Disambiguation of Morphological Structure using a PCFG

Helmut Schmid
Institute for Natural Language Processing (IMS)
University of Stuttgart
Germany
schmid@ims.uni-stuttgart.de

Abstract

German has a productive morphology and allows the creation of complex words which are often highly ambiguous. This paper reports on the development of a head-lexicalized PCFG for the disambiguation of German morphological analyses. The grammar is trained on unlabeled data using the Inside-Outside algorithm. The parser achieves a precision of more than 68% on difficult test data, which is 23% more than the baseline obtained by randomly choosing one of the simplest analyses. Remarkable is the fact that precision drops to 52% without lexicalization.

1 Introduction

German words may be as complex as the following title of a bill: *Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz* (law for the transfer of the task of controlling the labeling of beef). The complexity is due to the productive morphological processes of derivation (e.g. *Etikettierung* (labeling) = *Etikett* (label) + *ier* (derivational suffix) + *ung* (nominalization suffix)) and compounding (e.g. *Rindfleisch* (beef) = *Rind* (cattle) + *Fleisch* (meat)). For many words, there is more than one possible analysis. The German SMOR morphology (Schmid et al., 2004) e.g. generates 24 analyses for the word *Abteilungen*. If differences in the case feature are ignored, there are still six analyses, all of them plural:

- *Abt* (abbot) *Ei* (egg) *Lunge* (lung) *n* (plural inflectional ending)
- *Abt* (abbot) *ei* (abbot → abbey) *Lunge* (lung) *n* (plural inflectional ending)
- *Abt* (abbot) *eil* (hurry) *ung* (nominalization suffix) *en* (plural inflectional ending)
- *Abtei* (abbey) *Lunge* (lung) *n* (plural inflectional ending)
- *Abteilung* (department) *en* (plural inflectional ending)
- *ab* (separable verb prefix) *teil* (divide) *ung* (nominalization suffix) *en* (plural inflectional ending)

Here – and in many other cases, as well – the least complex analysis (defined as the number of derivation and compounding steps), namely the plural of *Abteilung* (department), is the correct one. This heuristic is not always successful, however. The word *Reisende* e.g. is analyzed as the compound of *Reis* (rice) + *Ende* (end), and alternately as the nominalization of the present participle *reisend* (traveling). The latter one is correct although it requires two derivational steps (formation of the participle plus nominalization), while the former requires only one compounding step.

The least complex analysis is not necessarily unique. One reason is, that German word forms are often ambiguous wrt. number, gender and case. The adjective *kleine* (small) e.g. receives 7 analyses by SMOR which differ only in the agreement features.

Another reason is that word forms are ambiguous wrt. the part of speech. The word *gerecht* e.g. is either an adjective (fair) or the past participle of the verb *rechen* (to rake). Similarly, the word *gerade* is either an adjective (straight) or an adverb (just). These ambiguities can be resolved based on the context e.g. with a part-of-speech tagger.

Other types of ambiguities are not disambiguated by the syntactic context because the morphosyntactic features are invariant. Compounds with three elements like *Sonderpreisliste*, for instance, are systematically ambiguous between a left-branching (list of special prices) and a right-branching structure (special price-list), but unambiguous regarding their part of speech and agreement features. Some word forms like *Schmerzfreiheit* (absence of pain) can either be analyzed as derivations (schmerzfrei-heit – "painless-ness") or as compounds (Schmerz-freiheit – "pain freedom"). Again, there is no difference in the morphosyntactic features. A further source of ambiguity are the stems: Consider the word *Mittelzuweisung* (allocation of resources). The compounding stem *mittel* could either originate from the adjective *mittel* (average) or from the noun *Mittel* (means). All these ambiguities are not resolvable by the syntactic context because their syntactic properties are identical. However, most of these words have a preferred reading. *Nah verkehrs zug* (commuter train) e.g. is likely to have a left-branching structure, whereas the correct analysis of *Computer bild schirm* (computer monitor) is right-branching.

