Parsing Morphologically Complex Words

Kay-Michael Würzner*
University of Potsdam, Psychology Dept.
Karl-Liebknecht-Str. 24-25
14476 Potsdam, Germany
wuerzner@uni-potsdam.de

Thomas Hanneforth
University of Potsdam, Linguistics Dept.
Karl-Liebknecht-Str. 24-25
14476 Potsdam, Germany
thomas.hanneforth@uni-potsdam.de

Abstract

We present a method for probabilistic parsing of German words. Our approach uses a morphological analyzer based on weighted finite-state transducers to segment words into lexical units and a probabilistic context free grammar trained on a manually created set of word trees for the parsing step.

1 Introduction

Most existing systems for automatic, morphological analysis of German focus on flat structures, i.e. the segmentation into morphemes and the identification of their features and the involved operations. But as soon as more than one operation leads to the word in question, possible orderings of these operations can be captured in different hierarchical structures. Consider Ex. (1) from Faaß et al. (2010),

(1) un$_{Pref}$ übersetz$_{V}$ bar$_{Suff}$
un translate able
'untranslatable'

The adjective *unübersetzbar* is analyzed as a combination of the prefix *un*, the verbal stem *übersetz* and the suffix *bar*. This analysis could be assigned two structures (depicted in Fig. 1): either the prefixation (a) or the suffixation (b) occurs first.

Research on human morphological processing has long moved from linear segmentations to more advanced representations of the morphological structure of words (Libben, 1993; Libben, 1994). We aim to provide researchers in this field with hierarchical morphological analyses for all words in our lexical database `dlexDB` (Heister et al., 2011).

In the following, we present an approach for the automatic assignment of hierarchical structures to complex words using flat morphological analyses and a PCFG[1]. As a case study, we apply our method to the parsing of German adjectives. To do so, we created a corpus of manually annotated word trees for 5,000 structurally ambiguous adjectives. We describe types of ambiguity and their distribution in the training set and report results of the parsing process in dependence of various grammar transformations.

1.1 Word Formation and Structures

Word formation is the combination of morphemes to form new words. We distinguish between *inflection* (combination of a free morpheme with one or more affixes to fulfill agreement), *compounding* (combination of several free morphemes) and *derivation* (combination of a morpheme with an affix to change the category and/or the meaning of a word). *Conversion* might be considered a special case of derivation. Here, a change of a word's category occurs without any affixes being involved.

Word formation processes which are involved in the creation of a complex word can be linearly ordered. Multiple possible orderings lead to structural ambiguities. Ex. (2) gives examples for the possible different types of ambiguity[2]: *Compound – Suffix*, *Compound – Compound*, *Prefix – Suffix*, *Prefix – Compound*.

(2) a. Mensch$_{N}$ en Freund$_{N}$ lich$_{Suff}$
human link *friend ly*
'humanitarian'
b. dunkel$_{A}$ Asche$_{N}$ grau$_{A}$
dark ash gray
'dark ashen'
c. nicht$_{Pref}$ Objekt$_{N}$ iv$_{Suff}$
non object ive
'non-objective'
d. ab$_{Pref}$ Gas$_{N}$ frei$_{A}$
off gas free
'zero-emission'

The decision which ordering is the correct one is driven by morphological as well as semantic restrictions on the involved morphemes. The tree given in Fig. 1a for example could be ruled out by the fact that verbs may

* This author's work was funded by the DFG (grant no. KL 955/19-1).

[1] We assume here the usual definition of a context-free grammar (CFG) $G = (V,T,S,P)$ consisting of non-terminal (V) and terminal symbols (T), a start symbol $S \in V$ and a set of context-free productions P. In a *probabilistic CFG* (PCFG; Booth, 1969), each production is assigned with a probability.

[2] Since inflection in German is triggered by a word's context to ensure agreement and from a productive point of view always takes place last in word formation, we ignore it in our list and for the remainder of this work and restrict ourselves to base forms (*lemmas*) of the words in question.

