Morphar: A Morphological Parser for Swedish

Thomas Raneland
Lund Institute of Technology,
University of Lund
d00tr@efd.lth.se

Abstract

This paper introduces the model and implementation for Morphar, a morphological parser for Swedish. The parser approach is intended to be as simple and natural as possible, taking advantage of the characteristics of Swedish morphology. It is based around a lexicon and a parser inspired by compiler construction techniques. The reference implementation has shown the model to work. The program is fast and returns correct results for more than 70 % of random input words. The implementation is distributed freely for use in non-commercial applications.

1 Introduction

The purpose of a morphological parser is, given an inflected word, to analyse the word and provide the user with information on what the root word is, and what inflections it has undergone. Such a computer program may be a stand-alone tool, but is often used in conjunction with other language processing components to analyse complete texts. The parser discussed in this paper, Morphar, is intended for use in both kinds of situations.

Most morphological parsers today take the approach of the two-level model presented by Kimmo Koskeniemmi (Koskenniemi, 1997). The Morphar project takes a different approach. The intention was to use as natural a model as possible. Data structures are laid out much as in an ordinary non-electronic dictionary, with entries for each word, holding information on syntactic category and possible inflections as well as suffixes for compound forms. The parser is constructed as typical computer language compiler.

The report will discuss general morphology (chapter 2) and Swedish morphology (chapter 3). Then the theory will be used to model the morphological parser Morphar (chapter 4). Some important implementation notes are included (chapter 5). Finally, pros and cons of the model and implementation are discussed (chapter 6).

2 Morphology

Morphology is the study of *morphemes*, the minimal units of meaning in a language. There are two kinds of morphemes, grammatical morphemes and lexical morphemes. Lexical morphemes correspond to the word stems, while grammatical morphemes can be either grammatical words or affixes.

Furthermore, affixes can be divided into four groups: *prefixes* (before the stem), *suffixes* (after the stem), *infixes* (in-between parts of the stem), and *circumfixes* (surrounding the stem). Examples of prefixes are pre-, sur- and in- used as above, and examples of suffixes are -s and -ed.

In European languages, words are built up by one or more morphemes (Nugues, 2003). Often, a lexical morpheme that defines the meaning of the word is concatenated with a number of grammatical morphemes (prefixes and/or suffixes) that defines the semantic function in the phrase or meaning.

2.1 Inflection

Grammatical affixes are often added to a stem in order for the word to agree in tense, number, gender or case to its neighbour words in a meaning. This is called *inflection*.

Inflection is, in most languages, relatively predictable (Nugues, 2003). For instance, in English, plural is indicated by an -s suffix, and past tense for verbs is indicated by an -ed suffix. However, most languages include a number of exceptions from these simple rules: The plural of *sheep* is *sheep*, and the past tense form of *eat* is *eaten*.

2.2 Derivation

Another class of affixes are the *derivational* affixes. Such affixes, when added to a stem, may change the syntactic category and/or the meaning of the word. Examples of English derivational morphemes are prefixes un-, con- and suffixes -ly, -ist and -ish (Fromkin, 1998). Derivation rules can be combined (as in un-system-atic-al-ly, where un-, -atic, -al, and -ly all are derivational morphemes).

Unlike inflectional morphemes, derivation rules often have many exceptions. Furthermore, derivation is irregular; although the adjective *doable* can be derived from the verb *do*, no adjective **pleasable* can be derived from the verb *please*. There is no logical explanation for this, and hence no rule to decide when the rule may be applied.

2.3 Compounds

Combining words together may form new words. Such words are called *compounds*. The category of a compound word is the category of the last word. The last word is the only word that is inflected. However, the words may be "glued" together by *compositional morphemes*, such as in the Swedish compound *tidsmaskin* (time machine), composed of *tid* (time) *-s-* (compositional morpheme) and *maskin* (machine).

2.4 Paradigms

Since the inflectional system is rather predictable, one may construct patterns of inflections that apply to a class of words. For instance, In Swedish, many nouns with ∅-plural[1] use the -et suffix to denote definite form, and the -en suffix to denote both plural and definite form, e.g. *bord* (table), *bordet* (the table), *bord* (tables), *borden* (the tables). We may say that all such words share the same *paradigm*. The paradigm is an inherent property of the word.