Considering these features of German morphology, the following disambiguation strategy for morphological ambiguities is proposed: Frequent words should be manually disambiguated and the correct analysis/analyses should be explicitly stored in the lexicon. Ambiguities involving different morphosyntactic features should be resolved by a tagger or parser. The elimination of the remaining ambiguities, namely ambiguities of rare words which are not reflected by the morphosyntactic features, requires a different method. A general strategy is to generate the set of possible analyses, to rank them according to some criterion and to return the best analysis (or analyses). One very simple ranking criterion is the complexity of the analysis e.g. measured by the number of derivational and compounding steps. We will use this criterion as a baseline to which we compare our method.

Given an FST-based morphological analyzer and a training corpus consisting of manually disambiguated analyses, it is also possible to estimate transition probabilities for the finite state transducer and to disambiguate by choosing the most probable path through the transducer network for a given word. A drawback of this approach is the limitation of the type of analyses that finite state transducers can generate. A finite state transducer maps a regular language, the set of word forms, to another regular language, the set of analyses. Therefore it is not able to produce structured analyses as shown in figure 1 (for arbitrary depths). It also fails to represent non-local dependencies, like the one between *Vertrag* (contract) and *Lösung* (solution) in the second analysis of figure 1.

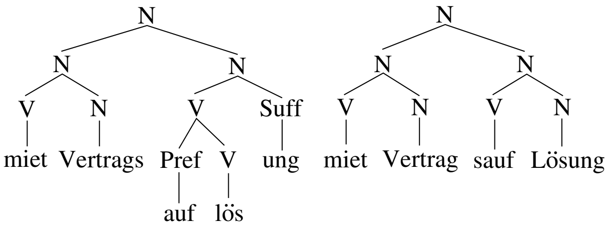

Figure 1: Two morphological analyses of the German word *Mietvertragsauflösung* (leasing contract cancellation); the first one is correct.

Given the limitations of weighted finite-state transducers, we propose to use a more powerful formalism, namely head-lexicalized probabilistic context-free grammars (Carroll and Rooth, 1998; Charniak, 1997) to rank the analyses. Context-free grammars have, of course, no difficulties to generate the analyses shown in figure 1. By assigning probabilities to the grammar rules, we obtain a probabilistic context-free grammar (PCFG) which allows the parser to distinguish between frequent and rare morphological constructions. Nouns e.g. are much more likely to be compounds than verbs. In head-lexicalized PCFGs (HL-PCFGs), the probability of a rule also depends on the lexical head of the constituent. HL-PCFGs are therefore able to learn that nouns headed by *Problem* (problem) are more likely to be compounds (e.g. *Schulprobleme* (school problems)) than nouns headed by *Samstag* (Saturday). Moreover, HL-PCFGs represent lexical dependencies like that between *Vertrag* and *Lösung* in figure 1.

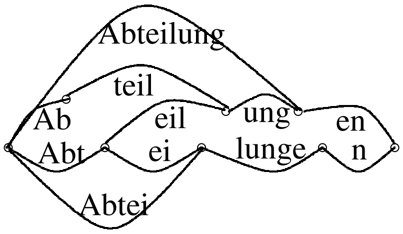

Figure 2: Morpheme lattice

In this paper, we present a HL-PCFG-based disambiguator for German. Using the SMOR morphological analyzer, the input words are first split into morpheme sequences and then analyzed with a HL-PCFG parser. Due to ambiguities, the parser's input is actually a lattice rather than a sequence (see the example in figure 2).

The rest of the paper is organized as follows: In section 2, we briefly review head-lexicalized PCFGs. Section 3 summarizes some important features of SMOR. The development of the grammar will be described in section 4. Section 5 explains the training strategy, and section 6 reports on the annotation of the test data. Section 7 presents the results of an evaluation, section 8 comments on related work, and section 9 summarizes the main points of the paper. Finally, section 10 gives an outlook on future work.

2 Head-Lexicalized PCFGs

A *head-lexicalized parse tree* is a parse tree in which each constituent is labeled with its category and its *lexical head*. The lexical head of a terminal symbol is the symbol itself and the lexical head of a non-terminal symbol is the lexical head of its (unique) head child.

