

How to Disassemble Alphabetical Processions - Morphological Treatment of Unknown Words

Stephan Bopp, Sandro Pedrazzini, Elisabeth Maier

Canoo Engineering AG
Kirschgartenstr. 7
CH-4051 Basel

{stephan.bopp,sandro.pedrazzini,elisabeth.maier}@canoo.com

Abstract

This paper describes an approach how to integrate the decomposition of non-lexicalized word compounds and derivations into the morphological analyzers of a NLP product line. The component employs word formation rules and filtering techniques to decompose words, which are not contained in the underlying dictionary database, thereby increasing the average word recognition rate of the morphological analyzers from 90.6% to 95.4%.

1. Introduction

The amount of information offered on the World Wide Web is growing at an enormous rate. The time an average person spends on searching for relevant information in large knowledge bases is an important cost factor. Tools to facilitate and automate such tasks provide a significant reduction of costs.

In a relevant text, a search term may occur in numerous variants, e.g. as inflected form, as part of a compound, in regional writing, or according to company-specific spelling standards. Therefore, sophisticated morphological techniques are necessary for their normalization:

- recognition and generation of spelling variants,
- mapping of inflected forms onto their respective base forms
- decomposition of words into simpler components

It is crucial that such products are based on dictionaries, which cover a broad range of words – the quality of the products is directly proportional to the size and quality of the dictionary data.

It is typical of languages like German that new words can be created on the fly by means of word composition and word derivation. Therefore, in addition to large dictionaries, we also need rules that capture these highly productive word formation processes; cf. (Lüdeling 2002). This paper describes the extension of morphological components to handle the recognition of words not contained in a dictionary, i.e. non-lexicalized words, by employing mechanisms of intelligent word decomposition.

2. Word Manager and WMTrans

Word Manager (WM) is a system for the specification, use and maintenance of morphological dictionary databases (ten Hacken & Domenig, 1996), (Domenig 1992). A WM database describes the morphological system of a language, including inflectional, word formation and spelling rules. To deploy WM dictionary databases within complex applications, lexicon entries are compiled into finite state transducers (WMTrans). A whole range of WMTrans products has been developed and can easily be extended to meet customer-specific requirements.

Currently, Word Manager contains over 250.000 entries for German. 89% of these entries are composed or derived out of 11% base entries. This illustrates the highly generative character of the German language and the potential of considering word formation rules.

If word formation rules are used to analyze non-lexicalized words for a possible decomposition into known base words and adjuncts, the rate of recognized words in a text grows considerably.

This generative use of word formation rules, while allowing the analysis of non-lexicalized correct words, causes the new problem of overgeneration: the result set also contains non admissible analyses, which must be removed by the application of specialized filters.

In the following, we describe a mechanism for the analysis of non-lexicalized words in the context of the WMTrans product line. In the remainder of this paper we refer to the products that integrate this functionality as Unknown Word Transducers.

3. Intelligent Decomposition of Unknown Words using WMTrans

Currently, Word Manager contains over 250.000 entries for German of which 89% are composed or derived out of 11% base entries. If word formation rules are used to analyze non-lexicalized words the recognition rate grows considerably.

The *Unknown Word Transducers* for the analysis of non-lexicalized words are based on a WM dictionary. The generative use of word formation rules causes overgeneration: the result set contains non-admissible analyses. To eliminate these wrong analyses, the following steps are applied:

3.1 Prefilters: Admissible Word Formation Rules

Based on the WM dictionary and a large text corpus analysis, we selected 150 out of 500 word formation rules:

- Unproductive rules and affixes forming only few words were excluded, assuming that no – or hardly any – new words are likely to be formed with them;
- Some rules and affixes were eliminated, because their rate of undesired analyses is much higher than their rate of correct results: e.g. the adjective

forming suffix *-en*, the noun prefix *in-*, and all word formation rules for conversion (zero-derivation);

- Recursive application of word formation rules is not allowed, i.e. only one word formation rule may be applied for the segmentation of a string. Exceptions are some of the most productive word formation rules such as the rules for German noun-noun-compounding or the suffix combination *verb+bar+keit*.

