SMOR: A German Computational Morphology Covering Derivation, Composition, and Inflection

Helmut Schmid, Arne Fitschen, Ulrich Heid

Institute for Computational Linguistics
University of Stuttgart
Azenbergstr. 12, 70174 Stuttgart, Germany
{schmid,fitschen,heid}@ims.uni-stuttgart.de

Abstract

We present a morphological analyser for German inflection and word formation implemented in finite state technology. Unlike purely lexicon-based approaches, it can account for productive word formation like derivation and composition. The implementation is based on the Stuttgart Finite State Transducer Tools (SFST-Tools), a non-commercial FST platform. It is fast and achieves a high coverage.

1. Introduction

German word formation morphology is characterized by a considerable number of productive compounding and derivation processes (both suffixation and prefixing). A morphological analyser for real text (e.g. newspapers) must be able to cope with complex words not contained as such in its lexicon, as they are built ad hoc according to productive word formation rules. Examples include particle and prefix verbs (*hinein-quietschen, ent-eisenen*), suffixation, e.g. with *-bar* (*enteisenbar*) and certain types of compounding, such as noun+noun (*Enteisenungsapparat*).

Existing (FST-based) morphology systems for German are either mainly based on large lexicons like the public version of WordManager (Domenig and Hsiung, 1996) and GerTWOL (Haapalainen and Majorin, 1995), or they cover only parts of German word formation like DMOR (Schiller, 1996) which lacks a derivation component, and DeKo (Schmid et al., 2001; Heid et al., 2002) which lacks inflection. With the exception of an experimental system described in (Lorenz, 1997), we are thus not aware of any other computational morphology for German covering productive word formation and inflection in one system.

2. SMOR

SMOR is designed for the morphological analysis of German word forms. For the input in (1), it produces analyses consisting of a sequence of morphemes with feature decorations, as shown in (2):

(1) *unübersetzbarstes* 'most untranslatable'

(2) un<PREF>übersetzen<V>bar<SUFF><+ADJ><Sup><Neut><Nom><Sg>

un<PREF>übersetzen<V>bar<SUFF><+ADJ><Sup><Neut><Acc><Sg>

The linguistic modelling principles behind SMOR are:

- SMOR implements a concatenative approach to morphology, postulating that affixes have their own lexical entries which encode selection constraints. Affixes select their base in terms of word class, stem type, origin and complexity (cmp. Lüdeling and Fitschen, 2002).
- Affixation is implemented as concatenation with feature checking; features encode the properties of bases and the selection constraints of affixes; this applies to suffixation and prefixing.
- Inflection is handled via continuation classes.
- Morphophonological rules are implemented with two-level rules which map analysis strings to surface strings (and vice versa).
- The lexicon plays a central role in SMOR. It encodes the properties of bases with respect to the
  - Entry type: <Stem> <Suffix> <Prefix>
  - Word class: <V> <ADJ> <NN> ...
  - Stem type: <base> <deriv> <compound>
  - Origin: <native> <foreign> <classical> ...
  - Complexity: <simplex> <prefderiv> <suffderiv>
  - Inflectional class: <Adj+> <NFem-Deriv> ...

We distinguish base, derivation, and compounding stems. The first element of compounds is always a compounding stem. The base of a derivation is usually a derivation stem, but some suffixes which originate from compounds etymologically, combine with compounding stems, instead. Only base stems are inflected.

The lexicon encodes the selectional constraints of affixes. In (3), we represent a lexicon entry for the adjectival prefix *un-*, and in (4) for the suffix *-bar* which derives an adjective from a native verb. (5) shows the lexicon entry of the suffix *-schaft* '-ship' which derives a noun from a noun. It combines with compounding stems to form a derivation stem. This lexicon entry would be needed to analyse the word *freundschaftlich* 'friendly'.

(3) <Prefix>un<ADJ><native>

(4) <Suffix><simplex><native><deriv><V>bar<ADJ><base><nativ><Adj+>

(5) <Suffix><simplex><native><comp><NN>schaft
<NN><deriv><native>

SMOR contains several rules deriving derivation and compounding stems from base stems. The nominative singular of a noun e.g., is, by default, a compounding as well as a derivation stem. The genitive singular of feminine nouns and the nominative plural of all nouns are also default compounding stems. Similar rules apply to other word classes. Derivation and compounding stems which are not covered by these rules, are explicitly listed in the lexicon.