[1] ∅ is the symbol for the "zero" morpheme. The ∅ morpheme does not change the textual representation of the word.

3 Swedish Morphology

The general morphology described in the previous chapter can be used directly when constructing the parser. However, most languages do not use all the possible features, and so the model can be simplified. As a first – important – example, inflections are only realized by suffixes in Swedish.

The Swedish language is built up by words from the following grammatical categories:

- Nouns,
- Adjectives,
- Pronouns,
- Numerals,
- Verbs,
- Adverbs,
- Prepositions,
- Conjunctions,
- Subjunctions,
- Interjections.

These categories should be well known, and so for the rest of this chapter, I will concentrate on special cases for Swedish and inflections.

3.1 Nouns

Swedish nouns may be inflected to agree in number, definiteness, and case. Number can be singular and plural, definiteness is definite or indefinite and case is either normal form or genitive.

An inherent property of Swedish nouns is gender, which may be neuter or the "common" gender (Dalgish, 2003).

The paradigms for nouns are called *declensions*. Swedish nouns can be categorized into four declensions: -or, -ar, -er, and ∅ declension (Hellberg, 1978). Each declension has several exceptions. Additionally, words may not belong to any of these declensions. Most such words are borrowed from

other languages, e.g. English (*musical, cocktail*) and Latin (*examen, spektrum*).

3.2 Adjectives

Adjectives may have comparative forms: positive (the normal form), comparative, and superlative form. Depending on the function of the adjective, it may be inflected to agree with the noun (or pronoun) in number, gender and definiteness. The rules are rather complex, and there is no need to go into detail on these issues, so instead a list of possible inflections is presented. These are all the forms that an adjective can take:

- Common gender form,
- Neuter form,
- Plural form,
- Definite form,
- Masculine definite form,
- Comparative form,
- Superlative definite form,
- Superlative indefinite form.

The first five inflections are in the positive form. The comparative form cannot be further inflected. The superlative form may be definite or indefinite.

As you can see, definite singular positive form may be in the normal form or in *masculine* form. If the sex of the noun is masculine, the sex of the adjective may (but need not) be masculine. Examples: *den vackre/vackra mannen* (the beautiful man), *den vackra kvinnan* (the beautiful woman), *den vackra stolen* (the beautiful chair). As you might have noticed, sexless nouns always use the non-masculine form.

Many adjectives may be compared by adding *-are* and *-ast* to the normal form to get the comparative and superlative forms. These are the regular adjectives. Other adjectives are irregular, e.g. *liten–mindre–minst* (small–smaller–smallest) and *gammal–äldre–äldst* (old–older–oldest). Finally, many adjectives are compared periphrastically, e.g. *handikappad–mer handikappad–mest handikappad* (handicapped–more handicapped–most handicapped) or not at all, e.g. *död* (dead), *blind* (blind), and *tom* (empty). (Above examples taken from Stroh-Wollin, 1998.)

3.3 Verbs

The dictionary form of the verb is called the infinitive form. This form is often preceded by the infinitive marker (*att skriva*, to write). Verbs are inflected by tense (present, past, and supine) and mood (indicative, imperative, conjunctive), where the indicative form is the normal form. Conjunctive forms may be in the present or past form. The present form is no longer used. Therefore, the past conjunctive may be denoted just "conjunctive".

The indicative forms may be divided into active and inactive. Active form is the normal form. Passive form is constructed by adding an -s to the stem, e.g. *skrämma–skrämmas, skräm–skräms, skrämde–skrämdes*.

The paradigm for verbs are called *conjugations*. There are four conjugations in Swedish. Conjugation 1–3 are weak and are easy to implement. The 4th conjugation is strong and may include ablauts.

Participle

Participle is sometimes considered to be a grammatical category of its own. In this theory, participles are considered to be inflected verbs. The present participle takes one form in Swedish. The past participle is inflected on gender and number. The three forms are common gender, neuter, and plural.

3.4 Other categories

The other grammatical categories are considered indeclinable in this theory. Despite this, some pronouns (e.g. possessive pronouns) are inflected to agree in gender and number just like adjectives, e.g. *min–mitt–mina*, (my/mine). Also, numerals can be divided into cardinals and ordinals. Despite this, each word in the closed categories is considered to be a lexeme in its own right.

