we use gold tags for the figures above. One of the reasons for the difference might be that Klatt (2004b) only annotates non-recursive topological fields. In the case of Klatt (2004b), it is also hard to compare results because he uses non-standard categories. There are, for example, no results for MFs but only for a combination of LK, MF and VC. Again, a shared task could help find the reasons for these differences.

Brants (1999) only annotates NCs and PCs. Since he uses ten-fold cross validation, he is able to take the whole NEGRA corpus as a test section. These are 17,000 sentences (300,000 tokens). The input to the parser is text which is segmented into sentences and tokenized. PoS tagging is achieved as the first step of the parser. PoS tagging accuracy is 96.5%. We achieve an accuracy of 96.36% on the whole TüBa-D/Z and 95.72% on the test section. Since Brants (1999) uses a cascaded Markov model he gives the results for each layer.

The maximum precision for NCs/PCs is 91.4%. However, this score is reached only for the first layer, which can only annotate less complex chunks. According to Brants (1999), even theoretically, only 72.6% of all chunks could be annotated by this layer. Consequently, recall is low on this level (54.0%). By the time the parser would be able to annotate all chunks (fourth layer), precision has fallen below 90%. Recall is highest for the ninth layer (84.8%). $F_{\beta=1}$, which combines the scores of both values, is highest on the seventh level: 86.5. This shows that our parser is highly competitive compared to this parser as well, since we score 89.03% precision, 89.90% recall and $F_{\beta=1}$ 89.46 for NCs and PCs combined (albeit with a far smaller test section).

To summarize, we can say that our shallow parser is in a very good position in the field of shallow parsing of German. The evaluation results are highly competitive for all categories which are annotated. This is true of the comparison with approaches which are rule-based and primarily finite-state like ours; but it is also true of approaches which apply learning algorithms or hybrid approaches. What is even more notable is that our parser annotates all shallow categories, i.e. chunks, topological fields and clauses. With respect to other shallow approaches, we can also stress that our approach stays completely within the finite-state paradigm and, thus, does not make use of any other more powerful formalisms. This is crucial for effectiveness. In Braun (1999); Neumann, Braun, and Piskorski (2000); Neumann and Piskorski (2002) and Schiehlen (2002, 2003a,b), it does not become clear in how far they are still within the finite-state formalism. Only Kermes and Evert (2002, 2003) stay completely within the FS formalism, but they achieve poorer results.

Furthermore, it has to be stressed that our shallow parser also works with much less information than the other parsers. The other three parsers which use FS technology also apply an agreement check, for which they need morphological information. Our approach simply makes use of PoS tag information and is still highly competitive. This fact is not trivial since morphological analyzers may work far less reliably on unknown text types and domains; and the performance of shallow parsers relying on such information may considerably drop if performance of the respective morphological analyzers drops. It is thus preferable to use as little information as possible to reach the intended goal.

The same is true of the comparison with our approach to the approaches using learning algorithms: Our shallow parser uses a very straightforward grammar consisting of only 288 chunk rules. 103 of these are for verb chunks since we place a great deal of emphasis on this structure. 80 rules are for all the other chunks. 78 rules are for topological fields and 27 rules are for clauses. By contrast, Schmid and Schulte im Walde (2000) use an existing grammar of 4,619 rules which they train on an unannotated corpus of 4–10 million words; and both Becker and Frank (2002) and Brants (1999) use the NEGRA corpus as their training material, a corpus which has been developed by several linguists over a number of years.

12.3 Grammatical Functions Annotation

Tables 12.8 and 12.9 which show the quantitative evaluation results of the GF annotation component of our parser as described in section 11.3 are the same tables as tables 11.2 and 11.3. They are repeated here for convenience. They can be taken as a guideline for the comparison of our evaluation results with those of the other approaches. The results of the different approaches are presented here in the order in which those approaches are introduced in section 3.3.

Schiehlen (2003a,b) annotates various types of grammatical roles as dependencies. This is different to our approach since we annotate grammatical functions as attributes of the phrases by which they are realized. Nevertheless, we contrast the results of the two approaches with each other since Schiehlen (2003a,b) is the approach which comes closest to ours. He also uses FS technology as the basis of his annotation (although he also introduces some non-FS components for final disambiguation; see section 3.3.2 for details). Again, it has to be kept in mind that the test material also differs. In the case at hand, the NEGRA corpus (Skut, Krenn, Brants, and Uszkoreit, 1997) was used for evaluation, this time a version containing about 340,000 tokens in 19,546 sentences.

When presenting the results for the annotation of grammatical roles,

	precision	recall	$F_{\beta=1}$
overall	85.52%	80.02%	82.68
OA	82.94%	82.77%	82.86
OD	83.66%	59.40%	69.47
OG	100.00%	7.69%	14.29
ON	90.93%	91.27%	91.10
OPP	72.06%	48.50%	57.98
OS	76.92%	60.67%	67.83
PRED	78.53%	72.87%	75.59

Table 12.8: Final results of evaluation of GF annotation

	precision	recall	$F_{\beta=1}$
overall	83.49%	75.40%	79.24
OA	80.85%	77.09%	78.92
OD	81.73%	57.08%	67.21
OG	100.00%	7.69%	14.29
ON	88.62%	87.84%	88.23
OPP	71.66%	44.63%	55.00
OS	72.96%	55.97%	63.34
PRED	75.70%	62.72%	68.60

Table 12.9: Final results of evaluation of GF annotation with TnT PoS tags

	precision	recall	recall Schiehlen lower bound	recall Schiehlen upper bound
overall	85.52%	80.02%		
ON	90.93%	91.27%	82.6%	84.5%
OA	82.94%	82.77%	70.6%	83.5%
OD	83.66%	59.40%	71.9%	77.1%
PRED	78.53%	72.87%	55.1%	–
OPP	72.06%	48.50%		
OS	76.92%	60.67%	91.2%	91.9%

Table 12.10: Our results contrasted with those in Schiehlen (2003b)

Schiehlen (2003b) gives two different figures. The reason for this is that grammatical roles are left underspecified by the FS part of the parser and are supposed to be disambiguated later by a non-FS component. As a *lower bound* (according to the definition given in Riezler, King, Kaplan, Crouch, Maxwell, and Johnson (2002)), Schiehlen (2003b) gives the performance (recall) as measured if one makes a random choice from the set of analyses yielded by the parser. As an upper bound, the results with optimal performance are given, i.e. a hypothetical performance in the cases where final disambiguation gets everything right.

