Heiki-Jaan Kaalep, Tarmo Vaino
Tartu

Complete Morphological Analysis in the Linguist's Toolbox

1. Linguist's toolbox

A linguist uses various kinds of linguistic data – both text corpora or text collections and dictionaries. We will describe those tools that are at the disposal of a linguist who uses textual material in electronic form.

The latter is now much more available than ever before. But what can one do with texts that are so numerous that one will never be able to read them through? The solution is obvious - a linguist must somehow filter out the material from the text mass that is of interest to him or her. At this it would be useful to present (group) the material so that manual checking, which is inevitable, would be easier.

Thus the linguist needs some software. At the same time in a typical case, an interesting linguistic problem has a non-standard nature, and usually there is no ready-made computer program at hand. One might even say that a ready-made program may be available only for the treatment of an uninteresting problem. Now the linguist has two options: to ask an IT-expert to write the necessary program or to write it himself. Neither option is attractive. In the first case, it is natural that communication problems will arise between the IT-specialist and the linguist. The former requires precise formulations, but the latter, solving a linguistic problem, is understandably too confused for froviding them. Consequently, the IT-specialist has to redo the program over and over again… On the other hand, if the linguist starts to write the program, he will, as a rule, spend much more time than the IT-specialist, whereas his own specific skills will remain unused.

In our opinion, the Lego software approach is a good way to solve this problem. The underlying rationale is that a specific task can be solved by combining a small number of standard programs, similarly to the way Lego blocks are used to build complex systems.

We are not trying to provide an exhaustive list of the tools that are necessary for the linguist but will point out only some simple ones. At this we will rely on the years-long experience of our colleagues and ourselves, which has sifted out the widely used tools.

We use the UNIX platform. There are a number of reasons why we use it. First, UNIX has a wide selection of commands that can be used for text processing. Second, these commands can be combined very easily. And last but not least, the UNIX platform is stable – a person who learned to use UNIX in 1980 can still use his skills in 2000 and possibly in the future as well. In contrast, the one who learned to use DOS or Windows has to relearn certain things after every couple of years. It is evident that constant relearning hampers specialization.

So, let us take a look at the blocks or commands from which the filters are made for the linguist.

1. grep. It enables us to retrieve all the lines from the text that contain the word, expression, or their combination. For example, we can retrieve all the words ending in *–ma* that are not verbs or all the corpus lines that contain *poole rohkem* 'twice as many'.
2. sed. This one makes it possible to convert lines. For example, when studying author's speech one can remove from all the lines strings that are enclosed by quotation marks.
3. tr. This one enables us to convert individual letters more conveniently than *sed*, e.g. upper-case letters into lower-case letters.
4. sort. This one makes is possible to sort lines.
5. head. This one enables us to extract lines that are of interest to us from the beginning of a file.
6. tail. This one enables us to extract lines that are of interest to us from the end of a file.
7. paste. This one makes it possible to put together lines like columns in a table.

8. join. This one enables us to put together tables of sorted lines (similar to a relational database where the columns of a table are put together according to the key). This is convenient for combining various dictionaries.

All the commands can be combined so that the output of one command serves as the input for the next without the need to write anything in the file in the meanwhile. There are many textbooks and UNIX manuals describing how to use UNIX commands and how to combine them, so we will not discuss these in detail here. Perhaps the best source describing how a linguist can use UNIX commands to his or her best advantage is Ken Church's manuscript "Unix for Poets" (Church). This paper has largely inspired us.

It is clear that although the standard tools are good, one would still need some commands that are specifically intended for linguists, for example, to establish either sentence boundaries or to present results as KWICs. It is good if they can be used as building blocks similarly to UNIX commands.

In the case of Estonian there is no doubt that the morphological analyzer serves as a necessary 'building block' - a program that can provide the base form, structure (formatives), and morphological information (e.g. part of speech, case, or person, etc.) for any word occurring in any form in the text.

A typical sequence of tasks that a linguist uses for the extraction of linguistic data could be as follows:

Text -> establishing sentence boundaries -> morphological analysis -> grouping, sorting, etc.

2. The complete morphological analysis of text

The article focuses on an important tool in the study of Estonian – the complete morphological analyzer of text. It is a program where the text serves as the input and the output consists of the morphologically analyzed words, whereas it ascribes to each word those variant(s) of analysis that are suitable in the given context. In an ideal case, there would be only a single variant. Unfortunately, the present program is not perfect in that it does not allow it to a full degree.