4 The Morphar Model

The Morphar morphological parser is a lexicon-based compiler-inspired system. Every non-compound word is described in the lexicon, including all inflectional endings. When the system analyses a word, the word is looked up in the lexi-

con. During look-up, each part of the word is replaced by its description. At the end, we have a structured morphological description of the complete word. There are some differences between the work of an ordinary computer language parser and the morphological parser Morphar. These are described in detail in 4.3.

4.1 The Lexicon

The lexicon has a number of entries, one for each lexeme. The lexeme can be retrieved by its stem. Here, the *stem* is the longest common beginning of all forms (inflected on tense, number, gender etc.) of the lexeme. There may exist zero-length stems.

Every lexeme consists of the *lemma* (the "canonical" form of the word, e.g. the infinitive for verbs or the singular indefinite of the noun) and some inherent properties. One inherent property is the syntactic category. Another is the paradigm.

The paradigm may be shared among all words with the same exact inflected forms, or may be known by a single word only. In the Morphar system, the paradigm consists not only of the inflectional endings (suffixes), but also the *compositional endings*. The compositional endings are all the possible compositional morphemes that are used when the lexeme is the non-last word of a compound. As an example, the word *boll* (ball) has the inflectional ending -s, as in *fot-bolls-skor* (football shoes). It can be noticed that *boll* also have the inflectional ending ∅, as in *boll-plan* (ball park). The use of inflectional endings is not completely arbitrary, but the rules can be quite complex and there is little need to constraint the parser to valid compositional endings only.

4.2 Creating The Lexicon

To make the lexicon memory efficient, we need to keep track of the paradigms created, in order to share the paradigms between words as far as possible. Introducing the syntactic category entity, which is nothing but a list of paradigms, does this. We need one list for each syntactic category.

A paradigm can be constructed if we know all the forms of a lexeme. After computing the stem and all suffixes, we may compare with existing paradigms. If a paradigm matches, we use it. Otherwise, we create a new paradigm with the new suffixes.

The standard paradigms for nouns (declensions) and verb (conjugations) may be added beforehand. In that case, we are able to add more information, for instance gender and compositional endings.

4.3 The Analyser

The analyser used in the Morphar system works almost like a computer language parser. One difference, however, is that each input string may result in several abstract syntax trees. Another difference is that an ordinary parser returns a tree structure for each possible interpretation, but the Morphar analyser returns only a list. So, instead of one syntax tree, we may get several *morpheme lists*.

The analyser computes all the possible stems of a word (that is, all initial substrings of the word) and for each retrieves the lexemes with matching stems in the lexicon. If a lexeme is found, the analyser tries to inflect it using its paradigm to match the input word. If a match is found, it is added to the list of results. Then, if possible, the analyser adds a compositional ending to the stem and concatenates the result so far with the results of the analyse for the rest of the word. In short, the algorithm can be described in the following way:

1. Find all stem candidates.
2. Find every lexeme that has a stem equal to the stem candidates.
3. If a lexeme can be inflected to match the input word, we have found a morpheme list (syntax tree).
4. If a lexeme has a compositional ending that matches the part following immediately after the stem in the input word, repeat recursively from step 2.

When all morpheme lists are found, we should sort them on probability. The user (or client program) may then use a first-N algorithm to consider only the N most probable analyses.

5 Implementation notes

The Morphar reference implementation is programmed entirely in Java. Each model entity (lexicon, lexeme, inflectional ending, paradigm, syntactic category, analyser etc.) is implemented as a class.

The source of the lexicon is the word list used by *Den stora svenska ordlistan*[2]. The word list is distributed under the Creative Commons Share-Alike 1.0 license[3]. The source currently consists of about 25.000 lexemes.

The lexicon is built by a hash table holding lists of lexemes sharing the same stem. The stems are the keys in the table.

The result is returned in a tree structure. The structure can be printed to a PrintStream or a Writer, which can be directed to the console or a text field in a graphical user interface. The abbreviations used in the output is the same as in the Stockholm-Umeå Corpus (SUC) of Written Swedish[4].

The implementation contains two user interfaces. The first is a simple console program, which prompts the user for input and displays the result (the input can be specified on the command line as well). The second implementation is a graphical user interface (GUI) written using Java Foundation Classes (JFC). The GUI can be run stand-alone[5] or as an applet. However, since applets in web browsers are disallowed to access files on disk, the applet version can be run from the AppletViewer tool only . The AppletViewer is included in the Sun Java SDK release.

