http://spilplus.journals.ac.za/

A MORPHOLOGICAL PARSER FOR AFRIKAANS

L.G. de Stadler and M.W. Coetzer
University of Stellenbosch

1. Introduction[1]

South Africa is a complex society experiencing a taxing period in its history. In the new South Africa there are many problems to be addressed if the country wishes to survive the times that lie ahead. Two of the problems pertinent to the theme of discussion at this congress are

a. the immense problem of education, and
b. the problem of cross-linguistic communication in a multilingual society.

Against this background the importance of research in the field of computational linguistics must become clear. At the moment we are experiencing a wave of interest in this field of study sweeping the country and especially the linguistic community. It is important however to state that, at this moment, in South Africa computational linguistics is a new and weakly developed field of research. Some work has been done in the fields of computational lexicography, syntactic analysis, morphological analysis and speech synthesis.

2. Text-to-speech systems

Some years ago the communication industry in South Africa saw the need to start developing sophisticated communication systems, one type being systems linking text and speech (in both directions) for the different languages, including Afrikaans, the third largest language in South Africa.

During the past few years researchers at the University of Stellenbosch have been developing a text-to-speech system for Afrikaans (as well as other languages such as Xhosa). The main components of the system include

[1] This paper was first presented at the International Coling Conference held in Helsinki in August 1990. Due to a decision on a partial boycott of my participation only the abstract of the paper was published.

a. phonetic transcription of the written text,
b. synthesis of a speech signal from the phonetic transcription, and
c. provision of the correct stress contours to the result of b.

There are at least two approaches to the problem of phonetic transcription (see also MITalk, on which our system was modelled at first):

a. the use of rules transforming text into speech, or
b. the use of a lexicon consisting of as many items as possible, each listed with its own phonetic transcription.

The researchers opted for option a. simply because it is the more intelligent option and because option b. would imply developing a lexicon of vast proportions, something which would have a very adverse effect on the speed of the program and the possibility to convert unrestricted text.

A system of approximately 120 rules were developed. It soon became clear, however, that the rule component alone would not provide an acceptable success ratio. Option a. had to be enriched by developing a combination of components such as one finds in a true grammar of language. These components include

a. the rules transforming written text into speech;
b. obligatory phonological rules, morphonological rules and spelling rules;
c. a morpheme dictionary combined with a morphological parser;
d. syllabification rules;
e. a syntactic parser (phrase level); and
f. stress rules.

In this demonstration the attention will focus on the morphological component, a part of the system known as MORFON.

3. The morphological parser

The MORFON subsystem is made up of different components including (a) a morpheme dictionary, consisting of approximately 12,000 monomorphemes and a number of morphological complexes of

Romance origin, each marked for its syntactic category and for certain permissible morphological derivations and inflexion; (b) an algorithm to do a complete search for all possible morpheme combinations in a word form; (c) a number of word formation rules to act as a filter to test a morphological analysis; (d) a number of morphonological rules to account for different realizations of the same morpheme; and (e) a number of spelling rules to account for spelling changes such as in the case of the doubling of the *t* in the plural form *katte* (*cats*).

The dictionary of approximately 12,000 items include the following subclasses: stems, affixes and a number of morphological complexes of Romance origin. They include forms such as *aborsie*, *abortief*, *agnosties*, *agnostisisme*, etc. The Romance loan words are included because in many cases their morphology is rather complex and sometimes not systematic to a high degree. Romance affixes more often than not combine with bound stems/roots with variable forms. Furthermore Romance forms are not productive in Afrikaans, thus facilitating the option that we took, namely to list them in the dictionary. In those cases where there is a systematic relationship between derived forms, only one category is listed while the other is derived by a morphological rule, as is the case with derivations with -eer and *-asie* in word forms such as *abdikeer*, *abdikasie*; *adapteer*, *adaptasie*; agiteer, *agitasie*; etc. In cases such as these, only one subcategory is listed in the dictionary (in this case word forms ending in *-asie*), while the other subcategory is derived by a rule.

The items in the dictionary are marked for syntactic category, permissible derivations and, in the case of affixes, the applicability of morphonological rules. The entries also give information on phonetic representation, stress and syllabification. Leaving the phonological information aside, an entry would look something like the following:

http://spilplus.journals.ac.za/

```
Stems
     absorbeer v#volzm
     abstraheer v#vlokzm
     abstrak a#cmeqA
Affixes
     aar A#cbgisf|vke
     aard A#bfs|TDE
     aardig A#cmesq|TDE
```

with the symbols before # denoting the syntactic category, the symbols after # the permissible derivations and, in the case of the affixes, the symbols after | denoting the applicability of certain morphonological rules (including the spelling rules) associated with that affix.

The word formation rules include rules such as the following:

```
-aar      --> n,[n,v].
-aard     --> n,a.
-agtig    --> a,[n,a,v].
```

(symbols before the comma denote the resultant syntactic category, symbols after the comma denote the syntactic category of the base)

These rules function as filters designating permissible derivations.