Proceedings of the 11th International Conference on Finite State Methods and Natural Language Processing, pages 39–43,
St Andrews–Sctotland, July 15–17, 2013.

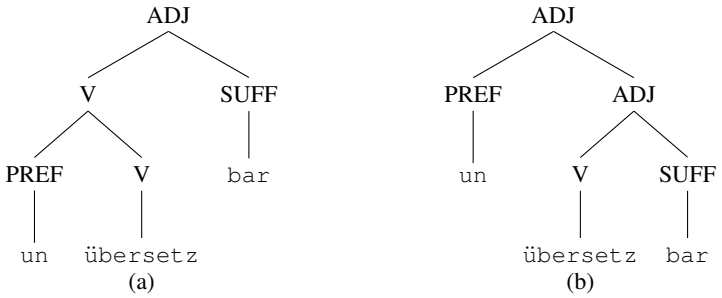


Figure 1: Possible tree structures for the morphological analysis in Example (1).

not be combined with the prefix *un* in German. As an example for semantic restrictions consider the analysis for *antirheumatisch* given in Ex. (3).

(3) anti$_{Pref}$ Rheuma$_{N}$ t isch$_{Suff}$
anti *rheumatism* link *ish*
'antirheumatic'

Since the concept of "antirheumatism" does not exist, we assume that the suffixation with *isch* takes place first.

1.2 Morphological Analysis

Before parsing, input words must be segmented into their basic units. In addition, the parser needs sufficient categorical annotation to get started. For that purpose, we used the TAGH morphology (Geyken and Hanneforth, 2006), a comprehensive computational morphology system for German based on weighted finite-state transducers.

Computer morphology systems normally suffer from oversegmentation: a sufficiently long enough word gets segmented in all possible ways, resulting in a lot of ambiguous readings most of which are nonsensical. To tackle this problem, the TAGH morphology (TAGH-M) makes use of three strategies:

1. TAGH-M measures morphological complexity by associating each derivation and compounding rule with a context-dependent penalty weight. These weights are taken from a *tropical semiring* weight structure (Kuich and Salomaa, 1986), that is, weights are added along a path in the weighted finite-state automaton representing a set of morphological analyses, and, among the competing analyses, the one with the least weight is selected.

2. TAGH-M is not strictly morpheme-based, but instead more oriented towards semantics. In German, there are a lot of overtly morphologically complex words which nevertheless denote simple concepts. Take for example the exocentric compound *Geizhals* ('scrapepenny'). But it can be also segmented into *Geiz* ('miserliness') and *Hals* ('neck'). TAGH-M's base lexicon now contains morphological simple entries like *Hals*, but morphologically complex ones like *Geizhals* as well. In association with the weighting mechanism, this means, that lexicalized but complex forms will be always given priority.

3. The word formation grammar underlying TAGH-M is very carefully crafted. Looking at adjective formation, the corresponding subgrammar contains approx. 3,000 rules. These rules are divided into groups, responsible for prefixation, suffixation, compounding and conversion. The suffixation part of the grammar is itself divided into further groups, one for each productive adjective suffix like *-isch*, *-ig* or *-lich*.[3] Every suffixation rule is associated with a number of base stems of different category (nouns, names, etc.) which happen to take this particular suffix. The association of affixes and stems is derived from a huge list of entries taken from the German Google books corpus (see also Sec. 2.2).

By incorporating these three strategies, TAGH-M avoids a lot of segmentation ambiguities which would otherwise enter into the subsequent parsing phase.

In addition, TAGH-M inserts marker symbols (for separable and non-separable prefixes, suffixes, linking morphemes and free morphemes), reduces allomorphic variants to their underlying citation form and annotates each segment with a morphological category taken from a set of approx. 20 categories.

Ex. (4) shows the preferred segmentation of the adjective *länderspezifisch* ('country-specific').

