Knowledge-Poor Grammatical Function Assignment for German

Beata Kouchnir
Department of Computational Linguistics
Tübingen University
72074 Tübingen, Germany
kouchnir@sfs.uni-tuebingen.de

Abstract

We present a knowledge-poor memory-based approach to the task of grammatical function assignment for written German text. Two major challenges are the relatively free word order and morphological ambiguity inherent to the language. Nevertheless, we prove that good results can be achieved by using lexical and syntactic information only. Furthermore, we show that the error rate can be considerably reduced by fine-tuning the parameter setting of the learning algorithm, whereas stacking classifiers yielded only marginal improvement.

1 Introduction

Grammatical functions are an important component of the syntax-semantics interface, and a valuable resource for applications such as question answering, text summarization and information extraction. For English, many successful systems have been implemented using only syntactic features (Buchholz, 2002; Blaheta and Charniak, 2000; Carroll and Briscoe, 2001). The same approach, however, cannot be applied without modifications to other languages, especially ones with a relatively free word order and high morphological ambiguity.

(Müller, 2004), one of the few works on grammatical functions for German achieves competitive results but relies heavily on external resources that provide morphological and semantic information. As these resources are expensive to annotate and maintain, it is desirable to find an alternative approach to this task

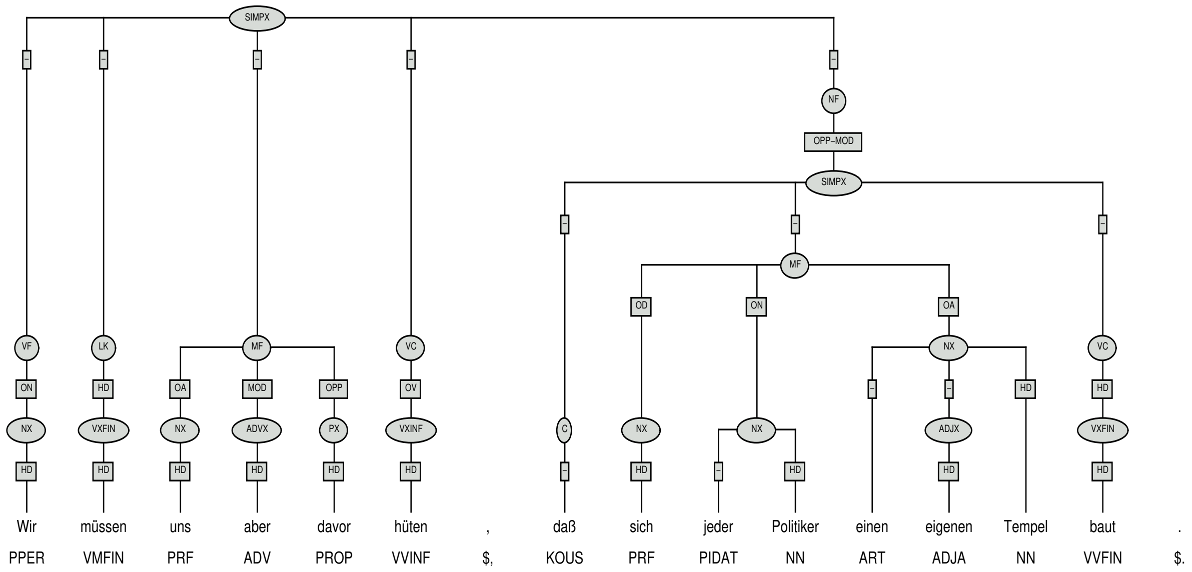
In this paper, we propose a knowledge-poor approach to assigning complement and modifier functions to German text. We show that with optimization of the learning parameters, such a system can achieve competitive results.

2 Resources

Our source of data is the Tübingen Treebank of Written German (Tüba-D/Z) (Telljohann et al., 2003), a collection of some 15,000 sentences from German newspaper *die tageszeitung*. It has both a deep and a shallow level of syntactic annotation. In addition to the generic constituent structure, it is annotated with topological fields (Höhle, 1986). Figure 1 shows an example of an annotated sentence.

The concept of topological fields is based on the position of the finite verb, which forms a sentence bracket that divides a sentence into an initial field, a middle field, and a final field. While the word order in German is fairly free, some types of constituents are restricted to certain fields. Thus, topological fields create a regularity which a constituent structure cannot provide.

The grammatical functions are annotated as edge labels of the constituents attached to the topological fields. The main distinction is between complements and modifiers. As crossing brackets are not allowed in Tüba-D/Z, the modifier edge labels are the only means to denote non-local de-



Wir müssen uns aber davor hüten, dass sich jeder Politiker einen eigenen Tempel baut.

We must us but thereof beware, that himself every politician one own temple builds.

'We must be careful not to let every politician build his own temple.'

Figure 1: A sample TüBa-D/Z tree.

pendencies. Table 1 shows a description of each grammatical function.

