Getting a Grip on Morphological Disambiguation

Erhard W. Hinrichs and Julia S. Trushkina
Department of Computational Linguistics
Eberhard-Karls-Universität Tübingen
Wilhelmstr. 113, D-72074 Tübingen, Germany
{eh,jul}@sfs.uni-tuebingen.de

Abstract

The research reported here is part of a larger project on the development of a robust dependency parsing scheme GRIP (GeRman Incremental Parsing) that uses the Xerox Incremental Deep Parsing System (XIP) (Ait and Chanod and Roux, 2002) and provides syntactic annotation in an incremental fashion. It is shown that morphological disambiguation is a crucial step in narrowing down the search space for the correct assignment of dependency structures. Customized disambiguation rules of XIP provide the necessary computational tools to efficiently carry out morphological disambiguation for languages with rich inflectional morphology such as German. A quantitative evaluation on a German test corpus shows that application of XIP disambiguation rules in conjunction with heuristics about the syntactic context of German NPs yield unique morphological analyses as input for assigning dependency relations in 80% of all cases.[1]

1 Introduction

The research reported in this paper is part of a larger project on the development of a robust parsing scheme GRIP (GeRman Incremental Parsing) that uses the Xerox Incremental Deep Parsing System (XIP) and provides syntactic annotation in an incremental fashion: after textual input is tokenized, morphologically analyzed and disambiguated, syntactic annotation is added in two distinct stages of processing. First, a chunk parser provides a partial constituent analysis. In a second stage, the chunked input is further annotated by dependency links that reflect the function-argument structure for each chunked clause. This latter stage of processing is inspired by ideas originating in frameworks of dependency grammar which express grammatical relations as independent notions, rather than as a secondary concept derivable from constituent structure only.[2]

The current paper addresses one specific subtask in the overall GRIP parsing scheme: morphological disambiguation. We will demonstrate that morphological disambiguation is a crucial step in narrowing down the search space for the correct assignment of dependency structures. Furthermore, we will describe in detail the customized disambiguation rules of XIP that provide the necessary computational tools to efficiently carry out morphological disambiguation for languages with rich inflectional morphology.

2 Incremental Syntactic Annotation

Due to its incremental nature, GRIP crucially relies on the accuracy of annotation at previous levels. Chunking will depend on the accuracy of part-of-speech disambiguation, while dependency parsing relies crucially on the structure of the pre-chunked input and on the morphological properties of individual chunks. For example, in order to determine the subject of a clause, case and number information associated with the NP chunks that occur in the clause are of primary importance. For languages with rich inflectional morphology, it can often be difficult to determine such case and number information

[1]The authors deeply appreciate the contribution of Dr. Claude Roux of XRCE, Grenoble who is the chief implementor of the XIP System and who customized the disambiguation tools necessary for German morphological disambiguation.

[2]See (Mel'cuk, 1988) for an introduction to dependency grammar and for recent applications of dependency grammars to syntactic annotation and parsing see, among others, (Hajic, 1998), (Tapanainen and Järvinen, 1994) and (Duchier, 1999).

uniquely since one and the same word form may be associated with more than one combination of case, number and gender values. Consider the German sentence in (1):

(1) Die Politiker gaben verdienten
the politicians gave worthy
Beamten ein höheres Gehalt.
civil servants a higher salary

'The politicians gave worthy civil servants a higher salary.'

As shown in (2) - (4), the lexical nodes for each NP in the sentence are, almost without exception, morphologically many times ambiguous. The analyses shown are those provided by the morphological analyzer for German developed by the Xerox Research Centre Europe (XRCE). [3]

(2)

Die	Pron+Dem+FMN+Pl+NomAcc
Die	Pron+Dem+Fem+Sg+NomAcc
Die	Pron+Rel+FMN+Pl+NomAcc
Die	Pron+Rel+Fem+Sg+NomAcc
Die	Det+Def+Fem+Sg+NomAcc+St
Die	Det+Def+FMN+Pl+NomAcc+St
Politiker	Noun+Masc+Sg+NomAccDat
Politiker	Noun+Masc+Pl+NomAccGen