As can be seen from the results in table 12.10, our results are highly competitive, even if one considers the reservations about comparisons which we already mentioned. The results show that Schiehlen (2003b) has a higher recall for dative and sentential objects but a considerably lower recall for the far more frequent subjects. Furthermore, even the (hypothetical) upper bound of accusative objects is only slightly above our effective recall. Still, it is not clear how the difference between the two parsers can be explained and a more apt method to compare the two approaches (e.g. a shared-task scenario) would be helpful.

Trushkina (2004) also uses the TüBa-D/Z as evaluation data. Therefore, the categories which are used can be contrasted even if they are annotated as dependencies unlike in our approach. Unfortunately, another section of the TüBa-D/Z is used for evaluation. Trushkina (2004) uses a 12,020 token subsection of the TüBa-D/Z as her test section while we use a 73,926

	precision	recall	$F_{\beta=1}$	precision	recall	$F_{\beta=1}$
	Trushkina (2004)			thesis at hand		
OA	91.38%	93.08%	92.22	82.94%	82.77%	82.86
OD	83.15%	83.69%	83.42	83.66%	59.40%	69.47
OG	100.00%	100.00%	100.00	100.00%	7.69%	14.29
ON	95.50%	96.28%	95.89	90.93%	91.27%	91.10
OPP	—	—	—	72.06%	48.50%	57.98
OS	71.27%	70.74%	71.00	76.92%	60.67%	67.83
PRED	70.83%	69.58%	70.20	78.53%	72.87%	75.59

Table 12.11: Final results of evaluation of GF annotation

token subsection. Table 12.11 shows the results of Trushkina (2004), p. 163, on gold PoS tags as contrasted with our results. It shows results which are in part considerably better than ours. However, Trushkina (2004) does not only use gold PoS tags but also includes the gold morphological information. We have explained in detail in section 9.1 how important morphological disambiguation is for GF annotation. It is, in fact, interesting how well a GF annotation approach does with gold morphological information but it is by far more interesting how well it does with automatic disambiguation of morphology, since this is one of the main tasks in the annotation of GFs.

As a matter of fact, complete disambiguation of morphology is in some cases, even for humans, only possible if the grammatical function of a constituent is recognized. Seen from another perspective, the annotation of morphology already resolves many of the ambiguities in GF annotation. This is especially true of those GFs connected with case but also of others since these can sometimes be identified by way of exclusion once the GFs connected with case are identified. Unfortunately, Trushkina (2004) does not give any detailed figures on how her parser does with automatically assigned morphology, although automatic morpho-syntactic disambiguation is one of the two main tasks of her thesis. Only the overall figures for the dependency parser with automatically assigned morphology as disambiguated by her system are given.

Table 12.11 shows that those GFs connected with case show considerably better results in Trushkina (2004) if contrasted to our approach. This is strongly influenced by the use of gold morphology. If one takes a look at the results of GFs not (or not only) connected with morphology, the results are different. There are no results for OPPs since the annotation of prepositional objects needs lexical subcategorization information which is not used in Trushkina (2004). $F_{\beta=1}$ for PREDs is five points lower than ours. Predicatives may be realized by nominative NPs but they may also be realized by PPs or adverbial phrases. $F_{\beta=1}$ for OS is 3.17 points lower in our approach but we have already mentioned that gold morphology may help in the annotation of GFs not connected with case as well.

We can furthermore contrast the results given in Trushkina (2004), p. 170, which show the overall results of her parser on automatic PoS tags and morphology with our overall results on TnT tags. This shows a precision of 86.05%, a recall of 85.06% and an $F_{\beta=1}$ of 85.55 in Trushkina (2004) as contrasted to 83.49%/75.40%/79.24 in our system (cf. table 12.9 for details). However, apart from the general problems connected with comparing different approaches on different data, there is the problem that only the overall results for automatic disambiguation are given in Trushkina (2004). However, these also include some categories which we do not include in our GF definition. This makes comparison hard.

What is more is that these categories do not need any morphological disambiguation for their recognition and are, thus, supposedly not affected by potential errors in automatic morphological disambiguation. One of the categories is the category ROOT which requires the detection of the finite verb in the clause, another is the category VPT which requires the detection of the verbal particle in complex verbs (which is the right part of the sentence bracket) and another is the category OV, which signifies the relations in the verbal chain (often within the same chunk). We subsume these phenomena under shallow annotation. That these categories are the ones which can be annotated most reliably can be seen in the results given in Trushkina (2004), p. 163. These are the three categories with the highest scores.

Since they are very likely not affected by errors in automatic morphological disambiguation, these three categories may keep overall results high even if automatic morphological disambiguation is applied. The annotation of GFs connected with morphology (crucial ones like subject and direct object) could, thus, be lower than the overall figures suggest. It is totally unclear why detailed figures are not given for fully automatic annotation because they are given for annotation on gold data. Since “labelled precision and recall” are given, detailed results obviously must have existed. Without detailed results, however, a satisfactory comparison is not possible.

Schmid and Schulte im Walde (2000) only annotate GFs connected with case but without relating them to clauses. They use a test set of 378 sentences as compared to 4,174 sentences in our test set. The overall results for the annotation of all case-related GFs is a precision of 83.88% and a recall of 83,21% on automatic PoS tags. Again, as in Trushkina (2004), no detailed results are given, and results can therefore not be properly compared and contrasted. This would be interesting because, in our approach, results for ODs, for example were considerably lower than for ONs and OAs. In tests, it was only possible to improve results for ODs by accepting a lower score for ONs and OAs; but this led to a decline in overall performance. In fact, it would have been possible to improve overall performance even more but we wanted to avoid unbalanced results because we wanted to have acceptable results for all categories. In learning approaches, it may well be possible that high-frequency categories receive much better results than low frequency categories. Without detailed results, it is not possible to check whether this is the case in Schmid and Schulte im Walde (2000). Furthermore, Schmid and Schulte im Walde (2000) do not make a distinction between ONs and PREDs realized as nominative NCs.