2.1 Feature Representation

In keeping with a knowledge-poor approach, we restrict ourselves to the information that can be derived from the corpus annotation, avoiding external resources. At this stage, we only use explicit (i.e lexical and syntactic) information and do not apply any heuristics to derive morphological or semantic features. The following are the features used by our system:

- **Lexical head** of the focus constituent. This is important for prepositional phrases, where the lexical head is not the syntactic head. We also use the determiners of noun and prepositional phrases, and prepositions of the latter. In German, both nouns and determiners can be morphologically ambiguous; their combination however, can sometimes resolve the ambiguity. Furthermore, some determiners are merged with prepositions, thus giving them morphological information
- **Part of speech** of the lexical head, determiner and preposition, where applicable.
- **Constituent type**: NX (noun phrase), PX (prepositional phrase), ADJX (adjectival phrase), ADVP (adverbial phrase) or SUB (subordinate clause).
- **Topological field**: VF (initial), MF (middle), NF (final), LV (resumptive construction) or C (complementizer).
- The **target verb**: both parts of the verb construction.
- **Context**: the features **head**, **POS**, **constituent** and **field** for the constituents immediately preceding and follow the focus constituents.

In addition, we can easily determine the sentence type by examining the sequence of constituents. A sentence is verb-initial if a finite verb is the first element (ignoring complementizers, coordinating particles and resumptive constructions), verb-second if it is the second element, and verb-final otherwise.

Function	Description
Complements	
ON	nominative object (subject)[1]
OA	accusative object
OD	dative object
OG	genitive object
OS	sentential object
OPP	prepositional object
OADJP	adjectival object
OADVP	adverbial object
PRED	predicative
FOPP	optional prepositional object
Modifiers	
MOD	ambiguous modifier
ON-MOD, OA-MOD, OD-MOD, OG-MOD, OS-MOD, OPP-MOD, OADJP-MOD, OADVP-MOD, PRED-MOD, FOPP-MOD, MOD-MOD, V-MOD	modifiers modifying complements or modifiers e.g. V-MOD = modifier of the verb

Table 1: Description of the grammatical functions in Tüba-D/Z

3 Memory-Based Learning

The concept of Memory-Based Learning (MBL) (Lin and Vitter, 1994) is to classify unseen (test) instances based on their similarity to known (training) instances. In practice, this is done by storing all training in memory without abstraction and computing the similarity between new and old examples based on a distance metric. New instances are then assigned the most frequent class within a set of *k* most similar examples (*k*-nearest neighbors).

Memory-based learning algorithms have proven to be effective for several NLP tasks, including named entity recognition (Meulder and Daelemans, 2003), part-of-speech tagging(Zavrel and Daelemans, 1999), word sense disambiguation(Veenstra et al., 2000) and shallow parsing (Daelemans et al., 1999). It has also been successfully applied to the task of grammatical function assignment for English (Buchholz, 2002).

The experiments described in this paper are conducted with the TiMBL[2] software package, the most commonly used MBL implementation. By including a variety of representation, weighing and classification options, TiMBL allows "tailoring" an algorithm to the needs of a particular classification tasks. A detailed description of the parameters is supplied by (Daelemans et al., 2003).

For this task, we tested four different feature weighting methods (none, Gain Ration, Information Gain, Chi-squared (χ^2), Shared Variance), two distance metrics (Overlap and Modified Value Distance) and five values for *k* (1, 3, 5, 7, 9). Furthermore, once we had determined the optimal parameter setting for a single classifier, we used it as a base learner for a stack.

4 Evaluation

The first experiment was to determine the optimal feature weighing method. Each method was tested in combination with Overlap, which is the default distance metric. The results are summarized in Table 2. It is surprising that no weighing outperforms both Chi-squared and Shared Variance. It is also by far the fastest, and thus a good choice when speed is a priority. For this task, the default weighting (Gain Ratio) proves to be the most accurate.

For distance metrics, however, MVDM significantly outperforms Overlap (see Table **??**). Larger values of *k* additionally improve performance. To

[2]http://ilk.kub.nl/software.html

Feature Weighting	P (%)	R	$F_{\beta=1}$
None	70.93	72.15	71.32
Gain Ratio	72.26	72.85	**72.48**
Information Gain	72.19	72.66	72.35
Chi-squared(χ^2)	67.22	67.89	67.46
Shared Variance	70.18	70.76	70.40

Table 2: Results for different feature weighting methods

k	P (%)	R (%)	$F_{\beta=1}$
1	73.79	74.40	74.01
3	75.86	76.70	75.90
5	76.51	77.30	76.36
7	76.78	77.54	**76.52**
9	76.78	77.43	76.47

Table 3: Results for different values of *k* in combination with MVDM

an extent, the same phenomenon is observed by (Buchholz, 2002) and (Daelemans et al., 2003). However, (Daelemans et al., 2003) also note that MVDM is sensitive to data sparsity, which might rule it out for many NLP applications. Although parameter optimization considerably reduces the error rate (15.3%), it also slows down classification. While an analysis of speed/performance tradeoff is beyond the scope of this paper, it is an important issue for real-time applications.