3.2 Prefilters: Selection of Admissible Word Forms

The set of admissible word forms contains all strings that may be used by the word formation rules as base words for the segmentation. The number of undesired analyses can be reduced by cancelling irrelevant base words: only nouns, adjectives, verbs and adverbs are allowed. Within these categories individual lexemes and lexeme groups can be excluded:

- Entire lexemes with all their word forms are cancelled: e.g. the uncommon noun *Lichen* (engl. disease *lichen*) is excluded because it is homonymous with an inflected form of the much more common suffix *-lich*; other examples are the nouns *Amen*, *Anus*, *Acht*.
- specific word forms of lexemes are cancelled: e.g. certain stems of irregular verbs which are homonymous with nouns: *schloss*, *spross*, *zwang*,
- all two- and three-letter-words are eliminated except for the most common ones like e.g. *Ei* (egg), *Öl* (oil), *See* (lake), *neu* (new).

3.3 Postfilters: Deletion of Non-Valid Results

The reduced sets of word formation rules and possible base words still produce some wrong and undesired results. Postfilters handle many of them. Based on the result string, the applied word formation rule and other information, postfilters define conditions that, when met, trigger the deletion of a result from the list of analyses.

Example 1: German nouns ending in *-heit*, *-keit*, *-ität* and *-ion* are always followed by the linking element *s* when in first position of a compound. A postfilter deletes all analyses that do not meet this condition. Therefore only the first of the following two results of the analysis of *Missionssorte* is passed on to the final list of results: 1) "mission+s+orte" (*mission towns*), 2) "mission+sorte" (*mission type* - if the word existed in German, the correct compound would have to be *Missionssorte*).

Example 2: The adverb *gratis* (*free of charge*) may only be combined with nouns, but not with adjectives and verbs. A postfilter deletes all Adv+Adj and Adv+V compounds including the word *gratis* in first position: "gratis+lieferung" (*free delivery*) is accepted, "gratis+gelieferte" (*delivered free of charge*) is rejected. The latter must be written in two words. In this manner, postfilters can also be used for spell checking of unknown words.

3.4 Analyzer Output

According to the needs of a client application, the unknown word transducers offer the following results for analyzed words:

- citation form (word form used to refer to a lexical unit, i.e. infinitive of verbs, nominative singular of nouns, etc.),
- word category: noun, adjective, verb, etc.,
- inflection class,
- segmentation: segmented string where the segments are separated by a + sign
- applied word formation rule.

In future releases, we will also provide the following information:

- word form features (number, case, person etc.),
- indication of the base lexemes used for the segmentation.

Query	universitätsprofessors
Cit-Form	Universitätsprofessor
Features	(Cat N)(Gender M) (Num Sing)(Case Genitive)
Inflection Rule	RIRule N-Regular s/en
Segmentation	universität+s+professors
Word Formation Rule	WFRule Compounding N-Compound N+s+N
Base Elements	Universität (Cat N)(Gender F) s (WFCat Linking-Element) Professor (Cat N)(Gender M)

Example 3: Possible analyzer output for query *Universitätsprofessors*.

4. Implementation

The Unknown Word products have been implemented through a combination of finite state machines (FSMs). The necessary information is contained in the Word Manager database. The information consists of:

- complete inflection information for lexicalized words, i.e. for words contained in the Word Manager database,
- information about stems and surface variants,
- word formation information such as linking elements, suffixes and prefixes,
- word formation rules, including different aspects of derivation and word formation for German, English and Italian. Such rules have been tested using a large amount of data.

4.1 Different Finite State Elements

The above mentioned information types are transformed into optimized regular expressions and distributed to the following finite state components:

- Inflection transducer
This component is the basis tool for the analysis of lexicalized words. The component ensures high processing speed because it avoids further segmentation and analysis. The inflection transducer is also used to map single elements of the unknown word analysis to lexicalized components. Moreover, it

is the main tool to compute the citation form from the analyzed unknown word form. For further details on the use of the inflection transducer, see section 4.3.

- **Surface finite state automaton**
This automaton is used to code all possible word formatives, as generated from Word Manager. It contains stems, surface variants, word formation linking elements, suffixes and prefixes. It is used to analyse an unknown string and to deliver the set of all possible segmentations.
- **Word formation rules transducer**
This transducer contains the information on word formation rules and is used to identify segmentations and to match them to existing rules. By using these rules the system is able to compute target features, i.e. the category for the unknown word lemmatizer, the exact paradigm features for the more sophisticated unknown word analyzer, as well as the new citation form.