Kübler (2004a,b) evaluates syntactic categories and functional categories. The evaluation of syntactic categories belongs to a large extent to the domain of shallow parsing. However, like in the case of Trushkina (2004) and Schmid and Schulte im Walde (2000), no detailed results are given. Thus, it is not possible to even discriminate between shallow syntactic structures and deeper syntactic structures. The overall results for syntactic structures are 87.25% labelled precision and 82.45% labelled recall ($F_{\beta=1}$ 84.78). This includes structures which are as diverse as adverbial chunks, which mostly include only one element (i.e. an adverb), and topological fields, which might contain numerous complex phrases or even clauses. Consequently, there is no point in contrasting these results with any of ours.

Furthermore, Kübler (2004a), p. 215, and Kübler (2004b) evaluate syntactic categories together with functional categories. The latter largely subsume those phenomena which we define as GFs plus some modification phenomena. Since the TüBa-D/S, which is used by Kübler (2004a,b), has the same annotation scheme as the TüBa-D/Z, which we use, it would be possible to contrast the results (although, of course, the TüBa-D/S covers a completely different domain), but, again, no detailed results are given. The overall results for syntactic categories including their function are 75.79% labelled precision and 71.72% labelled recall ($F_{\beta=1}$ 73.70).

Kübler (2004a,b) observes that, for both evaluation methods, precision is higher than recall. This is attributed to the fact that more than 7% of the constituents are unattached. Consequently, Kübler (2004a,b) also evaluates the annotation of only those functional categories which are in fact attached. This evaluation yields an impressive result of 95.21% precision and 95.31% recall. Since the author attributes the high number of unattached constituents to sparse data problems, she expects overall performance to improve if more data is available. Although one can follow this reasoning, it has to be kept in mind that Kübler (2004a,b) already uses more than 34,000 sentences for training and that the creation of treebanks is an expensive and time-consuming task.

To summarize, we can say that our parser is the only one which remains within the finite-state framework when annotating GFs. With respect to quantitative results, it is highly competitive with the one in Schiehlen (2003a,b), which uses non-finite-state techniques when disambiguating underspecified GFs. With respect to the other parsers introduced above, we can only say that there is no reason to believe that they are significantly better than our parser but that there are no detailed results given which allow a fair evaluation at all.

Only the evaluation of our parser allows a detailed view of its performance while this is not the case for the other approaches. Schmid and Schulte im Walde (2000) and Kübler (2004a,b) only give overall performance, and Trushkina (2004) only gives detailed figures for the evaluation on gold PoS tags and morphology. This is all the more disappointing since the detailed evaluation of the effects of different components (e.g. POS tagging, morphological disambiguation etc.) of a parser gives valuable information to scientists dealing with similar problems. For this reason, we have extensively investigated the impact of the different components of our parser in chapter 11.

Chapter 13

Conclusion

We have presented a parser which uses FSAs to annotate both shallow syntactic structure and grammatical functions. While it was already known that chunks can be annotated effectively and accurately using FSAs, we have shown that it is also possible to extend this approach to topological fields and clauses. We have defined topological fields and clauses together with chunks as shallow annotation structures because all these structures adhere to the same ordering principle. They are subject to syntactic restrictions which can be covered by just using PoS tags. Consequently, they were annotated by one component. By contrast to most parsing approaches, we have not constructed a purely bottom-up or top-down parser but used a mixed bottom-up and top-down approach. At first, topological fields were annotated which were then combined to clauses (bottom-up). After this, chunks were annotated top-down. Recursive structures which do occur in topological fields and clauses and which cannot be annotated with the expressive power of an FSA were treated by iterating the FSAs up to a defined level. Evaluation for all shallow structures has shown satisfying results.

In annotating those structures first which could be annotated using the least information (i.e. PoS tag information), we applied an easy-first-parsing strategy. Grammatical functions, which also need lexical selection information and morphological features, were annotated after the shallow structures were annotated. Furthermore, the prior information of shallow structures is to be seen as a divide-and-conquer strategy since the scope of ambiguity has been largely reduced for GFs. Shallow parsing structures could, thus, be used as a basis for GF annotation and proved to be a useful basis for this task as evaluation results have shown. In addition to the scope reduction, linear order preferences in topological fields could be used to deal with ambiguity.

We have integrated the information of a sub-categorization lexicon into the parser to make the annotation of grammatical functions possible. Structures which were not covered by the information in the lexicon were annotated by a non-lexical component. In order to allow for the annotation of GFs, which are strongly connected with case features in German, we have assigned morphological ambiguity classes to the tokens in the corpus and used the shallow parsing structures to reduce the morphological ambiguity for all candidate GFs. The evaluation results for the non-lexical component have shown that our assumptions about the preferred GF order were largely correct and that our morphological ambiguity class reduction component was effective. The results have shown that most GFs can already be annotated with high precision without a sub-categorization lexicon. However, in order to annotate all GFs and in order to also achieve high recall, one needs such a lexicon.

Bibliography

Steven Abney. Parsing by chunks. In Robert Berwick, Steven Abney, and Carol Tenny, editors, *Principle-Based Parsing*. Kluwer Academic Publishers, Boston, 1991.

Steven Abney. Chunks and dependencies: Bringing processing evidence to bear on syntax. In Jennifer Cole, Georgia Green, and Jerry Morgan, editors, *Computational Linguistics and the Foundations of Linguistic Theory*, pages 145–164. CSLI, 1995.

Steven Abney. Chunk stylebook. Technical report, Seminar für Sprachwissenschaft, Universität Tübingen, Tübingen, 1996a. http://www.sfs.uni-tuebingen.de/˜abney/Papers.html#96i.

Steven Abney. Part-of-speech tagging and partial parsing. In Ken Church, Steve Young, and Gerrit Bloothooft, editors, *Corpus-Based Methods in Language and Speech*, An Elsnet Book. Kluwer Academic Publishers, 1996b.