In a head-lexicalized PCFG (HL-PCFG) (Carroll and Rooth, 1998; Charniak, 1997), one symbol on the right-hand side of each rule is marked as the head. A HL-PCFG assumes that (i) the probability of a rule depends on the category **and the lexical head** of the expanded constituent and (ii) that the lexical head of a non-head node depends on its own category, and the category and the lexical head of the parent node. The probability of a head-lexicalized parse tree is therefore:

$$p_{start}(cat(root)) \;\; p_{start}(head(root)|cat(root))*$$
$$\prod_{n \in N} p_{Rule}(rule(n)|cat(n), head(n))*$$
$$\prod_{n \in A} p_{Head}(head(n)|cat(n), pcat(n), phead(n))$$

where

$root$ is the root node of the parse tree
$cat(n)$ is the syntactic category of node n
$head(n)$ is the lexical head of node n
$rule(n)$ is the grammar rule which expands node n
$pcat(n)$ is the syntactic category of the parent of n
$phead(n)$ is the lexical head of the parent of n

HL-PCFGs have a large number of parameters which need to estimated from training data. In order to avoid sparse data problems, the parameters usually have to be smoothed. HL-PCFGs can either be trained on labeled data (supervised training) or on unlabeled data (unsupervised training) using the Inside-Outside algorithm, an instance of the EM algorithm. Training on labeled data usually gives better results, but it requires a treebank which is expensive to create. In our experiments, we used unsupervised training with the LoPar parser which is available at http://www.ims.uni-stuttgart.de/projekte/gramotron/SOFTWARE/LoPar-en.html.

3 SMOR

SMOR (Schmid et al., 2004) is a German FST-based morphological analyzer which covers inflection, compounding, and prefix as well as suffix derivation. It builds on earlier work reported in (Schiller, 1996) and (Schmid et al., 2001).

SMOR uses features to represent derivation constraints. German derivational suffixes select their base in terms of part of speech, the stem type (derivation or compounding stem)[1], the origin (native, classical, foreign), and the structure (simplex, compound, prefix derivation, suffix derivation) of the stem which they combine with. This information is encoded with features. The German derivation suffix *lich* e.g. combines with a simplex derivation stem of a native noun to form an adjective. The feature constraints of *lich* are therefore (1) part of speech = NN (2) stem type = deriv (3) origin = native and (4) structure = simplex.

[1] Suffixes which combine with compounding stems historically evolved from compounding constructions.

4 The Grammar

The grammar used by the morphological disambiguator has a small set of rather general categories for prefixes (P), suffixes (S), uninflected base stems (B), uninflected base suffixes (SB), inflectional endings (F) and other morphemes (W). There is only one rule for compounding and prefix and suffix derivation, respectively, and two rules for the stem and suffix inflection. Additional rules introduce the start symbol TOP and generate special word forms like hyphenated (*Thomas-Mann-Straße*) or truncated words (*Vor-*). Overall, the base grammar has 13 rules. Inflection is always attached low in order to avoid spurious ambiguities. The part of speech is encoded as a feature.

Like SMOR, the grammar encodes derivation constraints with features. Number, gender and case are not encoded. Ambiguities in the agreement features are therefore not reflected in the parses which the grammar generates. This allows us to abstract away from this type of ambiguity which cannot be resolved without contextual information. If some application requires agreement information, it has to be reinserted after disambiguation.

The feature grammar is compiled into a context-free grammar with 1973 rules. In order to reduce the grammar size, the features for origin and complexity were not compiled out. Figure 3 shows a compounding rule (building a noun base stem from a noun compounding stem and a noun base stem), a suffix derivation rule (building an adjective base stem from a noun derivation stem and a derivation suffix), a prefix derivation rule (prefixing a verbal compounding stem) and two inflection rules (for the inflection of a noun and a nominal derivation suffix, respectively) from the resulting grammar. The quote symbol marks the head of a rule.

W.NN.base → W.NN.compound W.NN.base'
W.ADJ.base → W.NN.deriv S.NN.deriv.ADJ.base
W.V.compound → P.V W.V.compound'
W.NN.base → B.NN.base' F.NN
S.ADJ.deriv.NN.base → SB.ADJ.deriv.NN.base' F.NN

Figure 3: Some rules from the context-free grammar

The parser retrieves the categories of the morphemes from a lexicon which also contains information about the standard form of a morpheme. The representation of the morphemes returned by the FST-based word splitter is close to the surface form. Only capitalization is taken over from the standard form. The adjective *ursächlich* (causal), for instance, is split into *Ursäch* and *lich*. The lexicon assigns to *Ursäch* the category *W.NN.deriv* and the standard form *Ursache* (cause).