(4) `Land ⟨N⟩ \er ⟨l⟩ # spezif ⟨f⟩ ∼ isch ⟨a⟩`

The symbols enclosed in angle brackets denote the morphosyntactic category of the segment: ⟨N⟩ is a free noun, while ⟨a⟩ represents a bound adjective (suffix *-isch*); ⟨f⟩ denotes a neoclassical formative and ⟨l⟩ a linking morpheme. The segmentation symbol # marks a free morpheme boundary, while ∼ flags a following suffix. Annotated segments as well as segmentation markers enter the subsequent parsing phase.

[3] In total, the adjective suffixation grammar lists almost 70 of these suffixes.

1.3 Parsing

To get an initial grammar for the training experiments reported on in Sec. 3, we manually derived a context-free grammar based on the grammar underlying TAGH-M. For parsing, this grammar was automatically converted into an unweighted *finite tree automaton, FTA* (Comon et al., 2007). Transitions of FTAs are either of the form

$$w \to q \tag{1}$$

which introduce the leaves of a tree, or rules of the form

$$f(q_1,q_2,\ldots,q_k) \to q \tag{2}$$

which describe k-branching tree nodes with label f; the q_is are the states of the FTA. The language of an FTA is the set of trees generated by the FTA.

Finite-tree automata offer an advantage over context-free grammars: their transitions decouple the label (functor) f of the transition (which corresponds to a context-free rule's left-hand side) from the destination state q of the transition (which reflects the context in which the subtree might be inserted). This for example make techniques like parent annotation (see below) easily applicable since annotated categories are represented in the states of the FTA, not its translation labels.

For word structure parsing, we used an intersection based approach (Hanneforth, 2013).

1.3.1 Lexicalization

In lexicalized grammars, individual productions are specialized for certain lexical items. Non-terminal symbols are extended with lexical information as shown in Fig. 2.

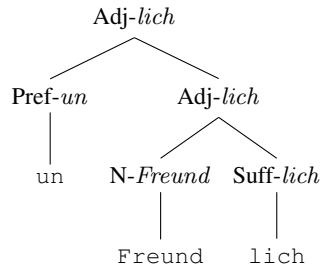


Figure 2: A lexicalized word tree.

This example is an instance of so called *head* lexicalization (Charniak, 1997). Lexical information of the rightmost constituent is percolated through the tree. Lexicalizing a grammar is a way to add some kind of contextual information to *context-free* grammars.

1.3.2 Parent Annotation

Parent annotation (Johnson, 1998) is another way of enriching a CFG with contextual information. The category of some non-terminal is added to the labels of its daughters as shown in Fig. 3.

Extending the grammar as described above increases the number of non-terminals and productions. This can lead to sparseness problems in the probabilistic case. These problems can be dealt with by applying some smoothing method (Collins, 1999).

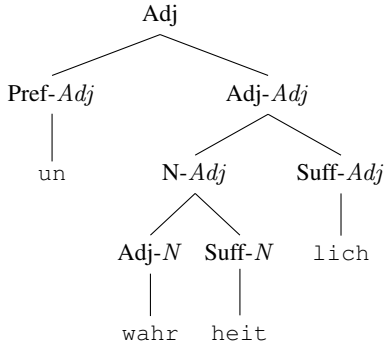


Figure 3: A word tree with parent annotation.

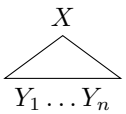
2 Method

In what follows, we describe our pilot study for the generation of parse trees for morphologically complex words. Our goal is to determine the most likely structure for each item in the test set, namely a large set of German adjectives.

2.1 Procedure

We decided to evaluate a statistical parsing approach using a PCFG. The aforementioned hand-crafted CFG which covers the different types of ambiguity was used to create candidate trees (cf. Fig. 1) for a set of training items (see Sec. 2.2). The design of the grammar also ensured that there is always an unary derivation from the pre-terminal to the terminal level. This allowed us to keep lexical rules separate from the rest of the grammar. The grammar contains no further unary productions.

After the manual annotation step described in Sec. 2.2, we divided the data into 10 equal parts and induced context-free productions from the trees in each part. For each subtree



a production $X \to Y_1 \ldots Y_n$ was added to P. We also stored the production's frequency in each of the 10 sub-parts.