Steven Abney. Partial Parsing via Finite-State Cascades. In *Proceedings of the ESSLLI-96 Workshop on "Robust Parsing"*, Prague, 1996c.

Salah Aït-Mokhtar and Jean-Pierre Chanod. Incremental finite-state parsing. In *Proceedings of the Conference on Applied Natural Language Processing (ANLP 1997)*, Washington, 1997a.

Salah Aït-Mokhtar and Jean-Pierre Chanod. Subject and object dependency extraction using finite-state transducers. In *ACL 97 Workshop on Information Extraction and the Building of Lexical Semantic Resources for NLP Applications*, Madrid, 1997b.

Salah Aït-Mokhtar, Jean-Pierre Chanod, and Claude Roux. Robustness beyond shallowness: incremental deep parsing. *Natural Language Engineering*, 8(2–3):121–144, 2002.

Markus Becker and Anette Frank. A stochastic topological parser for german. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan, 2002.

Douglas Biber, Stig Johansson, Geoffrey Leech, Susan Conrad, and Edward Finegan. *Longman Grammar of Spoken and Written English.* Longman, Harlow, 1999.

Thorsten Brants. Cascaded Markov Models. In *Proceedings of 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL 1999)*, Bergen, Norway, 1999.

Thorsten Brants. TnT – A Statistical Part-of-Speech Tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP 2000)*, Seattle, WA, 2000.

Thorsten Brants, Wojciech Skut, and Brigitte Krenn. Tagging Grammatical Functions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 1997)*, Providence, RI, 1997.

Christian Braun. Flaches und robustes Parsen deutscher Satzgefüge. Diplomarbeit, Universität des Saarlandes, Saarbrücken, 1999.

Joan Bresnan. Control and complementation. In Joan Bresnan, editor, *The mental representation of grammatical relations*, Cognitive theory and mental representation, pages 282–390. MIT Press, Cambridge, MA, 1982.

Hadumod Bußmann. *Lexikon der Sprachwissenschaft.* Alfred Kröner Verlag, Stuttgart, first edition, 1983.

Hadumod Bußmann. *Lexikon der Sprachwissenschaft.* Alfred Kröner Verlag, Stuttgart, third edition, 2002.

Miriam Butt, Tracy Holloway King, María-Eugenia Niño, and Frédérique Segond. *A Grammar Writer's Cookbook.* Number 95 in CSLI Lecture Notes. Center for the Study of Language and Information Stanford Publications, Stanford, 1999.

Nancy Chinchor and P. Robinson. Named Entity Task Definition (version 3.5). In *Proceedings of the 7th Message Understanding Conference (MUC-7)*, Washington, DC, 1998.

Noam Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2:113–124, 1956.

David Crystal. *A Dictionary of Linguistics and Phonetics.* Blackwell Publishers, London, fourth edition, 1997.

Jan Daciuk. *Incremental Construction of Finite-State Automata and Transducers, and their Use in the Natural Language Processing.* PhD thesis, Politechnika Gdańska, Gdańsk, Poland, 1998.

Steve J. DeRose. Grammatical Category Disambiguation by Statistical Optimization. *Computational Linguistics*, 14(1):31–39, 1988.

Erich Drach. *Grundgedanken der Deutschen Satzlehre.* Frankfurt am Main, 1937.

Judith Eckle-Kohler. *Linguistisches Wissen zur automatischen Lexikon-Akquisition aus deutschen Textcorpora.* Logos, Berlin, 1999.

Peter Eisenberg. *Grundriß der deutschen Grammatik*, volume 1: Das Wort. Metzler, Stuttgart, 1998.

Peter Eisenberg. *Grundriß der deutschen Grammatik*, volume 2: Der Satz. Metzler, Stuttgart, 1999.

Peter Eisenberg, Herrmann Gelhaus, Hans Wellmann, Helmut Henne, and Horst Sitta. *Duden, Grammatik der deutschen Gegenwartssprache*, volume 4 of *Der Duden: in 12 Bänden; das Standardwerk zur deutschen Sprache.* Dudenverlag, Mannheim, sixth, revised edition, 1998.

Ulrich Engel. *Deutsche Grammatik.* Julius Groos, Heidelberg, third, corrected edition, 1996.

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

Stefan Evert and Hannah Kermes. Annotation, storage, and retrieval of mildly recursive structures. In *Proceedings of the Workshop on Shallow Processing of Large Corpora (SProLaC 2003)*, pages 23–33, Lancaster, UK, 2003.

Stefano Federici, Simonetta Montemagni, and Vito Pirrelli. Shallow parsing and text chunking: A view on underspecification in syntax. In *Proceedings of the ESSLLI-96 Workshop on "Robust Parsing"*, pages 35–44, Prague, 1996.

Wolfgang Finkler and Günter Neumann. MORPHIX: A Fast Realization of a Classification-Based Approach to Morphology. In H. Trost, editor, *Proceedings der 4. Österreichischen Artificial-Intelligence Tagung, Wiener Workshop Wissensbasierte Sprachverarbeitung*, pages 11–19, Berlin, 1988. Springer.

Anette Frank, Tracy Holloway King, Jonas Kuhn, and John Maxwell. Optimality Theory Style Constraint Ranking in Large-Scale LFG Grammars. In Miriam Butt and Tracy Holloway King, editors, *Proceedings of the Third International Lexical Functional Grammar Conference (LFG 1998)*, Brisbane, 1998.

Roger Garside. The CLAWS Word-tagging System. In Roger Garside and Geoffrey Leech, editors, *The Computational Analysis of English: A Corpus-based Approach.* Longman, London, 1987.

James Paul Gee and François Grosjean. Performance structures: A psycholinguistic and linguistic appraisal. *Cognitive Psychology*, 15:411–458, 1983.

Barbara B. Greene and Gerald M. Rubin. Automated Grammatical Tagging of English. Technical report, Department of Linguistics, Brown University, Providence, RI, 1971.

Gregory Grefenstette. Light parsing as finite-state filtering. In *Proceedings of the ECAI-96 workshop on Extended Finite State Models of Language*, Budapest, 1996.