5 PCFG Training

PCFG training normally requires manually annotated training data. Because a treebank of German morphological analyses was not available, we decided to try unsupervised training using the Inside-Outside algorithm (Lari and Young, 1990). We worked with unlexicalized as well as head-lexicalized PCFGs (Carroll and Rooth, 1998; Charniak, 1997). The lexicalized models used the standard form of the morphemes (see the previous section) as heads.

The word list from a German 300 million word newspaper corpus was used as training data. From the 3.2 million tokens in the word list, SMOR successfully analyzed 2.3 million tokens which were used in the experiment. Training was either type-based (with each word form having the same weight) or token-based (with weights proportional to the frequency). We experimented with uniform and non-uniform initial distributions. In the uniform model, each rule had an initial frequency of 1 from which the probabilities were estimated. In the non-uniform model, the frequency of two classes of rules was increased to 1000. The first class are the rules which expand the start symbol TOP to an adjective or adverb, leading to a preference of these word classes over other word classes, in particular verbs. The second class is formed by rules generating inflectional endings, which induces a preference for simpler analyses.

6 Test Data

The test data was extracted from a corpus of the German newspaper *Die Zeit* which was not part of the training corpus. We prepared two different test corpora. The first test corpus (data1) consisted of 425 words extracted from a randomly selected part of the corpus. We only extracted words with at least one letter which were ambiguous (ignoring ambiguities in number, gender and case) and either nouns, verbs or adjectives and not from the beginning of a sentence. Duplicates were retained. The words were parsed and manually disambiguated. We looked at the context of a word, where this was necessary for disambiguation. Words without a correct analysis were deleted.

In order to obtain more information on the types of ambiguity and their frequency, 153 words were manually classified wrt. the class of the ambiguity. The following results were obtained:

- 39 words (25%) were ambiguous between an adjective and a verb like *gerecht* - "just" (adjective) vs. past participle of *rechen* (to rake).
- 28 words (18%) were ambiguous between a noun and a proper name like *Mann* - "man" vs. *Thomas Mann*
- 19 words (12%) were ambiguous between an adjective and an adverb like *gerade* - "straight" vs. "just" (adverb)
- 14 words (9%) showed a complex ambiguity involving derivation and compounding like the word *überlieferung* (tradition) which is either a nominalization of the prefix verb *überliefern* (to bequeath) or a compound of the stems *über* (over) and *Lieferung* (delivery).
- 13 words (8%) were compounds which were ambiguous between a left-branching and a right-branching structure like *Welt rekord höhe* (world record height)
- In 10 words (7%), there was an ambiguity between an adjective and a proper name or noun stem - as in *Höchstleistung* (maximum performance) where *höchst* can be derived from the proper name *Höchst* (a German city) or the superlative *höchst* (highest)
- 6 words (4%) showed a systematic ambiguity between an adjective and a noun caused by adding the suffix *er* to a city name, like *Moskauer* - "Moskau related" vs. "person from Moskau"
- Another 6 words (4%) were ambiguous between two different noun stems like *Halle* which is either singular form of *Halle* (hall) or the plural form of *Hall* (reverberation)

The second set of test data (data2) was designed to contain only infrequent words which were not ambiguous wrt. part of speech. It was extracted from the same newspaper corpus. Here, we excluded words which were (1) sentence-initial (in order to avoid problems with capitalized words) (2) not analyzed by SMOR (3) ambiguous wrt. part of speech (4) from closed word classes or (5) simplex words. Furthermore, we extracted only words with more than one simplest[2] analysis, in order to make the test data more challenging. The extracted words were sorted by frequency and a block of 1000 word forms was randomly selected from the lower frequency range. All of them had occurred 4 times. We focussed on rare words because frequent words are better disambiguated manually and stored in a table (see the discussion in the introduction).