Estimation of the probabilities for the productions in P and evaluation of the resulting PCFG was done by iterating over the sub-parts G_i which served as a test set while the rest was used for training.

Probabilities were computed via simple maximum likelihood estimation with add-one smoothing. Here, $c(X \to Y_1 \ldots Y_n)$ denotes the frequency of the rule $X \to Y_1 \ldots Y_n$ in the training materials.

Ambiguity	Number
Compound – Suffix	5,239
Prefix – Suffix	1,136
Compound – Compound	447
Prefix – Compound	0

Table 1: Numbers of different types of structural ambiguities within a set of 20,000 adjectives.

$$\Pr(Y_1 \dots Y_n | X) = \frac{c(X \rightarrow Y_1 \dots Y_n) + 1}{c(X \rightarrow (V \cup T)^+) + |P|} \quad (3)$$

In order to capture restrictions as those mentioned above, we applied various transformations on the trees prior to the grammar induction resulting in different grammar versions: (1) specialization of the pre-terminal level for bound morphemes, (2) specialization of the pre-terminal level for frequent free morphemes, (3) lexicalization of adjective suffixes and (4) parent annotation. The specialization of the pre-terminal level may be considered as lexicalization of only the lexical rules.

2.2 Materials

We chose the Google books N-grams (Google Incorporated, 2009) as our source for training and test materials. The list of unigrams contains all words with a frequency greater or equal to ten within the German portion of the Google books corpus (all in all 3,685,340 types). From this list, we extracted a large number of adjectives using a list of known adjective suffixes (see Sec. 1.2) and manually filtered this list for optical character recognition errors and false positives (e.g. verbal forms). This set was extended using known adjectives from various hand-maintained lexical resources resulting in a list of 338,423 adjectives.

Initially, we randomly selected 10,000 words with a length of $8 \leq n \leq 20$ (which were unique for their lemma; i.e., only one instance per lemma was selected) from this list. These words were morphologically analyzed and parsed along the lines of sections 1.2 and 1.3. The resulting analyses were manually checked for errors in the morphological analysis and the word trees. Detected errors led to readjustments of both the morphological analyzer and the grammar. Finally, only roughly a quarter of the items were assigned more than one tree (the main reason for this is that TAGH-M already removes a lot of possible ambiguities). That is why we added another 10,000 words (this time with a length of $10 \leq n \leq 25$) to get more ambiguous forms. Tab. 1 shows the numbers of the different possible types of ambiguity in the test set.

For training and evaluating the PCFG, we manually selected the preferred tree for 5,000 structurally ambiguous adjectives.

2.3 Evaluation

We used `evalb` (Sekine and Collins, 1997) to evaluate the different probabilistic grammars extracted from the training materials as described in Sec. 2.1. We report their performance in terms of (1) *tagging accuracy*, i.e., the proportion of correct pre-terminal to terminal assignments, (2) *bracketing accuracy*, i.e., the proportion of correct rule applications and (3) *complete matches*, i.e. the proportion of identities between manually selected and automatically generated trees.

3 Results and Discussion

Table 2 summarizes the results for the different grammar versions. The corresponding tree transformations are applied in a cumulative way. Due to the inclusion of the morpheme annotation done by TAGH-M into the grammar, tagging accuracy is always 100% and thus omitted in Table 2.

The biggest improvement is gained through the specialization of frequent free morphemes. This transformation helps us to model binding preferences for certain morphemes. Consider for example the noun *Freund* ('friend') which is very often combined with the suffix *lich* in order to form *freundlich* ('friendly'). There are many compounds with *freundlich* as *anwenderfreundlich* where, due to semantic restrictions, the noun compound as first component is not an option.

Head-lexicalization did not improve parsing results with one exception: The test materials contain many coordinative structures like *psychischphysisch* ('psycho-physical'), thus the production $Adj \rightarrow Adj\ Adj$ has a fairly high probability. But there is one notable exception to this rule: In words like Ex. (5),

(5) Nation$_N$ al$_{Suff}$ Sozialist$_N$ isch$_{Suff}$
nation al socialist ish
'Nazi'

a coordinative analysis is also available, but adjectives formed with *al* almost always combine with a noun if possible. Lexicalizing *al* successfully models this behavior. It will be subject to further work to systematically test for other equally successful lexicalization patterns.