The program that we describe is intended specifically as a linguist's tool and not to prove some theoretical principle or for illustration. Therefore, it focuses on the processing of the so-called real texts and not a carefully selected collection of words (such as a dictionary) An authentic text includes some elements that are not recorded in a dictionary: proper nouns, spelling errors, foreign-language quotations, formulae, neologisms, archaisms, slang words, dialect words, etc. A good tool should be able to interpret these somehow as well; in an ideal case, it should provide the correct analysis that would fit into the concrete context.

Traditionally, the morphological analyzer is a program that is able to analyze individual word forms. For example,

```
Mees 'man'
   mees+0 //_S_ sg n, //
   mesi+s //_S_ sg in, //
peeti 'was detained'
   peet+0 //_S_ adt, sg p, //
   pida+ti //_V_ ti, //
kinni
   kinni+0 //_D_ //
```

When we see such plurality of analyses in a concrete text, we think at first intuitively that something is wrong here – it does not even occur to a human that one is justified to generate some inappropriate variants. Therefore, it may seem to us that the computer creates too much noise. In order to comply with human intuition but also because of various practical needs of computational linguistics and language engineering, it is expedient to carry out the morphological analysis by taking into account the context, so that all the words would have a unique analysis in the output.

The complete morphological analysis of a text consists of two stages: the morphological analysis of individual words and disambiguation. In the case of Estonian, the morphological analysis of individual words is inevitable (in the case of a morphologically simpler language, such as English, it may be neglected). It provides a large number of possible variants for each word. Subsequently, the appropriate variant will be selected or disambiguated for the given context. The disambiguator under discussion treats the words in the context of the sentence; it does not examine a more extensive context. Thus, in the case of disambiguation it is assumed that the sentence boundaries have already been established. Therefore, it is

necessary to have a program – sentence splitter - that would divide the input text into sentences before the morphological analysis is carried out.

3. Sentence splitter

It might seem that it is a trivial task to establish sentence boundaries, but actually it is not so. For example, a full stop after a number, initial, or abbreviation may but need not mark the end of the sentence. Similarly, brackets, quotation marks, or some other symbols may follow a full stop at the end of the sentence, which means that the sentence may end later than the full stop.

Since the establishment of sentence boundaries is a standard task that is often used, then it is reasonable to turn it into a separate 'Lego block' that may be combined with other modules if necessary.

It is likely that the sentence splitter as an independent module may be of little interest if a linguist is interested in vocabulary. However, if he or she happens to study syntax, then sentence splitting is a necessary stage. It is also necessary when example sentences have to be found.

4. Dictionary-based morphological analysis

In order to carry out the morphological analysis of a text, one usually processes the word forms and compares them with the lexicon of the studied language. In addition, various heuristic rules are applied for those words that are absent in the lexicon.

About 98±1percentage of Estonian words in the input text can be analyzed by looking them up in a dictionary, by using various lists of morphemes and rules for combining them. This percentage is higher than in the case of English, where it is about 95% (Voutilainen *et al* 1992). The morphological analysis of Estonian is realized so that the words in the running text are compared with the combinations of lexemes in the dictionary. No two-level rules are applied to the comparison (Koskenniemi 1983) and the textual words are analyzed from the right to the left, i.e. the endings and suffixes are cut off, and the base(s) are checked with the help of the lexicon, which contains the stems of 38,000 words (67,000 items).

The main characteristics of this analysis are as follows:

1. It is intended for Written Estonian.
2. The treatment of inflection is complete; exceptional forms are analyzed as well.
3. The dictionary of the analyzer includes simple words that belong to the core vocabulary as well as more frequent proper nouns and abbreviations. The dictionary does not include those derivatives and compounds that are formed according to productive patterns.
4. The derivatives and compounds are analyzed algorithmically. Therefore, they need not be stored in the dictionary, and it is possible to provide the correct analysis of new derivatives and compounds also.
5. The algorithm for the analysis of derivatives and compounds has been composed so that for each word it would be possible to find the most probable division into components.
6. The analysis relies on the dictionary and does not include any heuristics.
7. The rate of correct analyses is about 98% for the input text. The program does not analyze any rare words such as proper nouns, abbreviations, terms, slang words, etc.
8. The analyzer takes care of the analysis of punctuation marks and foreign proper nouns that consist of a number of words (e.g. *New York*).
9. The program does not pretend to be original in the treatment of the Estonian morphological system, except in word formation.
10. The analyzer does not take into account such syntactic or semantic properties as valency, transitivity, or countability.
11. The analyzer serves as the basis for the commercial speller of Estonian.