The 1000 selected word forms were parsed and manually disambiguated. 193 problematic words were deleted from the evaluation set because either (1) no analysis was correct (e.g. *Elsevier*, which was not analyzed as a proper name) or (2) there was a true ambiguity (e.g. *Rottweiler* which is either a dog breed or a person from the city of Rottweil or (3) the lemma was not unique (*Drehtür* (revolving door) could be lemmatized to *Drehtür* or *Drehtüre* with no difference in meaning.) or (4) several analyses were equivalent. The disambiguation was often difficult. Even among the words retained in the test set, there were many that we were not fully sure about. An example is the compound *Natur eis bahn* ("natural ice rink") which we decided to analyze as *Natur-Eisbahn* (nature ice-rink) rather than *Natureis-Bahn* (nature-ice rink).

7 Results

The parser was trained using the Inside-Outside algorithm. By default, (a) the initialization of the rule probabilities was non-uniform as described in section 5, (b) training was based on tokens (i.e. the

[2] The complexity of an analysis is measured by the number of derivation and compounding steps.

frequency of the training items was taken into account), and (c) all training iterations were lexicalized. Training was quite fast. One training iteration on 2.3 million word forms took about 10 minutes on a Pentium IV running at 3 GHz.

Figure 4 shows the exact match accuracy of the Viterbi parses depending on the number of training iterations, which ranges from 0 (the initial, untrained model) to 15. For comparison, a baseline result is shown which was obtained by selecting the set of simplest analyses and choosing one of them at random[3]. The baseline accuracy was 45.3%. The parsing accuracy of the default model jumps from a starting value of 41.8% for the untrained model (which is below the baseline) to 58.5% after a single training iteration. The peak performance is reached after 8 iterations with 65.4%. The average accuracy of the models obtained after 6-25 iterations is 65.1%.

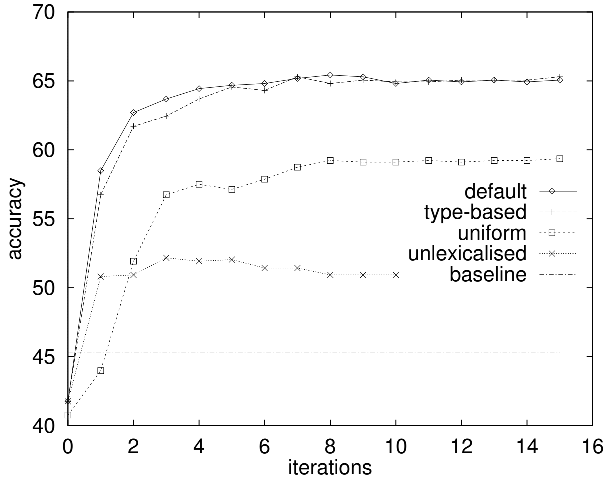

Figure 4: Exact match accuracy on data2

Results obtained with type-based training, where each word receives the same weight ignoring its frequency, were virtually identical to those of the default model. If the parser training was started with a uniform initial model, however, the accuracy dropped by about 6 percentage points. Figure 4 also shows that the performance of an unlexicalized PCFG is about 13% lower.

We also experimented with a combination of unlexicalized and lexicalized training. Lexicalized models have a huge number of parameters. Therefore, there is a large number of locally optimal parameter settings to which the unsupervised training can be attracted. Purely lexicalized training is likely to get stuck in a local optimum which is close to the starting point. Unlexicalized models, on the other hand, have fewer parameters, a smaller number of local optima and a smoother search space. Unlexicalized training is therefore more likely to reach a globally (near-)optimal point and provides a better starting point for the lexicalized training.

Figure 5 shows that initial unlexicalized training indeed improves the accuracy of the parser. With one iteration of unlexicalized training (see "unlex 1" in figure 5), the accuracy increased by about 3%. The maximum of 68.4% was reached after 4 iterations of lexicalized training. The results obtained with 2 iterations of unlexicalized training were very similar. With 3 iterations, the performance dropped almost to the level of the default model. It seems that some of the general preferences learned during unlexicalized training are so strong after three iterations that they cannot be overruled anymore by the lexeme-specific preferences learned in the lexicalized training.

