A Word Analysis System for German Hyphenation, Full Text Search, and Spell Checking, with Regard to the Latest Reform of German Orthography

Gabriele Kodydek

Institute of Computer Graphics, Algorithms and Data Structures Group,
Vienna University of Technology, Favoritenstraße 9–11/186, A–1040 Vienna, Austria
kodydek@apm.tuwien.ac.at
http://www.apm.tuwien.ac.at/

Abstract. In text processing systems German words require special treatment because of the possibility to form compound words as a combination of existing words. To this end, a universal word analysis system will be introduced which allows an analysis of all words in German texts according to their atomic components. A recursive decomposition algorithm, following the rules for word flexion, derivation, and compound generation in the German language, splits words into their smallest relevant parts (= atoms), which are stored in an atom table. The system is based on the foundations described in this article, and is being used for reliable, sense-conveying hyphenation, as well as for sense-conveying full text search, and in limited form also as a spelling checker.

1 Introduction

An essential feature of the German language is the possibility to form compound words as a combination of existing words. This peculiarity requires special treatment of German words in text processing systems. To this end, we introduce a universal word analysis system which allows the analysis of all words in German texts according to their smallest relevant components, the so-called *atoms*. The notion of the atom roughly corresponds to the linguistic expression *morpheme*, which denotes the smallest meaningful unit of a language. The word analysis system consists of two major parts: the atom table and a recursive decomposition algorithm. The atom table contains a set of approximately 6000 atoms. This number suffices for the analysis of almost all German words and the most common naturalized foreign-language words. The recursive decomposition algorithm, following the rules for word flexion, derivation, and compound generation in the German language, splits words into their atoms. The word analysis system is being used for reliable, sense-conveying hyphenation (called SiSiSi from the German "Sichere Sinnentsprechende Silbentrennung"), as well as for sense-conveying full text search, and in limited form also as a spelling checker. It can also be applied to other problems that arise in the context of processing German texts, e.g. capitalization.

2 Principles of Word Analysis

In an original version [1], atoms are being classified by their functionality into prefixes (P), stems (S), and suffixes (E). Accordingly, there are simple rules for forming legal words: A single word consists of an arbitrary number of prefixes, one stem, and an arbitrary number of suffixes; a compound word consists of an arbitrary number of single words. However, this primitive grammar allows for a large number of nonsensical words (e.g. stems followed by any number of copies of the same suffix). Figure 1 illustrates the grammar with the compound word *Wortzerlegungsverfahren*, which is made up by three single words meaning "word", "decomposition" and "method":

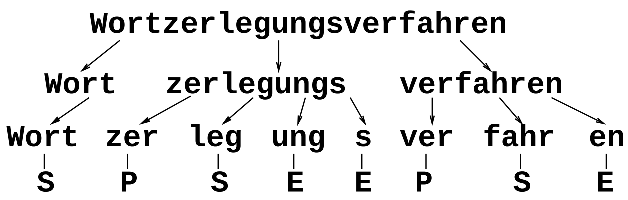

Fig. 1. Example for the decomposition of a compound word

Each atom is stored in the atom table along with a set of attributes according to its classification. An atom can be used for different purposes, e.g. *end* can be used as a stem as in *enden* (to end) or as suffix as in *gehend* (going).

The decomposition algorithm (see Fig. 2) consists of trying to find substrings of the given word in the atom table and to combine all found atoms according to the grammar rules. The outer loop tests for each possible beginning of `partial_word` whether it is an atom. If that is the case and the atom may be appended at the current position, the rest of the word is decomposed in the same way. Whether or not an atom may be appended at a certain position is determined by its atom class and the current state, which depends on the function of the previously appended atom. At the beginning of a word only prefixes and stems are allowed. Because an atom can be a member of more than one atom class all of them need to be considered one after the other. Each of the atom classes causes a recursive call with the resulting `new_state`. Inside the inner loop the results for any application specific task (e.g. marking component boundaries) can be stored in appropriate data structures. A valid decomposition is obtained when the state that is reached after processing the whole word qualifies as a final state. Note that the algorithm does not stop when one valid decomposition is found but rather looks for all valid decompositions of the given word.