Parent annotation does not add very much to the performance of the grammar which is due to the relatively simple structures that we encounter during parsing of words compared to the parsing of sentences.

If we look at the remaining errors, it is striking that most of these originate from exceptions from the typical formation patterns. The prefix *über* ('over') is usually combined with adjectives but in some rare cases it operates as a noun prefix (*Übermensch*, 'superman'). Derivations from these nouns are assigned with the wrong analysis by our grammar.

The approach we presented here is a promising first step in the direction of parsing morphologically complex words. Next, we will extend our approach to Ger-

Grammar	Number of prod.	Bracketing acc.	Complete match
baseline	63	91.64%	82.05%
specialized bound morphemes	196	92.20%	83.04%
specialized freq. free morphemes	238	94.92%	89.11%
lexicalized suffix `al`	274	96.26%	92.02%
parent annotation	481	95.91%	93.34%

Table 2: Number of non-lexical productions as well as proportions of correct bracketing and complete matches for different PCFGs.

man nouns where the great number of compounds will be the major challenge.

Acknowledgements

We would like to thank Edmund Pohl for creating the web-based tree training tool (`http://www.dlexdb.de/wordstruct`) and the anonymous reviewers for their helpful remarks.

References

Taylor L. Booth. 1969. Probabilistic Representation of Formal Languages. In *IEEE Conference Record of 10th Annual Symposium on Switching and Automata Theory*, pages 74–81.

Eugene Charniak. 1997. Statistical Techniques for Natural Language Parsing. *AI Magazine*, 18(4):33–43.

Michael John Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Hubert Comon, Max Dauchet, Remi Gilleron, Christof Löding, Florent Jacquemard, Denis Lugiez, Sophie Tison, and Marc Tommasi. 2007. Tree Automata Techniques and Applications. Available on: `http://www.grappa.univ-lille3.fr/tata`. release October, 12th 2007.

Gertrud Faaß, Ulrich Heid, and Helmut Schmid. 2010. Design and Application of a Gold Standard for Morphological Analysis: SMOR as an Example of Morphological Evaluation. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the seventh LREC conference*, pages 803–810.

Alexander Geyken and Thomas Hanneforth. 2006. TAGH: A Complete Morphology for German based on Weighted Finite State Automata. In *Finite State Methods and Natural Language Processing*, volume 4002 of *Lecture Notes in Computer Science*, pages 55–66, Berlin, Heidelberg. Springer.

Google Incorporated. 2009. Google Books Ngrams. http://books.google.com/ngrams.

Thomas Hanneforth. 2013. An Efficient Parsing Algorithm for Weighted Finite-tree Automata. *In preparation*.

Julian Heister, Kay-Michael Würzner, Johannes Bubenzer, Edmund Pohl, Thomas Hanneforth, Alexander Geyken, and Reinhold Kliegl. 2011. dlexDB – eine lexikalische Datenbank für die psychologische und linguistische Forschung. *Psychologische Rundschau*, 62(1):10–20.

Mark Johnson. 1998. PCFG Models of Lingusitic Tree Representations. *Computational Linguistics*, 24(4):612–632.

Werner Kuich and Arto Salomaa. 1986. *Semirings, Automata, Languages*, volume 5 of *EACTS Monographs on Theoretical Computer Science*. Springer.

Gary Libben. 1993. Are Morphological Structures Computed During Word Recognition? *Journal of Psycholinguistic Research*, 22(5):533–544.

Gary Libben. 1994. Computing Hierarchical Morphological Structure: A Case Study. *Journal of Neurolinguistics*, 8(1):49–55.

Satoshi Sekine and Michael John Collins. 1997. evalb – Bracket Scoring Program. http://nlp.cs.nyu.edu/evalb/.