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

Claire Grover, Colin Matheson, Andrei Mikheev, and Marc Moens. LT TTT - A Flexible Tokenisation Tool. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC 2000)*, Athens, Greece, 2000.

Gerhard Helbig and Joachim Buscha. *Deutsche Grammatik. Ein Handbuch für den Ausländerunterricht.* Langenscheidt, Leipzig, nineteenth edition, 1999.

Elke Hentschel and Harald Weydt. *Handbuch der deutschen Grammatik.* de Gruyter, Berlin, second edition, 1994.

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

Erhard W. Hinrichs and Julia Trushkina. Forging Agreement: Morphological Disambiguation of Noun Phrases. In *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories (TLT 2002)*, pages 78–95, Sozopol, Bulgaria, 2002.

Hideki Hirakawa, Kenji Ono, and Yumiko Yoshimura. 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, 2000.

Tilman Höhle. Explikationen für "normale Betonung" und "normale Wortstellung". In Werner Abraham, editor, *Satzglieder im Deutschen: Vorschläge zur syntaktischen, semantischen und pragmatischen Fundierung*, volume 15 of *Studien zur deutschen Grammatik*, pages 75–153. Narr, Tübingen, 1982.

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

Aravind K. Joshi. Computation of syntactic structure. *Advances in Documentation and Library Science*, 3(2), 1961. Interscience Publishers.

Aravind K. Joshi. A parser from antiquity: an early application of finite state transducers to natural language processing. In Andras Kornai, editor, *Extended Finite State Models of Language*, Studies in Natural Language Processing, pages 6–15. Cambridge University Press, Cambridge, UK, 1999.

Aravind K. Joshi and Phil Hopely. A parser from antiquity. *Natural Language Engineering*, 2(4):291–294, 1996.

René Kager. *Optimality Theory*. Cambridge Textbooks in Linguistics. Cambridge University Press, Cambridge, UK, 1999.

Fred Karlsson. Constraint Grammar as a Framework for Parsing Unrestricted Text. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING 1990)*, pages 168–173, Helsinki, Finland, 1990.

Lauri Karttunen and Kenneth R. Beesley. A Short History of Two-Level Morphology. In *Special Event "Twenty Years of Finite-State Morphology" at the European Summer School in Logic, Language and Information (ESSLLI 2001)*, Helsinki, Finland, 2001.

Hannah Kermes and Stefan Evert. YAC – A Recursive Chunker for Unrestricted German Text. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1805–1812, Las Palmas (Gran Canaria), Spain, 2002.

Hannah Kermes and Stefan Evert. Text Analysis Meets Corpus Linguistics. In Dawn Archer, Paul Rayson, Andrew Wilson, and Tony McEnery, editors, *Proceedings of the Corpus Linguistics 2003 conference*, volume 16 of *UCREL technical paper*, pages 402–411, Lancaster, 2003.

Stefan Klatt. Combining a rule-based tagger with a statistical tagger for annotating German texts. In *Proceedings of the 6. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2002)*, Saarbrücken, 2002.

Stefan Klatt. A High Quality Partial Parser for Annotating German Text Corpora . In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, 2004a.

Stefan Klatt. Segmenting Real-Life Sentences into Topological Fields – For Better Parsing and Other NLP Tasks. In *Proceedings of the 7. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2004)*, Vienna, Austria, 2004b.

Sheldon Klein and Robert F. Simmons. A Computational Approach to Grammatical Coding of English Words. *Journal of the Association for Computing Machinery*, 10:334–347, 1963.

Sandra Kübler. *Memory-Based Parsing.* John Benjamins, Amsterdam, 2004a.

Sandra Kübler. Parsing without grammar – using complete trees instead. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2003)*, Borovets, Bulgaria, 2004b.

Sandra Kübler and Heike Telljohann. Towards a Dependency-Based Evaluation for Partial Parsing. In *Beyond PARSEVAL – Towards Improved Evaluation Measures for Parsing Systems – Workshop at the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas (Gran Canaria), Spain, 2002.

Henry Kucera and W. Nelson Francis. *Computational analysis of presentday American English.* Brown University Press, Providence, RI, 1967.

Claudia Kunze. Lexikalisch-semantische Wortnetze. In Kai-Uwe Carstensen, Christian Ebert, Cornelia Endriss, Susanne Jekat, Ralf Klabunde, and Hagen Langer, editors, *Computerlinguistik und Sprachtechnologie*, pages 386–393. Spektrum Akademischer Verlag, Heidelberg, 2001.

Claudia Kunze and Andreas Wagner. GermaNet - representation, visualization, application. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1485–1491, Las Palmas (Gran Canaria), Spain, 2002.

Geoffrey Leech. Grammatical tagging. In Roger Garside, Geoffrey Leech, and Tony McEnery, editors, *Corpus Annotation. Linguistic Information from Computer Text Corpora*, pages 19–33. Longman, London, 1997.

Geoffrey Leech and Nicholas Smith. *Manual to accompany the British National Corpus (Version 2) with Improved Word-class Tagging.* UCREL, Lancaster University, Lancaster, March 2000. URL http://www.comp.lancs.ac.uk/ucrel/bnc2/bnc2postag_manual.htm.

Geoffrey Leech and Andrew Wilson. Recommendations for the Morphosyntactic Annotation of Corpora. Recommendations, Expert Advisory Group on Language Engineering Standards (EAGLES), March 1996. URL http://www.ilc.pi.cnr.it/EAGLES96/annotate/annotate.html.

Jürgen Lenerz. *Zur Abfolge nominaler Satzglieder im Deutschen*, volume 5 of *Studien zur deutschen Grammatik*. TBL Verlag Gunter Narr, Tübingen, 1977.

Beth Levin. *English verb classes and alternations: a preliminary investigation*. University of Chicago Press, Chicago, 1993.

Wolfgang Lezius, Stefanie Dipper, and Arne Fitschen. IMSLex - Representing Morphological and Syntactical Information in a Relational Database. In *Proceedings of The 9th Euralex International Congress*, Stuttgart, 2000.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA, 1999.