Stacking, i.e. using the output of one classifier as an additional feature for another, brought only a negligible improvement (see Table 4). This implies that the optimal setting for the first classifier might not be the best for the second classifier, but it is not feasible to determine a new setting for each one.

For a detailed error analysis we examined the precision and recall scores for each class. Table 5 summarizes the results, also showing the respective frequency. With a few exceptions, the more frequently a grammatical function occurs in the training set, the more accurately it is classified. Rarely occurring functions are virtually impossible to classify.

Considering that virtually all nouns and many determiners have the same surface representation for the nominative and accusative cases, the performance for the two respective functions is en-

Setting	P (%)	R (%)	$F_{\beta=1}$
Default (Overlap, *k*=1)	72.26	72.85	72.48
Best single (MVDM, *k*=7)	76.78	77.54	**76.52**
Stack (MVDM, *k*=7)	76.95	77.71	**76.69**

Table 4: Summary of results for different parameter settings

couraging. So is the ability to distinguish between ambiguous and verb modifiers. For applications that require a less fine-grained distinctions between different types of complements or modifiers, this system is competitive.

5 Conclusion and Future Work

We have shown that a knowledge-poor approach to German grammatical function assignment can achieve good results, by exploiting different parameter options of a memory based learner. Even for languages with a relatively free word order and high morphological ambiguity, a system can do without morphological and semantic features, which are more expensive to obtain that syntactic ones.

In the future, we would like to experiment with different machine learning algorithms and examine how they perform on different classes. It is likely that conceptually different classifiers will make different errors, in which case a combination by voting might yield further improvements. Furthermore, we would like to apply this approach to a more realistic setting, i.e. to automatically tagged and parsed/chunked data.

References

Don Blaheta and Eugene Charniak. 2000. Assigning function tags to parsed text. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 234–240, Seattle.

Sabine Buchholz. 2002. *Memory-based grammatical relation finding*. Ph.D. thesis, Tilburg University.

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

Label	#	P (%)	R (%)	$F_{\beta=1}$
Overall	73209	76.95	77.71	76.69
ON	20967	84.38	90.28	87.23
OA	10130	77.19	80.08	78.60
OD	1516	88.23	33.64	48.71
OG	28	0.00	0.00	0.00
OS	2010	71.58	79.70	75.42
OPP	2710	64.11	60.18	62.09
OADJP	175	44.12	17.14	24.69
OADVP	61	50.00	1.64	3.17
PRED	4071	78.64	72.27	75.32
FOPP	2303	62.75	32.48	42.80
MOD	13734	81.22	81.03	81.12
ON-MOD	850	43.32	20.59	27.91
OA-MOD	656	46.41	35.52	40.24
OD-MOD	46	0.00	0.00	0.00
OG-MOD	4	0.00	0.00	0.00
OS-MOD	64	0.00	0.00	0.00
OPP-MOD	190	62.14	45.79	52.73
OADJP-MOD	5	0.00	0.00	0.00
OADVP-MOD	4	0.00	0.00	0.00
PRED-MOD	314	44.12	33.44	38.04
FOPP-MOD	108	56.25	16.67	25.71
MOD-MOD	219	64.89	27.85	38.98
V-MOD	13044	71.55	81.61	76.25

Table 5: Detailed evaluation of grammatical function assignment (based on 10-fold cross-validation)

Walter Daelemans, Sabine Buchholz, and Jorn Veenstra. 1999. Memory-based shallow parsing. In *Proceedings of CoNLL-99*, Bergen.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2003. TiMBL: Tilburg memory based learner, version 5.0, reference guide. Technical report, Tilburg University.

Timan Höhle. 1986. Der Begriff 'Mittelfeld', Anmerkungen über die Theorie der topologischen Felder. In *Akten des Siebten Internationalen Germanistenkongresses 1985*, pages 329–340.

Jyh-Han Lin and Jeffrey Scott Vitter. 1994. A theory for memory-based learning. *Machine Learning*, 17:1–26.

Fien De Meulder and Walter Daelemans. 2003. Memory-based named-entity recognition using unannotated data. In *Proceedings of CoNLL-2003*, pages 208–211.

Frank Henrik Müller. 2004. Annotating grammatical functions for German using finite-state cascades. Submitted to ROMAND 2004.

Heike Telljohann, Erhard W. Hinrichs, and Sandra Kübler. 2003. Stylebook for the Tübingen Treebank of Written German (Tüba-D/Z). Technical report, Seminar für Sprachwissenschaft, Universität Tübingen.

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

Jakub Zavrel and Walter Daelemans. 1999. Recent advances in memory-based part-of-speech tagging. In *VI Simposio Internacional de Comunicacion Social*, pages 590–597, Santiago de Cuba.