Morphonological rules include rules such as

a = |K|Masie^0| { derivation with -asie }
example: *inisieer* x *inisiasie*

d = |A|Md^0| { d-insertion }
example: *konklusie* x *konkludeer*

e = |A|Me^0| { e-insertion }
example: *waarde* x *waardig*

h = |K|Mhede^0| { -hede-plurals }
example: *sekerheid* x *sekerhede*

n = |n|D1| { n-deletion }
example: *bewoë* x *bewoënheid*

r = |K|Maris^0| { derivation with -aris }
example: *militêr* x *militaristies*

s = |w|D1 Mf^0| { w-devoicing }
example: *graaf* x *grawe*

t = |A|Mt^0| { t-insertion }
example: *elegant* x *elegansie*

u = |T|D1| { t,d,e-deletion }
example: *hou* x *houdbaar*, *anargis* x *anargisties*

(*In the final programme, written in* Modula 2, the rules were not entered in this format.)

and the spelling rules include

k = |VK=|D1| { consonant singling }
example: *kat* x *katte*

v = |KUK|I2^1| { vowel doubling }
example: *skool* x *skole*

w = |KU|I1^0 Mt^0| { vowel doubling and t-insertion}
(a combination of spelling and morphonological variation)
example: *demokraat* x *demokrasie*

The symbols A, K, T, U and V denote the following groups of characters:

A = |abcdefghijklmnopqrstuvwxyz|
K = |bcdfghjklmnpqrstvwxyz|
T = |tde|
U = |aeou|
V = |aeiou|

The characters are categorised in this way to define subcategories as they participate in different processes. For instance, in Afrikaans the character /i/ does not take part in the spelling convention of vowel doubling, justifying category U.

The rules are identified by a lower case character. The symbols to the right of the first |-marker specifies the environment of the rule and those after the second |-marker describes the rule itself, with I = insertion, D = deletion and M = merge. The numbers identify the characters by numbering them from right to left in the word. Example: I2^1 Mt^0 can be translated as "Insert the penultimate character before the last character."

The morphological analysis is based on the principle of recursive matching from the left and the right, searching for the largest morpheme starting at the auslaut of the word and matching them recursively with morphemes in the dictionary until a perfect fit has been established, taking into account the restrictions posed on the whole process by the information in the dictionary, the filter function of the word formation rules and the applicability of the morphonological rule component.

The algorithm first of all duplicates the word to be analysed. All actions are then applied to the duplicate, doing a matching operation recursively from the right. Applied to an example like *onversekerbare* (Eng. *uninsurable*), the algorithm would immediately try to match -e to some morpheme in the dictionary, finding the attributive -e. This morpheme will trigger the application of either vowel doubling or consonant singling (spelling rules). The environment does not satisfy the environment for consonant singling, but vowel doubling is applicable resulting in the identification of the suffix *-baar*. The recursive matching mechanism goes ahead and identifies the morphemes *seker*, *ver-* and *on-*, while keeping in mind the applicability of morphonological rules along the way, resulting in a perfect fit. Because the system is designed to do a complete search, it repeats the whole procedure, starting a new matching sequence identifying the same or a larger chunk of the input word form, in this case identifying the plural *-e*, followed by vowel doubling and the eventual identification of the noun *aar* (Eng. *vein*, *ear* (corn)). Eventually, the recursive matching mechanism will not succeed in providing a perfect match for the remaining chunk of the input word form (i.e. *onversekerb-*), causing the system to discard this option. This procedure is followed through until all possibilities are accounted for.

Quite often more than one analysis results. A choice is then made on the grounds of morphological complexity: the system will choose the least complex analysis, by first of all checking the number of compounds and then the number of derivations constituting the complex word form. This option was chosen on statistical grounds.

4. Problems

As can be imagined, the system is not without its problems. The most taxing problems are those that will have to be solved by incorporating more semantics. This is the case in examples such as *reklameer* (Eng. *protest*), analysed by the system as *rek* (Eng. *elastic*) + *lam* (Eng. *lamb*) + *eer* (suffix) or *vlieëry* (Eng. *flight*, *flying*), analysed by the system as *vlieë* (*flies*) + *ry* (*ride*).

5. Perspective

At this very moment the system has proven itself to such an extent that it is being incorporated in different larger systems, such as text-to-speech conversion systems and reading systems for the blind. However, a lot of work still has to be done to deal with problems such as those mentioned in the previous paragraph.

http://spilplus.journals.ac.za/

Bibliography

Allen, J. 1985. "Speech synthesis from unrestricted text." In: F. Fallside & W.A. Woods (eds.). *Computer Speech Processing*. Prentice-Hall, Englewood Cliffs.

Allan, J., M.S. Hunnicutt & D. Klatt. 1987. *From text to speech: The MITalk system*. Cambridge University Press, Cambridge, London, New York.

Combrink, J.G.H. & L.G. de Stadler. 1987. *Afrikaanse Fonologie*. Macmillan S.A., Johannesburg.

Karlsson, F. 1989. "Computer-aided description of language systems III: Testing of rule systems." In: Bátori, I.S. et al. *Computational linguistics: an international handbook*. Walter de Gruyter, Berlin, New York.

Klatt, D.H. 1980. "Software for a Cascade/parallel formant synthesizer." *Journal of the Acoustical Society of America* 67:991-995.

Klatt, D.H. 1982. "The Klattalk text-to-speech system." *IEEE ICASSP*: 1589-1592, Paris.