For example, the noun *Politiker* has a unique value only for gender (Masc). Number and case values are not unique and co-vary. In the singular reading, the form can be any case but genitive. For the plural only the dative case is ruled out.[4] The preceding token *die* exhibits a three-way word class ambiguity between a determiner reading (Det), a demonstrative pronoun reading and a relative pronoun reading. The latter two will in all likelihood be eliminated by a reliable part-of-speech tagger. However, even for the remaining determiner reading there are several distinct readings: *die*, taken in isolation, can be (i) nominative or accusative, singular, feminine, or (ii) nominative or accusative plural for any gender.[5] However, in the context of the following noun *Politiker*, only the latter reading is valid since it matches the gender specification of the noun. In the other direction, the determiner also helps to partially disambiguate the contextually valid readings of the noun by retaining as possible values of case nominative and accusative. The discussion of this first example shows the nature of this kind of contextual morphological disambiguation: lexical nodes within the same chunk mutually constrain each other as to the set of possible readings.

While example (2) requires identity of case, number and gender values between determiner and noun, other combinations of lexical categories require distinct values for certain morphological features.[6] In German, word forms for adjectives and determiners can be classified as belonging to either weak or strong declension classes.[7] For example, all forms of the definite determiner *der* belong to the strong declension class, while the paradigm of the indefinite determiner *ein* is split between weak and strong forms. In addition, some nouns, in particular those derived from adjectives like *Beamter*, also exhibit a distinction between weak and strong forms.

If determiners co-occur with adjectives and nouns in the same NP, adjective and noun agree in declension class, whereas the declension value of the determiner is the opposite. The NP *ein höheres Gehalt* and the set of candidate analyses in (3) demonstrate this. The only contextually valid reading is provided by the following sequence of morphological tags: `Det+Indef+Neut+Sg+NomAcc+Wk`, `Adj+Neut+Sg+NomAcc+St`, `Noun+Neut+Sg+` `+NomAcc`.

The morphological analysis for the NP *verdienten Beamten* exemplifies agreement of declension values between adjective and noun.

[3] An on-line demo version of the XRCE morphological analyzer is available at `www.xrce.xerox.com/competencies/content-analysis/demos/german.de.html`.

[4] The morphological tag `NomAccGen` stands for *nominative*, *accusative* or *genitive*.

[5] The morphological tag `FMN` stands for any gender.

[6] Even for determiners and nouns identity of case, number and gender values is sometimes too strong a constraint. If the determiner is realized by a relative pronoun as in *dessen Mutter* ('whose mother'), a mismatch in case values needs to be allowed.

[7] For a comprehensive study of distributional properties of weak and strong forms in German NP see (Zwicky, 1986).

(3)

ein	Det+Indef+Masc+Sg+Nom+Wk
ein	Det+Indef+Neut+Sg+NomAcc+Wk
höheres	Adj+Neut+Sg+NomAcc+St
Gehalt	Noun+Neut+Sg+NomAccDat
Gehalt	Noun+Masc+Sg+NomAccDat

(4)

verdienten	Adj+Fem+Sg+DatGen+Wk
verdienten	Adj+Masc+Sg+AccGen+StWk
verdienten	Adj+Masc+Sg+Dat+Wk
verdienten	Adj+Neut+Sg+Gen+StWk
verdienten	Adj+Neut+Sg+Dat+Wk
verdienten	Adj+FMN+Pl+NomAccDatGen+Wk
verdienten	Adj+FMN+Pl+Dat+St
Beamten	Noun+Masc+Sg+AccGen+StWk
Beamten	Noun+Masc+Sg+Dat+Wk
Beamten	Noun+Masc+Pl+NomAccDatGen+Wk
Beamten	Noun+Masc+Pl+Dat+St

In this case the set of contextually valid readings is still quite large since all adjectival readings that are compatible with the gender specification of the noun will be retained.