3. Implementation

The implementation of SMOR was written in the SFST transducer specification language which is based on extended regular expressions with variables and operators for concatenation, conjunction, disjunction, repetition, composition, negation, context-dependent replacement, and more. The formalism is easy to learn, powerful, flexible, and open to different styles of implementing finite state morphology. A compiler translates the transducer specifications to minimised finite state transducers to be used by the analyser. Its efficiency is comparable to that of commercial systems. The SFST tools are available under the GNU public license.

The basic operations of the SMOR implementation are concatenation of morphemes, filtering of morpheme sequences (by checking feature agreement), and mapping of the resulting analysis strings to surface realisations (by applying phonological rules).

The SMOR transducer is created incrementally. Stems, prefixes and affixes are listed in the lexicon. Derived forms are generated by adding suffixes to stems. The resulting transducer is composed with a filter transducer to check and delete the suffix agreement features. We call the result S_0. Prefix derivations are generated by adding prefixes to S_0 and checking feature agreement, resulting in a transducer P_1. Further suffixes are added to P_1 with feature checking to obtain S_1. The disjunction of S_0 and S_1 forms the set of simplex and derived stems.

These stems are concatenated to form compounds. A filter ensures that all but the last stem are compounding stems. Inflectional endings are generated using continuation classes and added to the compounds. A filter eliminates incorrect endings. Finally, phonological rules are applied to map analysis strings to surface forms.

We will now present the implementation in more detail. The code given in the examples conforms to the SFST syntax, but we simplified it in order to increase readability.

The command `$LEX$ = "lexicon"` reads the lexicon from the file *lexicon*, generates a transducer which recognises each lexicon entry and assigns it to the variable LEX. The lexicon is split into sublexica. The command `$Prefix$ = $LEX$ || <Prefix> .*` e.g. extracts prefixes.

We add suffixes to the stems and compose the result with a suffix filter transducer as shown in (6).

```
(6) $S0$ = $Stems$ ($SimplexSuffix$
  $SuffDerivSuffix$*)? || $SuffixFilter$
```

The filter is implemented as a cascade of mappings which examine the origin, stem type, and category features. (7) shows the implementation of the filter for the stem type feature. The filter eliminates the feature markers. The expression `<deriv>:<>` maps the multi-character symbol `<deriv>` to the empty symbol represented by `<>`.

```
(7) $F$ = <deriv>:<><Suffix><deriv>:<> |
          <comp>:<> <Suffix><comp>:<>
    $STEMTYPE$ = (.* $F$)* .*
```

The complexity feature is treated differently: the set of suffixes is split into subsets according to the complexity selectional constraint (simplex, prefix derivation and suffix derivation) and then concatenated in the right order as shown in (6). Step (8) adds prefixes and checks their selectional constraints.

```
(8) $P1$ = $Prefix$ $S0$ ||
      $PrefixFilter$
```

Only one prefix is allowed because the compilation of multiple prefixes turned out to be intractable. The reason is that the compared features are separated by arbitrarily many suffixes. During analysis, the transducer must remember the prefix features in its state while analysing the intervening material until it reaches the matching suffix. With multiple prefixes, the number of transducer states explodes. A theoretical justification for the proposed restriction is given by Erben (2000) who argues that the second prefix of apparent counterexamples like *über* in *unübersetzbar*, is actually part of a listed word stem (*übersetz*).

```
(9) $S1$ = $P1$ ($PrefDerivSuffix$
 $SuffDerivSuffix$*)? || $SuffixFilter$
```

Step (9) adds additional suffixes to the prefix derivations in a similar way as before.

```
(10) $Compounds$ = ($S0$ | $S1$)+ ||
      $CompoundFilter$
```

Compounds are formed in (10) by concatenating (derived) word stems and checking that all but the last stem are compounding stems. This check is performed by the transducer stored in `$CompoundFilter$`. The transducer also eliminates markers which are not needed anymore.

```
(11) $Base$ = $Compounds$ $Inflection$
   || $InflectionFilter$
```

The last concatenation step (11) adds inflectional endings and an inflection filter eliminates incorrect endings by checking the inflection feature.

```
(12) $NFem-in$ = $NFem/Sg$ |
     {<>}:{nen} $NFem/Pl$
```

The inflectional endings are generated by a system of continuation classes which were automatically translated from the code of the DMOR morphology (Schiller, 1996). The command in (12) expands the inflection class `$NFem-in$` of nouns like *Freundin* (female friend) to either `$NFem/Sg$`, which will generate the singular inflection, or to `$NFem/Pl$` with insertion of the string *nen*, which produces the plural form *Freundinnen*.

```
(13) $Base$ = $Base$ || $PhonRules$
```

The phonological rules are applied in (13). Conceptually, the different rules are applied in sequence rather than in parallel in order to simplify the development by reducing possible interferences between rules. In the implementation, this means that the transducers for the individual rules are combined with composition rather than conjunction. The resulting filter transducer is stored in `$PhonRules$`.

```
(14) $Result$ = $Base$ || $UpLowFilter$
```

At the end, the transducer `$UpLowFilter$` ensures that nouns and proper names are capitalised.