Mitchell Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

Ian Marshall. Choice of Grammatical Word-Class without Global Syntactic Analysis: Tagging Words in the LOB Corpus. *Computers in the Humanities*, 17:139–150, 1983.

Walt Detmar Meurers. On the use of electronic corpora for theoretical linguistics. Case studies from the syntax of German. *Lingua*, 115(11):1619–1639, 2005.

Walt Detmar Meurers and Guido Minnen. A Computational Treatment of Lexical Rules in HPSG as Covariation in Lexical Entries. *Computational Linguistics*, 23(4):543–568, 1997.

Frank Henrik Müller. Annotating Grammatical Functions in German Using Finite-State Cascades. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland, 2004.

Frank Henrik Müller and Tylman Ule. Annotating topological fields and chunks – and revising POS tags at the same time. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan, 2002.

Günter Neumann and Jakub Piskorski. A Shallow Text Processing Core Engine. *Computational Intelligence, An International Journal*, 18(3):451–476, 2002.

Günter Neumann, Rolf Backofen, Judith Baur, Markus Becker, and Christian Braun. An information extraction core system for real world german text processing. In *Proceedings of 5th Conference on Applied Natural Language Processing (ANLP 1997)*, Washington, 1997.

Günter Neumann, Christian Braun, and Jakub Piskorski. 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)*, Seattle, WA, 2000.

Kemal Oflazer. Dependency Parsing with an Extended Finite State Approach. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL 1999)*, College Park, Maryland, 1999.

Kemal Oflazer. Dependency Parsing with an Extended Finite-State Approach. *Computational Linguistics*, 29(4):515–544, 2003.

Barbara Hall Partee, Alice ter Meulen, and Robert Eugene Wall. *Mathematical methods in linguistics.* Kluwer, Dordrecht, corrected first edition, 1993.

Li-Shiuan Peh and Christopher H. Ting. A Divide-and-Conquer Strategy for Parsing. In *Proceedings of the ACL/SIGPARSE 5th International Workshop on Parsing Technologies*, pages 57–66, 1996.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. *A comprehensive grammar of the English language.* Longman, London, 1985.

Lance A. Ramshaw and Mitchell P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the 3rd ACL Workshop on Very Large Corpora*, pages 82–94, Cambridge, MA, 1995.

Marga Reis. On Justifying Topological Frames: ‘Positional Field’ and the Order of Nonverbal Constituents in German. *DRLAV: Revue de Linguistique*, 22/23:59–85, 1980.

Marga Reis. Zum Subjektbegriff im Deutschen. In Werner Abraham, editor, *Satzglieder im Deutschen: Vorschläge zur syntaktischen, semantischen und pragmatischen Fundierung*, volume 15 of *Studien zur deutschen Grammatik*, pages 171–211. Narr, Tübingen, 1982.

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, and Mark Johnson. Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, Philadephia, PA, 2002.

Dietmar Rösner. Combining Robust Parsing and Lexical Acquisition in the XDOC System. In *Proceedings of the 5. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2000)*, Berlin, 2000.

Peter Schefe. Zur linguistischen und psychologischen Komplexität von Selbsteinbettungen. *Linguistische Berichte*, 35:38–44, 1975.

Michael Schiehlen. Experiments in German Noun Chunking. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei, Taiwan, 2002.

Michael Schiehlen. A Cascaded Finite-State Parser for German. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, pages 163–166, Budapest, Hungary, 2003a.

Michael Schiehlen. Combining Deep and Shallow Approaches in Parsing German. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, Sapporo, Japan, 2003b.

Anne Schiller. DMOR: Benutzer-Handbuch. Draft, Universität Stuttgart, Institut für maschinelle Sprachverarbeitung (IMS), Stuttgart, Germany, 1995.

Anne Schiller, Simone Teufel, and Christine Thielen. *Guidelines für das Tagging deutscher Textcorpora mit STTS.* IMS Stuttgart und SfS Tübingen, Stuttgart und Tübingen, 1995.

Helmut Schmid. Improvements in Part-of-Speech Tagging with an Application to German. In Susan Armstrong, Kenneth Church, Pierre Isabelle, Sandra Manzi, Evelyne Tzoukermann, and David Yarowsky, editors, *Natural Language Processing Using Very Large Coprora*, volume 11 of *Text, Speech and Language Technology*, pages 13–26. Kluwer Academic Publishers, Dordrecht, 1999.

Helmut Schmid and Sabine Schulte im Walde. Robust German Noun Chunking with a Probabilistic Context-Free Grammar. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, Saarbrücken, Germany, 2000.

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. An Annotation Scheme for Free Word Order Languages. In *Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP)*, Washington, D.C., 1997.

Rosmary Stegmann, Heike Telljohann, and Erhard W. Hinrichs. Stylebook for the German Treebank in VERBMOBIL. VERBMOBIL Report, Seminar für Sprachwissenschaft, Universität Tübingen, Tübingen, September 2000.

Heike Telljohann, Erhard W. Hinrichs, and Sandra Kübler. Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z). Technical report, Seminar für Sprachwissenschaft, Universität Tübingen, Tübingen, 2003. http://www.sfs.uni-tuebingen.de/resources/sty.pdf.

Erik F. Tjong Kim Sang and Sabine Buchholz. Introduction to the CoNLL 2000 Shared Task: Chunking. In *Proceedings of the 4th Conference on Natural Language Learning (CoNLL 2000) and LLL 2000.* Lisbon, Portugal, 2000.

Erik F. Tjong Kim Sang and Hervé Déjean. Introduction to the CoNLL 2001 Shared Task: Clause Identification. In *Proceedings of the 5th Conference on Natural Language Learning (CoNLL 2001).* Toulouse, France, 2001.

Julia Trushkina. *Morpho-Syntactic Annotation and Dependency Parsing of German.* PhD thesis, Universität Tübingen, Seminar für Sprachwissenschaft, Tübingen, 2004.

Tylman Ule and Frank Henrik Müller. KaRoPars: Ein System zur linguistischen Annotation großer Text-Korpora des Deutschen. In Alexander Mehler and Henning Lobin, editors, *Automatische Textanalyse. Systeme und Methoden zur Annotation und Analyse natürlichsprachlicher Texte*, pages 185–202, Opladen, 2004. VS Verlag für Sozialwissenschaften.