For a detailed description, see (Kaalep 1997), (Kaalep 1998).

5. Guesser

A text contains up to 3% of words, which cannot be found in dictionaries. The percentage varies considerably depending on the register. It is highest, about 3%, in media language and in informational and reference materials. In contrast, in the case of fiction and legal texts it is often less than 0.5%.

In the case of media texts the 3% is distributed as follows: about 66% of unknown word forms are proper nouns, 10% are common nouns, 9% are punctuation marks that occur in non-standard form, (e.g. dash), 8%

are abbreviations, 1% is various combinations of numbers, 1% adjectives, verbs, adverbs, 5% are foreign words, web addresses, and other sequences of symbols for which it is difficult to offer any reasonable analysis.

Our program guesses the base form of the word and its current form only on the basis of the word form. It takes into account the final letters of the word and the number of syllables. The guesswork does not take into account the context of the word.

During the guesswork the program checks if the word could belong to one of the following categories:

1. an abbreviation (up to two letters or a vowelless 'word', a word consisting of upper-case letters with a possible attachment of lower-case case ending);
2. a spelling error, a word that will be analyzable after the mistake is corrected (e.g. there is no space between words, or there is a sequence of three identical vowels);
3. a proper noun;
4. a derivative or compound with a rare formation pattern or one that includes a simple word not to be found in the dictionary;
5. an unknown simple word – a noun or a verb (the judgement will be passed on the basis of the end of the word and the number of preceding letters and syllables).

Guesswork may be facilitated by various typographical conventions. Proper nouns, for example, begin with a capital letter. Another factor that may facilitate guesswork is that those words that cannot be found in the dictionary belong to a small number of inflectional types.

Guesswork may be more complicated because the declension of proper nouns may entail an option to decline the word according to the gradational pattern, which is characteristic of Estonian, or to retain the original shape of the proper noun. For example, it has happened that within a single newspaper article the genitive case of the last name *Fink* may appear as the gradational form *Fingi* side by side with the non-gradational form *Finki*. If a human does not know how to make up word forms, then it is only natural that difficulties may arise in automatic analysis, too, because the program has to guess which pattern of morphological formation was used.

The counting of syllables is also problematic because the number of syllables that determine the morphological properties of a word has to be counted starting with the last stressed syllable. However, stress is not marked in the orthographic text. Therefore, there may be frequent errors in the analysis and synthesis of foreign proper names because words with a formally identical structure are declined differently depending on the position of the stressed syllable. Compare, for example, *Vertov* (with stress on the first syllable) and *Petrov* (stress on the second syllable) – the partitive singular is *Vertovit* and *Petrovi*, respectively. In other words, a trisyllabic word form ending in *–ovi* may stand for the genitive or partitive case of a proper noun ending in *–ov*. The latter possibility is eliminated if word stress falls on the first syllable. However, one cannot see it in the text, and, therefore, the guesser has to offer the spurious possibilities also.

6. Errors made by the analyzer

Occasionally our program may not offer the correct variant of analysis. Below we will describe some kinds of mistakes that may occur, so that the user of the program would be warned against them and take them into account or anticipate or correct them in some *ad hoc* manner.

Experiments have shown that if the dictionary-based approach is used, the rate of incorrect analyses could reach 0.1% (the rate of correct analyses being at least 97%). Guesswork that is applied to the remaining 3% of word forms may end up with 10% of incorrectly analyzed words. Thus, all in all 0.1+0.3=0.4% of words in the input text may remain without the correct variant.

In the case of dictionary-based analysis the common reasons for errors are as follows:

1. The input text is not exactly for what the analyzer is intended for – pure contemporary written language. Therefore, some words may get a strange analysis. For example, *puitund* 'woody, lignified' is analyzed as *puit_und* '*woody sleep'.
2. A proper noun may be similar to some form of a common noun. For example, the program suggests that the base form of *Rebast* is *rebane* 'fox' although actually the base form is *Rebas*.

Guesswork may result in two kinds of mistakes: either no correct analysis is provided, or the correct analyses include some wrong analyses as well.