However, further pruning of contextually valid readings is possible. If noun phrases do not include an overt determiner, as in the case of the NP *verdienten Beamten*, then only strong forms are grammatical. Therefore, the only contextually valid readings for this noun phrase are provided by the sequence of morphological tags `Adj+Masc+Pl+Dat+St`, `Noun+Masc+Pl+Dat+St`.[8]

3 Guiding Dependency Parsing by Morphological Disambiguation

The preceding discussion was designed to provide an overview of some of the empirical issues that are involved in morphological disambiguation for a morphologically rich language like German. The present section will demonstrate the utility of morphological disambiguation for further incremental syntactic annotation.

Consider once again our example sentence in (1). The ultimate goal for syntactic annotation with the Xerox Incremental Deep Parsing System is to assign a dependency structure to the input sentence. As an intermediate stage, the input is chunked into major constituents. This chunked structure then serves as input to the dependency analysis. For sentence (1) the output of the chunker is as in (5).[9]

(5) {VF {NP Die Politiker}} {LK gaben} {MF {NP verdienten Beamten} {NP ein höheres Gehalt}}.

The chunked input identifies the three NPs whose grammatical relations need to be linked to the main verb *gaben*.[10]

The intended dependency structure output is shown in (6).

(6) {VF {NP#1 Die Politiker}} {LK#2 gaben} {MF {NP#3 verdienten Beamten} {NP#4 ein höheres Gehalt}}.

SUBJ(#2,#1), OBJ_dir(#2,#4), OBJ_indir(#2,#3)

In GRIP, the dependency analysis is constructed with the aid of lexical resources such as CELEX and IMS-LEX which provide subcategorization information for German verbs. A simplified entry of the information that CELEX provides for the lemma *geben* is shown in (7).

(7) geben +VERB+Aux_H+Acc_Comp+ +Dat_Comp+Comp_Subj

The key to identifying the correct dependency links is to try to match the case specifications inherent in the chunk analysis with the subcategorization information provided by the lexical

[8] The morphological tag `FMN` is disambiguated for the adjective as Masc due to the gender specification on the noun.

[9] Apart from NP chunking, the structure in (5) labels the topological fields of the clause: VF (for: *Vorfeld*), MF (for: *Mittelfeld*) and LK (for: *Linke Klammer*). Such a topological field analysis is useful for identifying the overall structure of the clause (see (Hinrichs et al., 2000a) and (Hinrichs et al., 2000b) for details.

[10] The node structure in (5) is grossly oversimplified in that it leaves out morphological information percolated up from the daughter nodes. How such morphological information is percolated to the phrasal nodes will be explained in detail in section 5

entry of the main verb. Here is where morphological disambiguation plays a crucial role. In the previous section, we discussed how co-occurring lexical nodes mutually constrain the set of contextually valid morphological interpretations. For the three NPs in sentence (1), the set of analyses shown in (8) will be retained.

(8)

Die	Det+Def+Fem+Sg+NomAcc+St
Politiker	Noun+Masc+Pl+NomAcc
verdienten	Adj+Masc+Pl+Dat+St
Beamten	Noun+Masc+Pl+Dat+St
ein	Det+Indef+Neut+Sg+NomAcc+Wk
höheres	Adj+Neut+Sg+NomAcc+St
Gehalt	Noun+Neut+Sg+NomAcc

The remaining ambiguities concern the case values of the NPs *die Politiker* and *ein höheres Gehalt*. Both can be either nominative or accusative case. Thus, in principle, both NPs can serve as either the subject or direct object of the finite verb. However, even this ambiguity can be resolved due to subject-verb agreement. Since the finite verb is plural, only the plural NP *die Politiker* can be the subject and the NP *ein höheres Gehalt* should be direct object.

What this example has shown is that morphological disambiguation in conjunction with other morphological constraints such as subject-verb agreement can effectively reduce the number of candidate readings and uniquely determine the dependency structure to be assigned. In the remainder of the paper we will discuss how the XIP System provides the necessary computational environment to efficiently carrying out morphological disambiguation.