The whole source code (without the lexicon) comprises about 1500 lines (without comments).

4. Evaluation

For the purpose of evaluation, we took 900 word forms from the German part of the European Language Newspaper Text corpus (in the following *ELNC*). The corpus is available from the Linguistic Data Consortium LDC (item *LDC95T11*). This is unseen material for SMOR because it has been developed on material from another large German corpus. Of the 900 word forms, a third was taken from the high frequency word forms, another third from the medium frequency word forms, and a third from the low frequency word forms.

number of tokens	103.798.402	
number of types	1.392.834	
number of hapaxes	654.753	(47 % of all types)

Table 1: ELNC statistics

number of types analysed	885.590	(63.6 %)
number of types not analysed	507.244	(36.4 %)

Table 2: SMOR statistics

For testing, we only considered word forms consisting of letters a-z, A-Z, German umlauts (ä,ö, and ü), and German sharp s (ß), because the purpose of the morphological analyser is not to work on non-words that are quite frequent in text corpora, but to perform well on morphologically complex word forms.

number of tokens	80.214.318	
number of types	795.361	
number of hapaxes	333.433	(42% of all types)

Table 3: ELNC statistics for word forms of the type /[a-zäöüßA-ZÄÖÜ]+/

For obtaining the 300 most frequent word forms, we sorted the list of 795.361 types by frequency in descending order. The first 300 word forms cover more than 50% of all the 80 million tokens. This is the group where every morphological analyser should perform well. The medium frequency word forms were taken from the range between 300 and 18.601 of the sorted list. This is due to the fact that the most frequent 18.601 word forms cover 90% of all the tokens. Thus, performing well on this group of word forms means to likely perform well on a large part of an unseen corpus. The last group of 300 word forms was taken from the range between 18.601, and 795.361. This is the range that usually contains many proper names, mis-spellings, and, especially in German texts, word formations. Here, we hoped to perform better than DMOR, the old morphology system at the IMS.

number of types analysed	531.571	(66.8%)
number of types not analysed	263.790	(33.2%)
not analysed by DMOR	282.019	(35.5%)

Table 4: SMOR statistics for word forms of the type /[a-zäöüßA-ZÄÖÜ]+/

We then marked manually, for each of the 900 word forms, which morphological analysis it should get. Amongst German morphologists, unfortunately, there is no general agreement yet about what constitutes the *correct* analyses. Besides the ambiguous word forms, there is especially the problem of the depth of the analysis. Every native German speaker knows the relation between the verb *versichern* 'to insure', and the nominalisation *Versicherung* 'insurance', but how do we segment the compound *Versicherungspolice* 'insurance policy'? There are two immediate constituents *Versicherungs*, and *police*, but a morphological analyser aware of word formation may be tempted to segment deeper, into *versichern* + *ung(s)* + *police*. We do not solve this problem here, but we marked the word forms in a 'liberal' manner, i.e., we did not care about the depth as long as the segments 'make sense'. On the other hand, we see no relation between the adjective *pünktlich* 'on time' and the noun *Punkt* 'dot, point', so the single analysis that SMOR found (`Punkt<NN>lich<SUFF><+ADJ>` was taken as wrong. As long as there is no 'gold' standard for morphological and word formation analysis of German word forms, it is very difficult to evaluate German morphological tools.