Oliver Wauschkuhn. Ein Werkzeug zur partiellen syntaktischen Analyse deutscher Textkorpora. In Dafydd Gibbon, editor, *Natural language processing and speech technology: results of the 3rd KONVENS Conference, Bielefeld, October 1996*, pages 356–368. Mouton de Gruyter, Berlin, 1996.

Appendix A

Abbreviations

$C_{[1-n]}$	Constraint 1–n
CFG	Context-Free Grammar
FSA	Finite-State Automaton
FST	Finite-State Transducer
IR	Information Retrieval
MF	Mittelfeld (middle field)
MM	Markov Model
MT	Machine Translation
NE	Named Entity
NF	Nachfeld (final field)
NP	Noun Phrase or Nominalphrase
NC	Noun Chunk or Nominal-Chunk
NLP	Natural Language Processing
OG	Genitive Object
OD	Dative Object
OA	Accusative Object
ON	Nominative Object
OS	Sentential Object
OT	Optimality Theory
PC	Prepositional Chunk or Präpositional-Chunk
PCFG	Probabilistic Context-Free Grammar
PoS	Part-of-Speech
PP	Prepositional Phrase or Präpositionalphrase
PRED	Predicative
RE	Regular Expression
RM format	Ramshaw-Marcus format
STTS	Stuttgart-Tübingen Tagset
TüBa-D/Z	Tübinger Baumbank des Deutschen/Schriftsprache
VF	Vorfeld (initial field)
XML	Extensible Markup Language

Appendix B

The Stuttgart-Tübingen Tagset (STTS)

POS =	description	examples
ADJA	attributive adjective	*[das] große [Haus]*
ADJD	adverbial oder predicative adjective	*[er fährt] schnell [er ist] schnell*
ADV	adverb	*schon, bald, doch*
APPR	preposition; left circumposition	*in [der Stadt], ohne [mich]*
APPRART	preposition + article	*im [Haus], zur [Sache]*
APPO	postposition	*[ihm] zufolge, [der Sache] wegen*
APZR	right circumposition	*[von jetzt] an*
ART	definite or indefinite article	*der, die, das, ein, eine*
CARD	cardinal number	*zwei [Männer], [im Jahre] 1994*
FM	foreign language material	*[Er hat das mit "] A big fish [" übersetzt]*
ITJ	interjection	*mhm, ach, tja*
KOUI	subordinating conjunction with "zu" and infinitive	*um [zu leben], anstatt [zu fragen]*
KOUS	subordinating conjunction with clause	*weil, daß, damit, wenn, ob*
KON	coordinating conjunction	*und, oder, aber*

POS =	description	examples
KOKOM	particle of comparison, no clause	*als, wie*
NN	noun	*Tisch, Herr, [das] Reisen*
NE	proper noun	*Hans, Hamburg, HSV*
PDS	substituting demonstrative pronoun	*dieser, jener*
PDAT	attributive demonstrative pronoun	*jener [Mensch]*
PIS	substituting indefinite pronoun	*keiner, viele, man, niemand*
PIAT	attributive indefinite pronoun without determiner	*kein [Mensch], irgendein [Glas]*
PIDAT	attributive indefinite pronoun with determiner	*[ein] wenig [Wasser], [die] beiden [Brüder]*
PPER	irreflexive personal pronoun	*ich, er, ihm, mich, dir*
PPOSS	substituting possessive pronoun	*meins, deiner*
PPOSAT	attributive possessive pronoun	*mein [Buch], deine [Mutter]*
	relative pronoun	
PRELS	substituting	*[der Hund,] der*
PRELAT	attributive relative pronoun	*[der Mann ,] dessen [Hund]*
PRF	reflexive personal pronoun	*sich, einander, dich, mir*
PWS	substituting interrogative pronoun	*wer, was*
PWAT	attributive interrogative pronoun	*welche [Farbe], wessen [Hut]*
PWAV	adverbial interrogative or relative pronoun	*warum, wo, wann, worüber, wobei*
PAV	pronominally used preposition	*dafür, dabei, deswegen, trotzdem*
PTKZU	"zu" with infinitive	*zu [gehen]*
PTKNEG	negation	*nicht*
PTKVZ	separated verb particle	*[er kommt] an, [er fährt] rad*
PTKANT	answer particle	*ja, nein, danke, bitte*
PTKA	particle with adjective	*am [schönsten],*

POS =	description	examples
	or adverb	*zu [schnell]*
TRUNC	truncated word - first part	*An– [und Abreise]*
VVFIN	finite main verb	*[du] gehst, [wir] kommen [an]*
VVIMP	imperative, main verb	*komm [!]*
VVINF	infinitive, main	*gehen, ankommen*
VVIZU	infinitive with “zu”, main	*anzukommen, loszulassen*
VVPP	perfect participle, main	*gegangen, angekommen*
VAFIN	finite verb, aux	*[du] bist, [wir] werden*
VAIMP	imperative, aux	*sei [ruhig !]*
VAINF	infinitive, aux	*werden, sein*
VAPP	perfect participle, aux	*gewesen*
VMFIN	finite verb, modal	*dürfen*
VMINF	infinitive, modal	*wollen*
VMPP	perfect participle, modal	*[er hat] gekonnt*
XY	non-word containing special characters	*D2XW3*
$,	comma	,
$.	sentence-final punctuation	**. ? ! ; :**
$(	sentence -internal punctuation	– [ ] ( )

From Schiller et al. (1995) (p. 7–8).

Appendix C

The TüBa-D/Z Inventory of Syntactic Categories and Grammatical Functions

All tables are taken from Telljohann et al. (2003) (p. 20–22).

