Parsing German Text for Syntacto-semantic Structures

Christian Braun
Computational Linguistics, Saarland University
cbraun@coli.uni-sb.de

Abstract

For a range of NLP applications a deep linguistic analysis is not needed. Rather, a relatively shallow syntacto-semantic representation suffices to yield good results. In this paper, a modular parser for German text is presented that produces such a representation, which we call Partially Resolved Dependency Structures (PReDS).

1 Introduction

It is a computational-linguistics' commonplace that the full and in-depth analysis of free text still poses a daunting challenge. While it remains to be a worthy and strongly pursued goal it became also clear that for a number of NLP areas a more shallow approach to parsing might be sufficient. One of these areas is information retrieval in a broader sense, including information extraction and question answering. Various projects have attained impressive results here using systems where the syntactic processing and semantic representation is restricted to a varying degree.

In this paper we describe a modular and robust parser that produces a shallow semantic representation for free German text, called *Partially Resolved Dependency Structures* (PReDS)[1]. The goal was to attain a structure that is, on the one hand, sophisticated enough to facilitate reasoning beyond the mere level of surface realization; on the other hand, it needs to be simple enough to be produced reliably for large amounts of text from different domains.

The following sections describe the components of the parsing system (sec. 2) and the design of PReDS (sec. 3). In section 4, we outline an application scenario for the proposed NLP framework.

2 The Parsing Pipeline

The design of the processing chain that constitutes our parsing system is built around the notion of easy-first parsing. This concept was introduced by (Abney, 1996) and describes a pipelined NLP architecture where simple components are coupled together. Each component only performs actions that produce reliable results, pushing off the more difficult cases and trusting later components with more information to be able to solve those. This results in a very robust parsing system that produces outputs that are not necessarily complete, but reliable and easily utilizable. Our parsing system consists of the following four components: i) tokenization and morphology, ii) topological parser, iii) phrasal chunker, iv) PReDS builder. All components communicate via XML-based interface formats.

2.1 Tokenization and Morphology

Tokenization and morphology are performed in lockstep. The input is tokenized and sent to a comprehensive morphological analyzer. Note that we do not employ a Part-of-Speech tagger; rather, all possible morphological readings of a word are retained and channelled into the next component. This is in accordance with the notion of easy-first-parsing where each component only performs actions it is sure about.

2.2 Topological Parser

Parsing natural language is a highly complex task. One way to deal with this complexity is the divide-and-conquer strategy as proposed by (Shiuan and Ann, 1996), and its application to German text corpora in (Wauschkuhn, 1996). In this approach, the

[1]This work was performed in the project *Computational Linguistics and Language Technology for Real Life Applications* (COLLATE), funded by the German Ministry for Education and Research, Grant Number 01 IN A01 B.

Figure 1: Sentence topology

entirety of the problem is being broken down into smaller, more manageable parts. The topology of a German sentence is a perfect starting point for such a strategy since it is a very consistent and conceptually simple, yet effectiv way of structuring sentences.

Topological Grammar The general set of topological rules in German is quite limited (breaking up each sentence into up to five parts, namely Vorfeld, linke Klammer, Mittelfeld, rechte Klammer, Nachfeld; cf. fig. 1) and there are hardly any exceptions. Therefore, we use a compact, hand-coded context-free grammar for the topological parser. This differs from work reported in (Becker and Frank, 2002) where a German topological grammar is statistically induced from a large corpus. In contrast, we argue that the problem space is restricted enough to be dealt with efficiently by hand-coding the rules, thus skirting the notorious problems of a statistical approach such as maintainability and extensibility.

In order to keep the grammar to a reasonable size we enhanced it with a look-ahead mechanism. This grammar is processed by a Earley-style parser, implemented in Perl. The original version was obtained from a public repository[2], and then extensively modified.

The basic idea that the grammar is built on is to utilize a small set of reliable indicators (mainly verbs, subjunctions, conjunctions, verb prefixes) to figure out the topological structure; this idea was explored extensively and successfully in (Braun, 1999). Currently, the grammar consists of 273 rules, covering a wide range of constructions, including different types of verb complexes, various coordination constructions, cascaded subclauses, etc.