Using and extending the work presented in [7, 8], the method was significantly improved by classifying atoms into word categories (e.g. noun, verb, adjective, inflective ending for nouns, inflective ending for verbs, derivative ending, and a lot of special others) and coding appropriate grammar rules for the composition

```
procedure decompose(state, partial_word, app_spec_info)
  var
    i, n, new_state: integer;
  begin
    if (partial_word is empty_string) and (state is final_state) then
      { call post-processing method for the desired application; }
    else
      n := length of partial_word;
      for i := n downto 1 do
        if partial_word[1..i] is atom then
          for atom_class in atom_classes_of_this_atom do
            if transition(state, atom_class, new_state) then
              { store application specific information app_spec_info; }
              decompose(new_state, partial[i+1..n], app_spec_info);
  end.
```

Fig. 2. Pseudo code for the recursive decomposition algorithm

of these elements. For further improvement, the stem classes are subclassified according to their inflected forms, which can be formed in different ways: e.g. the noun *Kind/Kind***er** (child/children) belongs to a different subclass than the noun *Bett/Bett***en** (bed/beds). The suffix classes are subdivided accordingly. Along with proper grammar rules this guarantees that only the correct inflection endings may be used with a stem of a certain subclass. A number of classes for derivative endings is used for changing a stem of a certain class to another class. For example, a verb stem followed by the derivative ending *ung* is further treated as a noun: *trenn/trennung* (divide/division). In the improved version, a compound word can consist of a sequence of single words with the restriction that an inflectional suffix may only be used with the last stem occurring in the compound word.

While in the original version only a few grammar rules existed, which could easily be directly implemented as part of the decomposition algorithm, the large number of grammar rules necessary for the improved version needed to be incorporated into the system in a different way. The atom classes have been combined with a number of states to rules such that they form an automaton for word decomposition. A rule has the following syntax: *start state* → *atom class* → *target state*. A special state serves as initial start state; all states that qualify as final states are marked as such.

3 Reliable and Sense-Conveying Hyphenation

Hyphenation supports the reading process by avoiding large inter-word gaps, and is therefore vital in the generation of high-quality print documents. Common hyphenation methods, e.g. the pattern method as presented by Liang [4] are based on complete dictionaries and therefore not applicable to the German language with its unlimited number of compound words. The intrinsic incompleteness of dictionaries leads to problems in the recognition of word boundaries in compound words, which in turn can lead to serious hyphenation errors. For the same reason hyphenation based on looking up words in large dictionaries does not lead to satisfying results. Other methods make use of the combinations of vowels and consonants in order to generate rules derived from grammar books [3] for finding suitable hyphen points. Often prefixes and suffixes are considered separately. Nevertheless, these methods only work well for single words since they fail to recognize component boundaries in compound words.

In contrast, the SiSiSi method of hyphenation is based on the decomposition of compound words into their building parts. Word boundaries can always be recognized in the course of the decomposition process and are immediately marked as major hyphenation points ("Haupttrennstellen", represented by "="). Hyphenation points within the single words are then found by an additional algorithm which is based on the sequence of consonants and vowels, and are marked as minor hyphenation points ("Nebentrennstellen", marked by "–"). A preferred use of major hyphenation points promotes the sense-conveying hyphenation of compound words at the component boundaries.

The main emphasis of SiSiSi lies in the reliability of hyphenation: based on the fact that all valid decompositions are determined, the set of all possible positions for hyphens is generated, e.g. *Mes–ser=at–ten–tat* (formed by the words meaning "knife" and "assassination"), *Mes–se=rat–ten=tat* (of the components meaning "mass", "rat" and "deed"); only the hyphens which occur in all variants are safe, i.e. never incorrect; any others should only be used very restrictively, e.g. after consulting the user because it is possible that such an unsafe hyphen belongs only to an unintentional decomposition.

4 Sense-Conveying Full Text Search

The search for documents containing certain keywords is often realized using *pattern-matching* methods. This method has the disadvantage that sometimes documents are found which do not meet the user's expectations (e.g., searching for the keyword *car* may find a document containing *card*). Sense-conveying full text search [2] is based on the decomposition of keywords and the words in searched text documents into the atoms which contribute to their meaning. The meaning of a compound word is determined by the meaning of its components, e.g. *Textverarbeitungssystem* (text processing system) → *text, verarbeitung, system*; the meaning of a single word is given by its stem, possibly in conjunction with a prefix, while suffixes are in general irrelevant, e.g. *ver+arbeit* (process) without the suffix *ung* (ing).

Usually inflected forms of nouns, verbs, or adjectives, are created in a manner considered regular by adding specific suffixes to the stem. There is however a considerable number of words where the stem is changed when the word is inflected: This variant of the stem is often closely related to the original stem, yet spelled differently so that it needs to be represented by a different atom:

e.g. the noun *Maus* (mouse) and its plural *Mäus*e (mice), the verb *geh*en (go) and its past tense *ging*en (went) or the adjective *gut* (good) and its comparative form *besser* (better). In these cases, particular attention needs to be devoted to relating different versions of the stem to each other, so searching for words which have the same meaning is still possible. This is achieved by the newly introduced concept of word families: a word family comprises all the different ways of spelling for a particular stem, e.g. the word family `<gehen v>` (to go, `v` denotes a verb) comprises the stems {*geh*, *ging*, *gang*}. For an irregular stem, the word analysis will therefore deliver the corresponding word family instead of the single stem.

