A Morphological Parser for Swedish Regular Nouns

Victor Nilsson
Department of Computer Science
Lund University, Sweden
victor.nilsson.728@student.lu.se

Abstract

This report describes a simple morphological parser for Swedish regular nouns. It is based on a finite-state transducer (FST) designed by Marcus Uneson to represent a few typical cases of inflection. The parser is implemented in Prolog.

1 Introduction

A morphological parser analyses a single word and gives information on what components the word is made of. Given an inflected word, it should find the base form and what inflections it has undergone. This process is also called lemmatization, which refers to transforming a word into its canonical dictionary form. The morphological analysis can also be used to find out the grammatical features of an isolated word.

For English, the same has often been accomplished with a lexicon that simply lists all word-forms. This is not suitable for many other languages which may have lots of inflectional forms for each word. Therefore, a good parser need to use the morphological system of the language.

The FST described in this report was manually designed and hand-coded, and was not meant to be scalable. However, it has shown to give fair results and could be improved with little effort.

2 Morphology

The interest of morphology is *morphemes*, which are combined to form words. There are two kinds of morphemes, lexical and grammatical. Lexical morphemes correspond to the word stems; grammatical morphemes are either grammatical words or affixes that are added to the stem.

3 Morphological Parsing

Morphological parsing consists in splitting a word into morphemes. For example, *pojkarna* is parsed as *pojke+ar+na*. The *surface form* of a word – the word as it appears in a text – is transformed into its *lexical* or *underlying form*.

The lexical form may be represented as a concatenation of the stem and its grammatical features instead of morphs (the inflectional suffixes). As far as inflection is concerned, this yields more useful information. The parser output for *pojkarna* could then be *pojke+Noun+Plural+Definite*. The following examples still use morphs, as they are more easily described.

3.1 The Two-Level Model

Many morphological parsers adopt the two-level model of morphology first presented by Kimmo Koskenniemi in 1983. In this model, a word is represented with a letter-for-letter correspondence between its lexical form and its surface form. Null symbols are used to maintain alignment and reflect letter deletion or insertion. Here is the two-level representation of *pojkarna* (zeros are used for nulls):

Lexical form:	`pojke+ar+na`
Surface form:	`pojk00ar0na`

This model enables mapping in both directions, from lexical to surface form (generation) and from surface to lexical form (parsing).

3.2 Finite-State Transducers

Finite-state transducers (FSTs) are commonly used to implement the two-level model. They are automata that translate one string into another. Arcs are labeled with an input symbol and an output symbol. When a transition occurs on an arc, the input symbol is transduced into the output symbol.

FSTs which take lexical forms as input and translate into surface forms can easily be inverted in order to be used for parsing. The input and output symbols only need to change places.

4 Swedish Noun Morphology

Swedish nouns are inflected for number (singular or plural), definiteness (definite or indefinite) and case (nominative or genitive). There are two genders, neuter and common noun.

Swedish nouns may be categorized into five *declensions*, which are recognized from the plural morph.

5 The Parser

The parser consists of an FST, a simple wordlist, and some predicates that operate on these. Parsing is carried out independently of the wordlist, which is only used to filter out non-existing words.

The lexical form is represented by the lemma followed by an atom with the concatenated features (see Figure 1).

The wordlist was constructed from LEXIN and is basically a list of valid lemmas.

6 Testing

In order to test the correctness of the parser, I provided it with input from the SUC 2.0 corpus.[1] If the parser included the correct lemma among its suggestions, that was counted as a correct parsing. The grammatical features were not considered, which would have been better.

6945 nouns from different texts were used. With the original FST, which had 40 states and 60 arcs, 90.45% of the words could be parsed (after the parser was made case insensitive). With my improvements the FST had 55 states and 86 arcs, and could parse 95.97% of the words.

[1] Stockholm-Umeå Corpus 2.0

7 Conclusion

Considering its size, the parser was rather successful. There is a lot of ambiguity even with the wordlist as it only contains the words (gender could be used for example).

References

Marcus Uneson. 2005. *Morphological Analysis of Swedish Regular Noun Inflection using FST*. Dept. of Linguistics, Lund University, Sweden.

Pierre M. Nugues. 2006. *An Introduction to Language Processing with Perl and Prolog*. Springer.

```
?- cl_parse('pojkarna', no_wordlist).
pojkarna pojke N+PL+DEF+NOM+UTR 0 0 0 0 11 1 2 5 7 8
pojkarna pojkarna N+SG+INDEF+NOM+UTR 0 0 0 0 0 0 0 14 18
pojkarna pojkare N+PL+DEF+NOM+UTR 0 0 0 0 41 42 45 7 8
pojkarna pojk N+PL+DEF+NOM+UTR 0 0 0 11 1 2 5 7 8

?- cl_parse('pojkarna', wordlist).
pojkarna pojke N+PL+DEF+NOM+UTR 0 0 0 0 11 1 2 5 7 8
```

Figure 1. Parser output without and with wordlist.