4 Implementing Morphological Disambiguation

XIP provides two types of disambiguation rules: ordinary disambiguation rules, which can eliminate readings for a single lexical node on the basis of left and/or right contexts of the token, and double reduction rules which simultaneously reduce readings of sequences of tokens. We will discuss both types of rules in turn.

The general format for ordinary disambiguation rules is shown in (9).

(9) readings_filter = |left_context| selected_readings |right_context|.

The left side of the rule specifies which readings of lexical nodes the disambiguation rule should apply to. As the name suggests, the field *selected readings* will specify a proper subset of the readings that are specified in the field *readings filter*. The optional left and right context specifications constrain the environments under which the rule will apply. The effect of such a disambiguation rule can best be demonstrated by an example:

(10) det,pron = det |adj*, noun|.

The rule in (10) applies to lexical tokens which have determiner and pronoun readings and selects the determiner readings if the token is followed by any, including zero, number of adjectives and a noun. At the same time, pronominal readings will be eliminated in precisely this context.

A second disambiguation rule that is needed for reducing of the number of contextually valid readings for NPs in German is shown in (11).[11]

(11) noun[St,Wk] = |?[det:~], adj*| noun[St].

This rule eliminates all weak readings of nouns that have strong and weak forms if it is the head of an NP that does not contain a determiner. The rule will therefore eliminate the weak readings of the noun *Beamten* shown in (4) and retain only the reading `+Noun+Masc+Pl+Dat+St`.

While ordinary disambiguation rules reduce the contextually valid readings for a single lexical node, double reduction rules simultaneously reduce readings of sequences of tokens. The latter type of rule is therefore used for simplifying the candidate morphological analyses of lexical nodes that make up phrasal categories.

The general format for double reduction rules is shown in (12).

(12) |node_sequence| $\Rightarrow$ boolean_constraints.

[11] The rule shown in (11) is for illustrative purposes only and is therefore simplified. The actual rule used in the grammar takes into account a wider range of sequences of categories that can precede the noun.

(13) instantiates the double reduction rule schema to the disambiguation rule needed for German to eliminate all those readings of adjectives and nouns that do not match.

(13) |adj#1, noun#2| ⇒ #1[agr] :: #2[agr].

The condition on the right-hand side of the rule enforces strict identity of agreement features between adjective and noun, with agreement consisting of the gender, number and case features for each node. Therefore, the rule has the effect of eliminating all readings of adjective and noun sequences with conflicting agreement features.

The rule shown in (13) is a simplified version of the rule that is actually needed, since it applies only to noun phrases that contain a single adjective. The rule in (14) generalizes the previous rule to accommodate multiple occurrences of adjectives.

(14) |det#1, adj*, adj#2, adj*, noun| ⇒ (#1[agr] :: #2[agr]) & (#1[decl] ∼: #2[decl]).

The pattern matching algorithm of the XIP System will ensure non-deterministic application of the rule to each adjective that precedes a noun in a left-to-right fashion.

The rule in (15) accounts for the distinct declension class values required for contextually valid patterns of determiners and adjectives that we discussed in detail in section 2 above.[12]

(15) |det#1, adj#2, noun| ⇒ (#1[agr] :: #2[agr]) & (#1[decl] ∼: #2[decl]).

The rule illustrates another feature of the expressivity of double reduction rules in XIP: the constraint on the right-hand side of the double reduction rule may contain any combination of boolean operators (disjunction, conjunction and negation of features) that can be expressed in the system. To force distinctness of declension values the negated equality operator ∼: is used.

[12]Rule (15) requires strict identity of agreement features of determiner and noun and thus does not account for the special cases discussed in footnote 6 above, in which determiner and noun can disagree for certain agreement features. Thus rule (15) is a simplified version of an actual rule used by the grammar which narrows down the class of determiners to exclude, for example, relative pronouns.