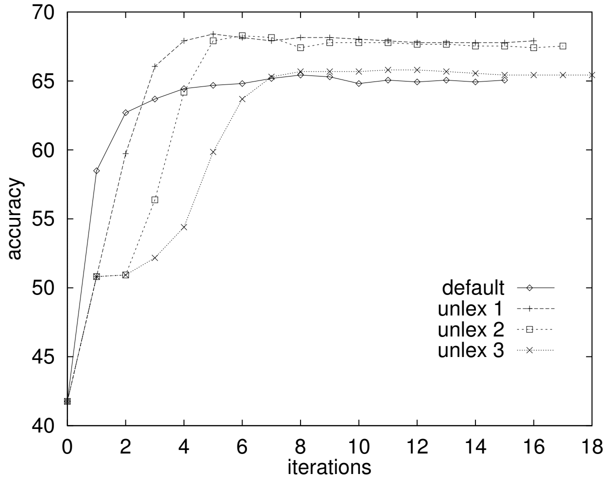

Figure 5: Results on data2 with 0 (default), 1, 2, or 3 iterations of unlexicalized training, followed by lexicalized training

In order to assess the parsing results qualitatively, 100 parsing errors of version "unlex 2" were randomly selected and inspected. It turned out that

[3] In fact, we counted a word with n simplest analyses as $1/n$ correct instead of actually selecting one analysis at random, in order to avoid a dependency of the baseline result on the random number generation.

the parser always preferred right-branching structures over left-branching structures in complex compounds with three or more elements, which resulted in 57 errors caused by left-branching structures. Grammars trained without the initial unlexicalized training showed no systematic preference for right-branching structures. In the test data, left-branching structures were two times more frequent than right-branching structures.

29 disambiguation errors resulted from selecting the wrong stem although the structure of the analysis was otherwise correct. In the word *Rechtskonstruktion* (legal construction), for instance, the first element *Rechts* was derived from the adjective *rechts* (right) rather than the noun *Recht* (law). Similarly, the adjective *quelloffen* (open-source) was derived from the verb *quellen* (to swell) rather than the noun *Quelle* (source).

Six errors involved a combination of compounding and suffix derivation (e.g. the word *Flugbegleiterin* (stewardess)). The parser preferred the analysis where the derivation is applied first (*Flug-Begleiterin* (flight attendant[female])), whereas in the gold standard analysis, the compound is formed first (*Flugbegleiter-in* (steward-ess).

In order to better understand the benefits of unlexicalized training, we also examined the differences between the best model obtained with one iteration of unlexicalized (unlex1), and the best model obtained without unlexicalized training (default).

30 cases involved left-branching vs. right-branching compounds. The unlex1 model showed a higher preference for right-branching structures than the default model, but produced also left-branching structures (unlike the model unlex2). In 15 of the 30 cases, unlex1 correctly decided for a right-branching structure; in 13 cases, unlex1 was wrong with proposing a right-branching structure. In two cases, unlex1 correctly predicted a left-branching structure and the default model predicted a right-branching structure.

32 differences were caused by lexical ambiguities. In 24 cases, only one stem was ambiguous. 15 times unlex1 was right (e.g. *Moskaureise* - Moskow trip[sg] vs. Moskow rice[pl]) and nine times the default model was right (e.g. *Jodtabletten* - iodine pill vs. iodine tablet). In 8 cases, two morphemes were involved in the ambiguity. In all these cases, unlex1 generated the correct analysis (e.g. *Sportraum* - “sport room” vs. “Spor[name] dream”).

Nine ambiguities involved the length of verb prefixes. Six times, unlex1 correctly decided for a longer prefix (e.g. *gegenüber-stehen* (to face) instead of *gegen-überstehen* (to “counter-survive”).

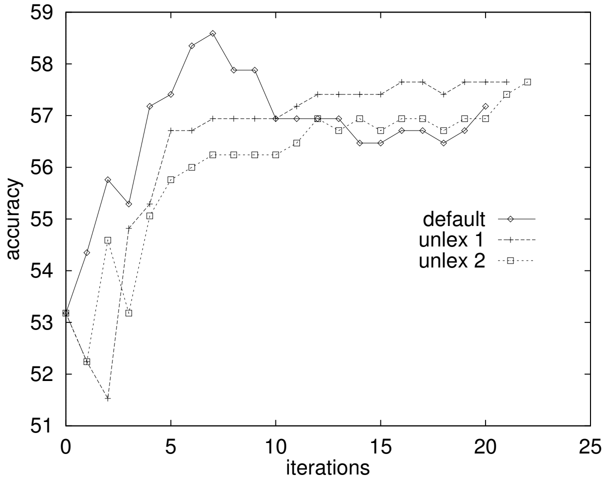

Figure 6: Accuracy on data1 after 0, 1, or 2 iterations of unlexicalized training followed by lexicalized training