Node Labels	**Description**
	Phrase Node Labels
NX	noun phrase
PX	prepositional phrase
ADVX	adverbial phrase
ADJX	adjectival phrase
VXFIN	finite verb phrase
VXINF	infinite verb phrase
FX	foreign language phrase
DP	determiner phrase (e.g. *gar keine*)
	Topological Field Node Labels
LV	resumptive construction (Linksversetzung)
VF	initial field (Vorfeld)
LK	left sentence bracket (Linke (Satz-)Klammer)
MF	middle field (Mittelfeld)
VC	verb complex (Verbkomplex)
NF	final field (Nachfeld)
C	complementizer field (C-Feld)
KOORD	field for coordinating particles
PARORD	field for non-coordinating particles
FKOORD	coordination consisting of conjuncts of fields
MFE	middle field between VCE and VC
VCE	verb complex with the split finite verb of *Ersatzinfinitiv* constructions
FKONJ	conjunct consisting of more than one field
	Root Node Labels
SIMPX	simplex clause
R-SIMPX	relative clause
P-SIMPX	paratactic construction of simplex clauses
DM	discourse marker

Table C.1: Node labels in TüBa-D/Z

Edge Labels	**Description**
Edge Labels denoting Heads and Conjuncts	
HD	head
-	non-head
KONJ	conjunct
Complement Edge Labels	
ON	nominative object
OD	dative object
OA	accusative object
OG	genitive object
OS	sentential object
OPP	prepositional object
OADVP	adverbial object
OADJP	adjectival object
PRED	predicate
OV	verbal object
FOPP	optional prepositional object
VPT	separable verb prefix
APP	apposition
Modifier Edge Labels	
MOD	ambiguous modifier
ON-MOD, OA-MOD, OD-MOD, OG-MOD, OPP-MOD, OS-MOD, PRED-MOD, FOPP-MOD, OADJP-MOD, V-MOD, MOD-MOD	modifiers modifying complements or modifiers e.g. V-MOD = modifier of the verb
Edge Labels in Split-up Coordinations	
ONK, ODK, OAK, FOPPK, OADVPK, PREDK, MODK, V-MODK	second conjunct (K) in split-up coordinations e.g. ONK = second conjunct of a nominative object (subject)
Secondary Edge Labels	
	dependency relation between:
REFVC	two verbal objects in VC
REFMOD	two ambiguous modifiers
REFINT	a phrase internal part and its modifier
REFCONTR	control verb and its complement across clause boundaries

Table C.2: Edge labels in TüBa-D/Z

Labels	**Description**
Phrase Node Labels	
EN-ADD	proper noun or named entity (additional label)
Secondary Edge Label	
EN	phrase internal relation between two parts of a proper noun

Table C.3: Labels for proper nouns and named entities in TüBa-D/Z

Appendix D

Lists of Verbs without SC Frame

D.1 112 Verb Types without SC Frame

ab#gehen: 1
ahnen: 1
an#sehen: 1
arbeiten: 1
auf#gehen: 1
auf#lockern: 1
auf#peppen: 1
auf#wachen: 3
auf#warten: 2
aus#helfen: 1
aus#realisieren: 1
aus#weichen: 1
begeben: 3
bejagen: 1
bekommen: 1
besitzen: 1
beteiligen: 1
bewerfen: 1
blöken: 1
changieren: 1

covern:	1
davor#sitzen:	1
dazu#gehören:	1
dazu#gewinnen:	1
dran#bleiben:	1
drauf#klatschen:	1
drauflos#reden:	1
durch#inszenieren:	1
ehren:	2
ein#hämmern:	1
ein#kaufen:	1
entgegen#flackern:	1
entgegen#hoppeln:	1
ernähren:	1
fackeln:	1
freuen:	1
gelangen:	1
glucksen:	1
haben:	3
her#wetzen:	1
herab#schreiten:	1
herbei#phantasieren:	1
herum#leiten:	1
herum#zeigen:	1
hin#bomben:	1
hin#schlendern:	1
hinab#blicken:	1
hinab#schlittern:	1
hinaus#ragen:	1
hinein#schliddern:	1
hinter#ziehen:	1
hoch#jubeln:	1
hoch#reißen:	2
hoch#würgen:	1
hypnotisieren:	1
irren:	1
kommen:	1
köcheln:	2

lassen:	1
liegen:	2
müssen:	1
nach#verhandeln:	1
outen:	2
plagen:	1
pluckern:	1
posaunen:	1
präsentieren:	1
pöbeln:	1
ran#machen:	1
raus#hauen:	1
rennen:	1
residieren:	1
runter#kochen:	1
sagen:	2
scheinen:	1
schnarren:	1
schreiben:	2
schwanen:	1
sichten:	1
sitzen:	17
sollen:	2
stehlen:	1
still#legen:	1
um#steuern:	1
unterlaufen:	1
unterstützen:	1
verhalten:	1
versichern:	1
verwirklichen:	1
vor#breiten:	1
vor#räubern:	1
vor#weinen:	1
weg#locken:	1
weiter#bomben:	1
weiter#schleppen:	1
wieder#erstehen:	1

wirken: 1
wohnen: 3
zerbröseln: 1
zu#gesellen: 1
zu#lassen: 1
zu#nehmen: 1
zurück#wollen: 1
zusammen#basteln: 1
äußern: 1
über#gehen: 2
über#wohnen: 1
übergehen: 1
#fallen:[1] 1
#lernen: 2
#treten: 1
#wehen: 1

D.2 44 Verb Types without Base Form

abschmeckten: 1
abzusteigen: 1
beinflußt: 1
bemopsen: 1
beruhig': 1
bewahre: 1
bezahltc: 1
bleiben: 1
charten: 1
drükken: 2
durchkämmt: 1
e-melden: 1
entschieden: 1
flösse: 1
gelinkt: 1
koscht: 1

[1]These are verbs in which the verbal particle was not analyzed.

lassen: 1
lebe: 1
offenbart: 1
pikken: 1
prüfe: 1
schwiemelt: 1
schwurbelte: 1
sein: 1
sozialversichert: 1
sterbien: 1
vernutten: 1
verortet: 1
verschrien: 1
vervierfachen: 1
vollstreckt: 1
vorliegen: 1
wegfallen: 1
werd: 1
zusamenstellte: 1
zwangsrekrutiert: 1
übersprüht: 1
FSC-zertifiziert: 1
Greifen: 1
Nutzen: 1
Sage: 1
Sehen: 1
Vergessen: 1
Weine: 1
(wieder-)hergestellt: 1