5 Percolation of Morphological Features and Putting it all Together

As mentioned above, the dependency analysis takes as input the output of the chunk parser and tries to link nodes of the chunked tree by dependency relations. For example, chunked NPs are linked to the finite verb via grammatical relations such as subject, direct object and indirect object, depending on the morphological features present on the NP nodes. This section will explain how the contextually valid morphological analyses for the lexical nodes that make up an NP can be percolated up to the NP node during chunking.

The chunker uses non-recursive rewrite rules to combine the lexical nodes that make up an NP, after these nodes have been disambiguated by the use of ordinary disambiguation rules and double reduction rules. The desired percolation of morphological features onto the mother node is carried out by side conditions on the rewrite rules that are specified by boolean constraints analogous to those shown in the previous section for double reduction rules.

The resulting interaction between ordinary disambiguation rules, double reduction rules and chunking rules can best be illustrated by the chunk analysis for one of the NPs of our example sentence. Figure 1 shows the result of morphological disambiguation for the NP *verdienten Beamten*.

Lines marked with a minus sign are eliminated due to morphological disambiguation – either by the ordinary disambiguation rule (10) or by the double reduction rule (13). The double reduction rule also reduces the FMN gender value (shown in the corresponding line in (4)) for the remaining reading of *verdienten* to Masc. The chunker then combines adjective and noun into an NP and percolates the agreement features of the remaining contextually valid reading onto this mother node.

Since XIP allows the inclusion of features on non-terminal nodes for chunking rules, readers might wonder why narrowing down contextually valid readings has to be done prior to chunking by the special-purpose mechanism of double reduction rules and could not, instead, be done during chunking by appropriate Boolean constraints on chunking rules. However, the

verdienten	Adj+Fem+Sg+DatGen+Wk	-	(by rule 13)
verdienten	Adj+Masc+Sg+AccGen+StWk	-	(by rule 13)
verdienten	Adj+Masc+Sg+Dat+Wk	-	(by rule 13)
verdienten	Adj+Neut+Sg+Gen+StWk	-	(by rule 13)
verdienten	Adj+Neut+Sg+Dat+Wk	-	(by rule 13)
verdienten	Adj+FMN+Pl+NomAccDatGen+Wk	-	(by rule 13)
verdienten	Adj+Masc+Pl+Dat+St		(by rule 13)
Beamten	Noun+Masc+Sg+AccGen+StWk	-	(by rule 11)
Beamten	Noun+Masc+Sg+Dat+Wk	-	(by rule 11)
Beamten	Noun+Masc+Pl+NomAccDatGen+Wk	-	(by rule 11)
Beamten	Noun+Masc+Pl+Dat+St		

Figure 1: Morphological disambiguation for the NP *verdienten Beamten*

latter is beyond the functionality of chunking rules, which do not allow to eliminate readings when forming chunks on the basis of feature values. Notice also that most chunk parsers do not allow the introduction of features on non-terminals and require, instead, that all non-terminals are atomic symbols. For such chunk parsers, the only option would be to create distinct non-terminal symbols for each combination of agreement values, resulting in a proliferation of phrasal and lexical categories and accompanying rule sets. The functionality of double reduction rules for such chunk parsers would therefore be at least as desirable as for XIP in order to reduce the processing load of chunk parsing.

6 Quantitative Evaluation

The efficiency of the double reduction rules was tested on a corpus of noun phrases extracted from the taz newspaper corpus (taz, 1999). 500 noun phrases were extracted at random from the corpus and morphologically analyzed by the XRCE morphological analyzer for German. This analysis produced an average number of 6.237 distinct readings per token in the corpus before any ordinary disambiguation rules or double reduction rules were applied. After applying these disambiguation rules the average number of contextually valid readings reduced to 1.732 per token. This corresponds to a 72.23% reduction of readings per token.