Evaluation In order to judge the performance of the topological parser we evaluated it on a previously unseen corpus comprised of 104 sentences from the German Civil Code (BGB) and 353 sentences taken from SÜDDEUTSCHE ZEITUNG (SZ) business reports. The time it took to parse this corpus was 100 seconds, which puts the parser's performance at roughly 4.5 sentences per second. For the evaluation we assumed that each sentence has exactly one correct topological structure; the results are then expressed in terms of two standard metrics, using *exact match* as correctness criterion:

Recall: number of found correct analyses relative to the total number of sentences.

Precision: number of found correct analyses relative to the total number of analyses found.

The results shown in Table 1 illustrate a couple of points. First, given their complex structure it is in general more difficult to parse legal language than the language used in newspapers; this is reflected by the difference between the numbers for the SZ corpus and the BGB corpus. But even for the legal sentences the parser reaches a performance level that is comparable to the stochastic parser by (Becker and Frank, 2002) who report a perfect-match accuracy of 80.4% on disambiguated POS sequences[3]. Second, it

[2]CPAN at http://search.cpan.org/

[3]Note, however, that for a truly meaningful comparison of

	Sentences					Trees			Recall	Precision
	Total	ASL	Wrong	Amb	Noa	Total	Correct	Wrong		
SZ	353	18	18	23	18	360	315	45	89.24	87.50
BGB	104	27	6	7	14	100	84	16	80.77	84.00
Total	457		24	30	32	460	399	61	**87.31**	**86.74**

ASL: average sentence length in tokens **Wrong: one, but wrong analysis**

Amb: more than one analysis **Noa: no analysis**

Table 1: Evaluation results for the topological parser

shows that structural ambiguity is not a big problem; while there are a number of sentences with more than one topological analysis, this number is small enough to be safely ignored in QA- and IE-applications. Interestingly enough, the two evaluation corpora do not differ in that respect: for both of them, sentences with more than one analysis account for about 6 percent of all of the sentences in the respective corpus.

All in all, the results justify our approach of a rule-based, manually designed grammar. Manual inspection of the error types strongly suggests that minor changes in the grammar should put the evaluation numbers onto the 90 % level, making the topological parser a very reliable component of the processing chain.

2.3 Phrase Chunking

Having the topological structures, it is very simple to extract the text of the different fields (i.e. Vorfeld, Mittelfeld, Nachfeld), and pass it on to the next stage in the processing chain, comprising a named entity recognizer and a NP/PP chunker. This open architecture obviously enables the easy addition of other phrase-level components. For named entity recognition we employ the extensive grammars for German NEs developed by the Collate IE group. For NPs and PPs, we employ a FSA-based chunker that processes a hand-built grammar and is capable of recognizing self-embedding phrase-structures; it is described in detail in (Fliedner, 2002). The result of the chunker is a list of phrases, which are merged back into the topological structure. We call the result *Extended Topological Structures* (XTS). These serve as input for the next component, the PReDS builder.

the approaches a detailed examination of the targeted output structures would be needed first.

3 PreDS

PReDS (*Partially Resolved Dependency Structures*), which are inspired by Microsoft's *Logical Form* (Elworthy, 2000) and are close to *Tectogrammatical Structures* (Böhmová et al., 2001) and *Grammatical Relations* (Carroll and Briscoe, 2001), serve as interface between the robust, domain-independent linguistic analysis and the more fine-grained processing of subsequent modules. We designed it as a shallow syntacto-semantic representation where linguistic completeness and precision is traded for reliability and unambiguousness for the sake of usefulness in NLP applications.

Conceptually, PReDSs are predicate-argument structures that are close to the syntactic surface structure but with semantically motivated modifications, for example showing the deep subject in passive constructions, and removing auxiliary and modal verbs into the feature set associated with nodes in the dependency structure. Technically, PReDS are conceptual graphs that are derived by recursively traversing the XTS trees. Using graphs allows for modelling reentrancy phenomena like relative clauses or clausal complements in raising/control constructions.