5 Spell Checking

In a limited way, our system for word analysis can also be used as a spelling checker. If the word analysis mechanism is unable to split a given word into atoms according to the grammar rules, usually it is a construct containing an orthographical or grammatical error; the only other possibility is that its atoms are not in the table, e.g. when the word to be analyzed is a biographical or geographical name or an uncommon foreign word. In this case, the atom in question can easily be added to the atom table with its appropriate attributes; then all future combinations with this atom will be recognized. However, the system cannot detect all spelling errors because sometimes a spelling error leads to another word for which the system finds a valid decomposition even though it might not be meaningful.

6 Incorporation of the Reform of German Orthography

The 1998 reform of German orthography [6] gave rise to a wide-reaching makeover of our word analysis system. The new rules assign different spelling to some words, e.g. *rau* instead of *rauh* (rough), *Fluss* instead of *Fluß* (river). Based on the improved version of the system, the analysis algorithm and atom table were adapted in a way that allows the word analysis to comply with both the old and the new rules, as the old rules continue to be valid until 2005. For new spellings, additional atoms have been introduced; all atoms which are only valid in one set of rules have been marked accordingly. The word formation rules are not affected by the reform and can therefore be immediately applied to the new version of the word analysis system.

Hyphenation rules have also been affected by the reform. The sequence *ck*, which was formerly hyphenated as *k-k*, remains now undivided, e.g. *Bä-cker* (baker), formerly *Bäk-ker*. On the other hand *st* must not be divided according to the old rules, but may be divided now: *kos-ten* (to cost), formerly *ko-sten*. The algorithm for finding hyphens in single words was changed accordingly.

SiSiSi is able to hyphenate words according to both sets of rules. Therefore, according to the new hyphenation rules, it allows both the hyphenation according to spoken syllables as well as according to etymological considerations in certain words such as *Helikopter* (helicopter), which can be hyphenated as *He-li-kop-ter* (according to syllables) or as *He-li-ko-pter* (according to its Greek components *helix* and *pterón*). The old rules, however, only allow the latter hyphenation. Also, SiSiSi recognizes compound words where, according to the old rules, one of three adjacent identical consonants has been dropped at a component boundary, and will still get the correct hyphenation, e.g. *Schiffahrt* (navigation) → *Schiff=fahrt*. The new rules do not require such special treatment as the rule for dropping consonants has been eliminated.

7 Outlook

The word analysis system has been tested on large text files, using a newly developed test environment, see [5]. Specifically developed test methods were used to filter the few problematic cases from the huge number of analyzed words. Based on the test results, the atom table was extended by some missing, mostly foreign-language, stems. In its current state, SiSiSi can be used for pre-hyphenation of texts. Plans to directly incorporate the SiSiSi algorithm into the TEX type setting system are underway. However, the adaption to TEX is not trivial because the concept of reliable hyphenation sometimes requires user interaction to correctly identify the intended meaning of ambiguous words. The word analysis system can readily be adapted to other languages that have to deal with compound words, such as Dutch. In general, only the language-specific parts of the system, i.e. the atom table and the rules, which are stored as text files, have to be replaced for this purpose.

References

1. Barth, W., Nirschl, H.: Sichere sinnentsprechende Silbentrennung für die deutsche Sprache. Angewandte Informatik 4 (1985) 152–159
2. Barth, W.: Volltextsuche mit sinnentsprechender Wortzerlegung. Wirtschaftsinformatik, vol. 32, no. 5 (1990) 467–471
3. Duden "Grammatik der deutschen Gegenwartssprache" (Duden Band 4). Fourth edition, ed. Günther Drosdowski, Bibliographisches Institut, Mannheim (1984)
4. Liang, F. M.: Word Hy-phen-a-tion by Com-put-er. PhD thesis, Department of Computer Science, Stanford University, Report No. STAN-CS-83-977 (1983)
5. Schönhacker, M., Kodydek, G.: Testing a Word Analysis System for Reliable and Sense-Conveying Hyphenation and Other Applications. To appear in Proc. of the Third Int. Workshop on Text, Speech and Dialogue, Brno, Czech Republic (2000)
6. Sitta, H., Gallmann, P.: Duden, Informationen zur neuen deutschen Rechtschreibung, ed. Dudenredaktion, Dudenverlag, Mannheim (1996)
7. Steiner, H.: Automatische Silbentrennung durch Wortbildungsanalyse. PhD thesis, Institute of Computer Graphics, Vienna University of Technology (1995)
8. Steiner, H., Barth, W.: Sichere sinnentsprechende Silbentrennung mit Berücksichtigung der deutschen Wortbildungsgrammatik. Tagungsband Konvens'94, ed. H. Trost, Vienna (1994) 330–340

This project was in part supported by *Hochschuljubiläumsstiftung der Stadt Wien* under the grant number H-75/99.