In another experiment, we tested the parser on the first test data set (data1) where simplex words, part-of-speech ambiguities, frequent words and repeated occurrences were not removed. The baseline accuracy on this data was 43.75%. Figure 6 shows the results obtained with different numbers of unlexicalized training iterations analogous to figure 5. Strictly lexicalized training produced the best results, here. The maximal accuracy was 58.59% which was obtained after 7 iterations. In contrast to the experiments on data2, the accuracy decreased by more than 1.5% when the training was continued. As said in the introduction, we think that part-of-speech ambiguities are better resolved by a part-of-speech tagger and that frequent words can be disambiguated manually.

8 Related Work

New methods are often first developed for English and later adapted to other languages. This might explain why morphological disambiguation has been so rarely addressed in the past: English morphology is seldom ambiguous except for noun compounds.

We are not aware of any work on the disambiguation of morphological analyses which is directly comparable to ours. Mark Lauer (1995) only considered English noun compounds and applied a different disambiguation strategy based on word association scores.

Koehn and Knight (2003) proposed a splitting method for German compounds and showed that it improves statistical machine translation. Compounds are split into smaller pieces (which have to be words themselves) if the geometric mean of the word frequencies of the pieces is higher than the frequency of the compound. Information from a bilingual corpus is used to improve the splitting accuracy.

Andreas Eisele (unpublished work) implemented a statistical disambiguator for German based on weighted finite-state transducers as described in the introduction. However, his system fails to represent and disambiguate the ambiguities observed in compounds with three or more elements and similar constructions with structural ambiguities.

9 Summary

We presented a disambiguation method for German morphological analyses which is based on a head-lexicalized probabilistic context-free grammar. The words are split into morpheme lattices by a finite state morphology, and then parsed with the probabilistic context-free grammar. The grammar was trained on unlabeled data using the Inside-Outside algorithm and evaluated on 807 manually disambiguated analyses of infrequent words. Different training strategies have been compared. A combination of one iteration of unlexicalized training and four iterations of lexicalized training returned the best results with over 68% exact match accuracy, compared to a baseline of 45% which was obtained by randomly choosing one of the minimally complex analyses. Without lexicalization, the accuracy dropped by 15 percentage points, indicating that lexicalization is essential for morphological disambiguation.

10 Future Work

There are several starting points for improvement. Guidelines should be developed for the manual annotation of data in order to make it less dependent on the annotator's intuitions. More data should be annotated to create a treebank of morphological analyses. Given such a treebank, the parser could be trained on labeled data or on a combination of labeled and unlabeled data, which presumably would further increase the parsing accuracy.

References

Glenn Carroll and Mats Rooth. 1998. Valence induction with a head-lexicalized PCFG. In *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing*, Granada, Spain.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word Statistics. In *Proceedings of the 14th National Conference on Artificial Intelligence*, Menlo Parc.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary.

K. Lari and S. Young. 1990. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computation Speech and Language Processing*, 4:35–56.

Mark Lauer. 1995. Corpus statistics meet the noun compound: Some empirical results. In *Proceedings of the 33rd Annual Meeting of the ACL, Massachusetts Institute of Technology*, pages 47–54, Cambridge, Mass. electronically available at http://xxx.lanl.gov/abs/cmp-lg/9504033.

Anne Schiller. 1996. Deutsche Flexions- und Kompositionsmorphologie mit PC-KIMMO. In Roland Hausser, editor, *Proceedings, 1. Morpholympics, Erlangen, 7./8. März 1994*, Tübingen. Niemeyer.

Tanja Schmid, Anke Lüdeling, Bettina Säuberlich, Ulrich Heid, and Bernd Möbius. 2001. DeKo: Ein System zur Analyse komplexer Wörter. In *GLDV - Jahrestagung 2001*, pages 49–57.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. SMOR: A German computational morphology covering derivation, composition and inflection. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, volume 4, pages 1263–1266, Lisbon, Portugal.