6 Pros and Cons of The Morphar Model

The Morphar model is simple, yet efficient. The analyser is fast. Words are analysed in milliseconds. The program returns the correct analysis sorted first in 70-80 % of real-world random input words. The reference program loads in under three seconds on any standard performance PC (around 1.000 MHz and 256 MB of internal memory).

Some disadvantages compared to the standard two-level model has shown to exist. In the model, there is no support for derivational morphemes. However, such support can be added without significant changes to the model. Also, the reference implementation returns too many incorrect results. This can be avoided by adding post-analysis rules, as is done in computer language compilers. The rules could prescribe that only a subset of the lexemes and the forms can exist in compounds, and words of syntactic categories A and B cannot be combined to form compound words.

For the reference implementation, it is a shortcoming that the source word list contains only around 25.000 words. Also, information on compositional endings is missing in the source and is added by hand when creating the standard paradigms for nouns and verbs.

As of today, paradigms cannot handle umlauts and ablauts. The 4th conjugation for verbs must thus be treated as many paradigms, one for each lexeme. This, of course, is not a disadvantage of the model, but of the implementation.

7 Conclusion

The Morphar model has proven to be useful and efficient. The reference implementation works satisfactory in most situations, and is pretty fast too. Some features are missing in the model, first and foremost a rule-based filter to remove incorrect analyses. Also, the handling of derivational morphemes is yet to be modelled, and the paradigms need to be refined.

The implementation is working and may be distributed freely for use in non-commercial applications.

Acknowledgement

Thanks to Richard Johansson, Department of Computer Science at The Institute of Technology, University of Lund. Without his supervision the Morphar model would have been less than good.

References

Gerard M Dalgish. Teaching the Computer Swedish: Morphology and Phonology. *CALICO Journal, Volume 7 Number 4.*

Victoria Fromkin and Robert Rodman. 1998. *An Introduction to Language, 6th edition.* Harcourt Brace & Company, Orlando, FL.

Staffan Hellberg. 1978. *The Morphology of Present-Day Swedish.*

Kimmo Koskenniemi. 1997. Representations and Finite-State Components in Natural Language. In Roche, E. and Y. Schabes, editor, *Finite State Natural Language Processing.* MIT Press, pp. 99-116.

[2] Created by Tom Westerberg. See http://sv.speling.org for more information.

[3] The license can be found at http://creativecommons.org/licenses/sa/1.0.

[4] see Appendix A.

[5] see Appendix B.

Pierre Nugues. 2003. *Morphology and Part-of-Speech Tagging*. Draft version.

Ulla Stroh-Wollin. 1998. *Koncentrerad nusvensk formlära och syntax*. Studentlitteratur, Lund, SE.

A. SUC Abbreviations

Category Code	Category
AB	Adverb
DL	Delimiter (Punctuation)
DT	Determiner
HA	Interrogative/Relative Adverb
HD	Interrogative/Relative Determiner
HP	Interrogative/Relative Pronoun
HS	Interrogative/Relative Possessive
IE	Infinitive Marker
IN	Interjection
JJ	Adjective
KN	Conjunction
NN	Noun
PC	Participle
PL	Particle
PM	Proper Noun
PN	Pronoun
PP	Preposition
PS	Possessive
RG	Cardinal Number
RO	Ordinal Number
SN	Subjunction
UO	Foreign Word
VB	Verb

Feature Code	Feature	
UTR	Common (Utrum)	Gender
NEU	Neutre	Gender
MAS	Masculine	Gender
SIN	Singular	Number
PLU	Plural	Number
IND	Indefinite	Definiteness
DEF	Definite	Definiteness
NOM	Nominative	Case
GEN	Genitive	Case
POS	Positive	Degree
KOM	Comparative	Degree
SUV	Superlative	Degree
PRS	Present	Verb Form
PRT	Preterite	Verb Form
INF	Infinitive	Verb Form
SUP	Supinum	Verb Form
IMP	Imperative	Verb Form
AKT	Active	Voice
SFO	S-form	Voice
KON	Subjunctive	Mood
PRF	Perfect	Perfect

B. Screen Dump from The Reference Implementation