Nonetheless, here are the results for the 900 word forms. In total, out of the 900, 767 were analysed (85%). Of these, 31 had the wrong analysis.

	all	true positives	false positives
analysed	767 (85%)	736 (95.96%)	31 (4.04%)
	all	true negatives	false negatives
not anal.	133 (15%)	109 (82.0%)	24 (18.0%)

Table 5: Results for all 900 word forms

Here are the numbers for the different parts of the evaluation:

	all	true positives	false positives
analysed	290 (96.7%)	288 (99.3%)	2 (0.7%)
	all	true negatives	false negatives
not anal.	10 (3.3%)	9 (90%)	1 (10%)

Table 6: Results for high frequency word forms

	all	true positives	false positives
analysed	264 (88%)	261 (98.86%)	3 (1.14%)
	all	true negatives	false negatives
not anal.	36 (12%)	27 (75%)	9 (25%)

Table 7: Results for medium frequency word forms

	all	true positives	false positives
analysed	213 (71%)	187 (87.79%)	26 (2.21%)
	all	true negatives	false negatives
not anal.	87 (29%)	73 (83.91%)	14 (16.1%)

Table 8: Results for low frequency word forms

As expected, the number of non-analysed forms rises with decreasing frequency of the item. Nine out of the ten word forms not analysed in the high frequency list are abbreviations and seem to be specific to the text: they mark the author of a news message. The one false negative was a verb form which was not analysed because of a small mistake in SMOR: this is still work in progress... In the medium frequency part, an adjective, *hilflos* ’helpless’, was not analysed. This was due to the fact that the derivation stem *hilf* for either *Hilfe* ’help’ or *helfen* ’to help’ was missing from the lexicon. Besides this, again there were some names we deemed common enough to be marked as false negatives: *Bellinzona*, *Hebron*, *Riad*, and some word formations including names: *Oberägypten* ’upper Egypt’, *Ostbosnien* ’eastern Bosnia’. In the low frequency list, finally, there were some more cases involving word formation: False positives included *Einzelbehörden* ’single authorities’, where *Einzel* ’singles (tennis)’ was misinterpreted for a noun (instead of an adjective), and two analyses given for mis-spellings. *Erkennnissen* is meant to be spelled *Erkenntnissen* ’cognitions’, but since there is a word *Nisse* ’nit (biol.)’ in German, SMOR segmented it wrongly into `erkennen<V>Nisse<+NN>` ’recognise + nit’. *Bauerhöfe* (lit.: ’farmer’ + ’yards’) is probably meant to be *Bauernhöfe* ’farms’ (*Bauern* is the correct compounding stem of the noun *Bauer* ’farmer’). It is arguable if it is a false positive at all.

All in all, in the data there were surprisingly little ’interesting’ word formations. Much of the data consisted of proper names, abbreviations, and noun-noun compounds which pose no problems to a good morphological analyser. From a few instances, however, we could see that it is important to enhance the lexicon by adding derivational and compounding stems. While adding proper names may add quantity (and quality) to the analyses more quickly, we see adding word-formation-specific information as an important additional facet of improving the lexicon, and the automatic morphological analysis.

Morphological analysis using the SFST tool fst-infl2 is performed at a speed of more than 4000 words per second using a Sun Blade 1000 Model 2750 (750 MHz CPU).

5. Summary

We described the development of a German morphological analyser covering prefix and suffix derivation, compounding, and inflection. Our implementation follows a concatenation approach. Stems, prefixes, suffixes, and inflectional endings are marked with agreement features and they are concatenated. Filters eliminate the sequences which violate agreement constraints. The resulting strings are mapped to their surface forms by means of two-level rules.

The analyser was implemented with the Stuttgart Finite State Transducer (SFST) tools. It is fast and achieves high coverage.

References

Domenig, Marc and Alain Hsiung, 1996. Concepts and tools for lexical knowledge acquisition. *AI communications*, 9(2):79–82.

Erben, Johannes, 2000. *Einführung in die deutsche Wortbildungslehre*. Berlin, Germany: Erich Schmidt Verlag, 4th edition.

Haapalainen, Mariikka and Ari Majorin, 1995. Gertwol: Ein System zur automatischen Wortformerkennung deutscher Wörter. Technical report, Lingsoft Inc.

Heid, Ulrich, Bettina Säuberlich, and Arne Fitschen, 2002. Using descriptive generalisations in the acquisition of lexical data for word formation. In *Proceedings of the 3rd Conference on Language Resources and Evaluation*, volume IV. Las Palmas de Gran Canaria, Spain.

Lüdeling, Anke and Arne Fitschen, 2002. An integrated lexicon for the automatic analysis of complex words. In *Proceedings of the Tenth EURALEX International Congress*, volume I. Copenhagen, Denmark.

Lorenz, Oliver, 1997. *Automatische Wortformerkennung für das Deutsche im Rahmen von MALAGA*. Master’s thesis, Friedrich-Alexander-Universität, Erlangen, Germany.

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

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