An important cornerstone of the PReDS design is the notion of underspecification: role assignment and modification relations are only carried out if there is sufficient evidence available in the XTS. Otherwise, if more linguistic or contextual knowledge is required, the PReDS builder resorts to using default labels and the strategy of low attachment. We chose low attachment as default since following components that might want to reconsider the attachment decisions simply need to move the attached node up along the tree to find other possible attachment points.

One example for this strategy is a situation

```
mitteilen[+sg, +past, +want]
   --DSub-->Carrier[+masc, +sg, +def]
               --Mod-->belgisch
               --PPModDef-->auf
                              --Arg-->Pressekonferenz
   --ClArg-->verhandeln[+pres, +pl]
                --DSub-->und
                              --Coor-->Airline[+def, +fem, +sg]
                                          --Mod-->belgisch
                              --Coor-->Luxair[+Nameexpr, +sg ... ]
                --PPMod-->ueber
                            --Arg-->Zusammenschluss[+sg ...]
```

Figure 2: PReDS for the sentence *Die belgische Airline und Luxair wollen über einen Zusammenschluss verhandlen, teilte der belgische Carrier auf einer Pressekonferenz mit.* (*The Belgian airline and Luxair want to negotiate about a merger, the Belgian carrier announced during a press conference.*)

where morpho-syntactic ambiguity makes it impossible to assign the grammatical roles of subject, object, and indirect object. In that case, the default label `NPArg` is used.

Another example are PPs, as illustrated in Fig. 2; obviously, calculating the correct PP-attachment is beyond the scope of XTS's linguistic potential; therefore PPs are always attached as low as possible, i.e. to the next NP or verb, using the label `PPModDef`. If, however, there is reliable evidence that a PP belongs to a specific NP or verb (for example, because it follows a NP in the Vorfeld), then the label `PPMod` is used. Note that currently we don't have subcat information available; therefore, we do not differentiate between PPs as objects or adjuncts, and rather assign indiscriminately the label `PPMod`.

4 Future Work

The work reported in this paper is still in progress. The PReDS builder especially still lacks support for a range of linguistic constructions. In order to demonstrate the value and viability of the proposed NLP framework for QA and IE purposes we will use it for an IE task in the legal domain: extracting definitions from statutes and verdicts. The legal domain seems to be a very suitable testbed for NLP-based information management strategies because it provides i) large amounts of electronically available high-quality texts, ii) a closed and yet complicated domain, which allows for refined QA and IE strategies, and iii) a need in the real world

Having central legal terms and their definitions at its disposal, will benefit a QA system in the legal domain tremendously. First, for certain questions highly targeted information can be returned; second, using term hierarchies enables us to employ simple inference mechanisms to answer questions; third, if the extraction strategies prove to be reliable across different areas of the legal domain, then we envision the possibility of building up domain-specific ontologies automatically.

References

S. Abney. 1996. Partial parsing via finite-state cascades. In *Proceedings of the ESSLLI '96 Robust Parsing Workshop*.

M. Becker and A. Frank. 2002. A stochastic topological parser for German. In *Proceedings of COLING 2002*.

A. Böhmová, J. Hajič, E. Hajičová, and B. Hladká. 2001. The Prague Dependency Treebank: Three-level annotation scenario. In Anne Abeillé, editor, *Treebanks: Building and Using Syntactically Annotated Corpora*. Kluwer Academic Publishers.

C. Braun. 1999. Flaches und robustes Parsen deutscher Satzgefüge. Master's thesis, Saarland University.

J. Carroll and T. Briscoe. 2001. High precision extraction of grammatical relations. In *Proceedings of the 7th international workshop on parsing technologies*, Beijing.

D. Elworthy. 2000. Question answering using a large NLP system. In *Proceedings of the Ninth Text REtrieval Conference (TREC 2000)*.

G. Fliedner. 2002. A system for checking NP agreement in German texts. In *Student Workshop. Proceedings of the 40th Annual Meeting of the ACL*.

P. L. Shiuan and C. T. Ann. 1996. A Divide-and-Conquer Strategy for parsing. In *Proceedings of ACL/SIGPARSE, Punctuation in Computational Linguistics*.

O. Wauschkuhn. 1996. Ein Werkzeug zur partiellen syntaktischen Analyse deutscher Textkorpora. In *Proceedings of the Third KONVENS Conference*.