The three finite state components collaborate in the Unknown Word products. The analysis of the surface finite state automaton is used as input for the analysis of the word formation rules transducer.

Because in some cases the surface finite state automaton delivers quite a long list of segmentations, heuristics are introduced before the word formation rule transducer is applied in order to reduce the set of segmentations and to improve the performance of the analysis process.

One heuristic chooses the segmentations with the smallest number of segments as the most likely solution.

Example 4: Of the two possible segmentations of *Gründungskapital* the one with only three segments is the most likely solution: "gründung+s+kapital" (*initial capital* literally *foundation capital*) vs. "grün+dung+s+kapital" (*green manure capital*).

Exceptions are frequent for word formations presenting linking elements. This will also be considered during analysis.

Example 5: Both of the segmentations of *Examenstage* are possible words: "examen+stage" *examination apprenticeship* and "examen+s+tage" *examination days*.

4.2 Jacaranda: The Framework for Finite State Components

For the implementation of the different finite state components we used Jacaranda (Jacaranda, 1999), a customizable object-oriented Java framework.

Jacaranda, serves as basis for the realization of an extensive German grammar and a lexicon portal (Pedrazzini & Knapp, 2003). This portal provides access to grammar rules, inflection, word formation and spelling dictionaries, multilingual glossaries and spelling applications. The system contains and manages several million cross-related entities.

Jacaranda is a pure Java framework realized with the intention to provide an easy and customizable design for the implementation of different finite-state machines. The added value with respect to other implementations is the framework character of the system, which offers various flexible and customizable parts, so-called *hot spots* (Pree, 1999). The user can change the content of the finite-state machine, updating the input document with different

states and arcs relations as usual, but he can also customize e.g. the type of arcs, the kind of output, the traversal, etc., by creating an instance of the desired FSM. Jacaranda provides a significant added value for the creation of the different finite-state machines, used and combined in order to model and manage the different kinds of data.

The internal architecture of the framework is partly based on Pedrazzini, 1999), which describes the architecture by means of a collection of design patterns (Gamma et al. 1995). Some sample solutions for the analysis of multi-terms and periphrastic phrases are described in (Pedrazzini, 2001).

4.3. Implemented Solution

We handle the recognition of non-lexicalized words by an implementation of two processing levels:

- **Specification Level**
As explained above, the data is generated by the Word Manager authoring tool, which contains a complete model of word formation. With this tool, we can tune the derivation and word formation rules needed for the Unknown Word Analyzer. Fine-tuning is based on cycles of rule selection and subsequent analysis of a large text corpus. For prefiltering (see paragraph 3.3), a wide range of elements can be identified and eliminated, which are not relevant for unknown word recognition.
- **Runtime Level**
Some overgenerated results are eliminated by postfilters (see paragraph 3.4) at runtime. The framework provides a full range of possibilities for the definition and generalization of filters necessary to maximize the precision of word analysis:
 - on segmentation level, defined as regular expression, e.g.: Delete all analyses with the segmentation "*ung + los" (nouns ending in *-ung* obligatorily take *s* when combined with the suffix *-los*: e.g. *bedeutungslos*. Therefore, segmentations without *s* must be wrong)
 - on word formation rule level, where it can be specified, for which word formation rule, and under which conditions, the result must be eliminated. Example: Delete all results with WFRule Adv+Adj, if the segmentation is "gratis + *" (compounds of *gratis* with an adjective have to be written in two words)
 - on single component level, filters can be specified by means of lexeme citation form, lexeme category, and lexeme inflection rule. Example: Delete all results "*ung + los", if the first element is a noun, has a citation form "*ung", and inflects with the inflection class *-en* (cf. example *bedeutungslos* above).

The postfilters are specified in XML and are read at load time.