6.1 Morphological Disambiguation

The number of distinct contextually valid readings per noun phrase after disambiguation is identical to the number of readings per token, since only readings shared among all tokens are kept. Thus, the average number of contextually valid readings per noun phrase is the same as that for tokens, i.e. 1.732. Figure 2 shows the distribution of the number of disambiguated readings over the entire corpus.

	count of NPs	percentage
1 reading	173	34.6%
2 readings	315	63.0%
3 readings	3	0.6%
$\geq$ 4 readings	9	1.8%

Figure 2: Results of morphological disambiguation

Thus, in more than 97 % of all cases, at most two readings are retained, with more than a third of all noun phrases being uniquely disambiguated.

6.2 Adding Syntactic Heuristics

For the set of 315 NPs that retain two valid analyses after morphological disambiguation, the syntactic environment in which they occur in the corpus helps to uniquely disambiguate more than two thirds of them. In the full paper, the full range of such syntactic heuristics will be presented. Due to space limitations, we mention only the most effective one here: more than half of the NPs with two readings involve nominative/accusative ambiguities. If these NPs occur within a prepositional phrase, the case required by the preposition will lead to further disambiguation. Since no preposition requires nominative, only accusative readings will be retained.

Figure 3 shows the results of applying the full set of syntactic heuristics to the set of 315 NPs identified in Figure 2.

	count of NPs	percentage
unique reading	223	68.20%
ambiguous gender	6	1.83%
nondisambiguated case	98	28,97%

Figure 3: Disambiguation with Syntactic Heuristics

The combination of syntactic heuristics and morphological disambiguation, thus, yield a total of 396 unique readings for the 500 NPs in the test corpus. Since gender ambiguities never play a role in the determination of dependency relations, the number of NPs that have a unique value for the determination of syntactic dependencies is actually slightly higher (402). Thus, in 80.4% of all cases, a unique reading can serve as input to the dependency parsing module of GRIP.

7 Conclusion

The present paper has shown that morphological disambiguation constitutes a crucial step in narrowing down the search space for the correct assignment of dependency structures. A quantitative evaluation on a German test corpus has shown that application of XIP disambiguation rules in conjunction with heuristics about the syntactic context of German NPs yield unique morphological analyses as input for assigning dependency relations in 80.4% of all cases. For those NPs that still have multiple readings, the lexical resources (CELEX and IMS-LEX), which are used by the dependency parsing module of GRIP and which give subcategorization information for German verbs, can provide further constraints for disambiguation. A quantitative analysis of such disambiguation at the level of dependency parsing itself will be the subject of future research.

References

S. Ait-Mokhtar, J.-P. Chanod, and C. Roux. 2002. Robustness beyond Shallowness: Incremental Deep Parsing. The NLE special issue on Robust Methods in Analysis of Natural Language Data (to appear).

D. Duchier. 1999. Axiomatizing Dependency Parsing Using Set Constraints. In: Sixth Meeting on the Mathematics of Language (MOL6).

The CELEX Lexical Database, Release 2 (CD-ROM). 1995.

J. Eckle-Kohler. 1999. Linguistisches Wissen zur automatischen Lexikon-Akqisition aus deutschen Textcorpora. Logos, Berlin.

J. Hajic. 1998. Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. In: Issues of Valency and Meaning. Karolinum, Praha. pp. 106-132.

E. Hinrichs, J. Bartels, Y. Kawata, V. Kordoni, and H. Telljohann. 2000a. The Verbmobil Treebanks. In: 5. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2000). Ilmenau. pp. 107 – 112.

E. Hinrichs, S. Kübler, V. Kordoni, and F. Müller. 2000b. Robust Chunk Parsing for Spontaneous Speech. In: W. Wahlster ed. Verbmobil: Foundations of Speech-to-Speech Translation. Springer Verlag, Berlin. pp. 163 – 182.

I. Mel'cuk. 1988. Dependency Syntax: Theory and Practice. New York: State University of New York Press.

P. Tapanainen and T. JÆrvinen. 1997. A non-projective dependency parser. : In: Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP'97). ACL, Washington, D.C. pp. 64–71.

taz – die Tageszeitung (CD-ROM). 1999. September 1986 – May 1999. www.taz.de.

A. Zwicky. 1986. German adjective agreement in GPSG. Linguistics, Vol. 24, 957–990.