4.4 Postfilters and Spell Checking

At runtime level, postfilters are also used for spell checking with unknown words, in particular for the

treatment of composed or derived words that are affected by the German spelling reform. They deal with cases like:

- words written in one or two words (old: *auseinanderreißen* vs. new: *auseinander reißen* = *to tear apart*)
- orthographic phenomena related to word formation (old spelling: *Schnelllieferung*, new spelling: *Schnelllieferung* = *express delivery*).

In these cases, the postfilters add relevant spell labels to the results instead of deleting them.

4.5 Integration and Output

Based on our approach to unknown word analysis, we have developed three different products: recognizer, lemmatizer and analyzer. All three products have a public Java API that allows their integration into other programs.

- The WMTrans Unknown Word Recognizer is used to determine if an input string is a lexicalized form or a possible legal ad hoc form.
- The WMTrans Unknown Word Lemmatizer delivers the citation form and category.
- The WMTrans Unknown Word Analyzer returns a full range of (parameterizable) morphosyntactic information.

We used our Unknown Word Lemmatizer for the analysis of a large test corpus. The corpus includes a broad range of text types, i.e. fiction, newspaper texts, scientific and technical documents.

The current version of the Unknown Word Lemmatizer recognizes an average of 95.4% of words occurring in this corpus. Most of the 4.6% unknown words not recognized are proper names, foreign words and uncommon technical terms. The analysis showed that the range of unknown words can be reduced by up to 77% if decomposition is applied. These results will be additionally improved by augmenting the WMTrans products with a named entity recognizer.

5. Conclusion

This paper has shown how an existing range of products for the morphological analysis of German could be significantly improved by the integration of a component for the recognition of unknown words, employing word formation rules and filtering techniques.

In the near future these components will be additionally improved with respect to

- recognition rate
The recognition results reported in the previous chapter refer to the first release of the WMTrans Unknown Word Analysis products. We will improve these results further by
 - tuning the pre- and postfilters going through cycles of corpus analysis and filter modification;
 - embedding a component for the recognition of named entities.
- processing speed
While the WMTrans products, which exclusively rely on finite state technology exhibit outstanding processing speed, the Unknown Word products are significantly slower. We will work on the improvement of the processing speed by code

optimisation and by the application of more efficient filtering techniques.

The WMTrans components are currently available for C++ and Java (Unknown Word products only in Java), supporting the most common platforms such as Linux and Microsoft platforms. The vocabulary of currently over 250.000 entries is continuously being updated by a dedicated team of lexicographers.

The product range also includes morphological components for English and Italian, covering, for the latter, the clitic construction.

References

- Domenig, M. & ten Hacken, P. (1992). *Word Manager: A System for Morphological Dictionaries*: Georg Olms Verlag, Hildesheim.
- Gamma E., Helm R., Johnson R., Vlissides J. (1995). *Design Patterns*, Addison Wesley.
- Jacaranda (1999). Framework, published on the Web at: <http://www.dti.supsi.ch/~pedrazz/jacaranda>
- Lüdeling, A. & Fitschen, A. (2002). An Integrated Lexicon for the Analysis of Complex Words, in: *Proceedings of EURALEX 2002*, Copenhagen.
- Pedrazzini S. (1999). The Finite-State Automata's Design Patterns, WIA'98, Third International Workshop on Implementing Automata, Rouen, France, September 1998. Published in J.-M. Champarnaud, D. Maurel, D. Ziadi (Eds.): *Automata Implementation*, Lecture Notes in Computer Science, p. 213, Springer.
- Pedrazzini S. (2001). Periphrastic Inflection Clustering for Term Extraction, In *Proceedings of the Seventh International Symposium on Communication and Applied Linguistics*, Editorial Oriente, Santiago de Cuba.
- Pedrazzini, S. & Knapp, J. (2003). From E-Learning to Complete Software Development Projects: Canoo.Net and ELDIT. In: Jutz, C., Flückiger, F., Wäfler, K., Fifth International Conference on New Educational Environments, p. 309-314, Sauerländer Verlage.
- Prece W.: *Hot-Spot-Driven Development* (1999). Fayad, Schmidt and Johnson: *Building Application Frameworks*, Chapter 16, John Wiley & Sons.
- ten Hacken, P. & Domenig, M. (1996). *Reusable Dictionaries for NLP: The Word Manager Approach*, *Lexicology* 2:232-255.