Some more typical errors are as follows:

1. The program suggests a wrong part of speech. For example, an upper-case word is perceived as a proper noun although it may not be so; *budjete* 'you will, Russian' is classified as a noun although it is a quotation from a foreign language (in this case from Russian). If a word consists of two letters, it is treated as an abbreviation. However, this may be misleading, especially in the case of English prepositions and Chinese names.
2. The program fails to find the correct base form, for example, in the case of a proper name *Loidi* the program may suggest that the base form is *Loid* and not *Loit*.
3. Since the shape of a word does not provide any clue about the position of word stress, the guesser may offer such base forms that look wrong to people who know the pronunciation of the word.

These are problems that cannot be solved by perfecting the analysis of individual words. It would be much more promising to examine the context and to look for some other forms of the same word in the text if an erroneous analysis is suspected. In that case, we could discover that the base form of *Rebast* could be *Rebas*, or that *puitunud* must be a verb.

7. Disambiguator

Morphological disambiguation means that each word in a morphologically analyzed sentence will get the correct morphological tag, which is selected from the range of possible morphological tags. For example, the sentence *Mees peeti kinni* 'the man was detained' was originally analyzed in the following way:

```
Mees
   mees+0 //_S_ sg n, //
   mesi+s //_S_ sg in, //
peeti
   peet+0 //_S_ adt, sg p, //
   pida+ti //_V_ ti, //
kinni
   kinni+0 //_D_ //
```

After disambiguation the sentence was analyzed in the following way:

```
Mees
   mees+0 //_S_ sg n, //
peeti
   pida+ti //_V_ ti, //
kinni
   kinni+0 //_D_ //
```

Morphological disambiguation proceeds from two premises (Merialdo 1994: 156):

1. Each word has only a small range of suitable morphological tags. This range is established by means of the morphological analyzer.
2. If a word has a number of possible morphological tags in the sentence, then the local context makes it possible to establish the only correct tag for each word.

The disambiguator that features in our toolbox is based on the Hidden Markov Model (HMM) - the probabilistic model that is described in greater detail in (Kaalep and Vaino 1998). It is based on the statistics of texts and does not use such rules in the selection of suitable tags that are intuitively understandable to a linguist. We applied the Hidden Markov Model in its purely classical form, in which case we assume that:

1. The sentence is not treated as a sequence of words but as a sequence of some special disambiguating tags (M's). They have been obtained by transforming morphological tags, and they are used primarily to enhance the operation of the algorithm.
2. Since a word may have a number of M's, then a concrete sentence may have a number of possible sequences of M's, but only one of them is correct.
3. Some sequences are typical in the given language, others are not.
4. One has to select the most typical sequence, i.e. the most probabilistic one. This is the correct sequence for the given sentence.
5. When calculating the probability of the sequence of M's for a new sentence, one proceeds from the probabilities that were in the training phase calculated on the basis of manually tagged sentences.

In brief, the algorithm for the selection of tags that are suitable for the context is as follows:

1. We transform the morphological tags into the disambiguating tags (M's).

2. We take into account two kinds of probabilities. The first probability is that some M may suit the word if we do not consider the context. For example, it is much more probable that *veel* 'still' is an adverb than a form of the word *vesi* 'water'. The second probability is that some M may suit the word if preceded by a specific M. For example, if a word is preceded by a preposition, then it is much more likely that it may be a noun than a verb. On an ad hoc basis, there exists a table of probabilities for the first words in a sentence because they are not preceded by any M.
   In order to find the probability of the sequence of M's for a sentence, one has to add up the probabilities of individual M-s. Thus, we will have a large number of alternative M sequences, from which we will choose the one with the highest sum. The respective M's are the ones that suit the words under discussion. Thus, we are looking for the best overall sequence and not for the highest probability of M for a single word. It is quite possible that in the best sequence, we may have to choose for some word an M that does not have the highest probability.
3. Finally we re-transform M's into morphological tags.

In order to cut down disambiguation errors, we have decided that in the case of very complicated situations we give up disambiguation and preserve ambiguity. Such cases make up 13.5% of input words. The more important word groups that retain their ambiguity are as follows:

1. Participles, ending in *–nud* and *–tud*. They constitute 25% of words that retain their ambiguity. It is impossible to say whether they are verbs or adjectives unless the immediate context is provided.
2. The word *ta* 'he, she; his, her', 16%. No decision is made whether the word is in the nominative or genitive case.
3. The word *on* 'is, are', 13%. No decision is made whether the word is in the singular or plural.
4. The words *kui* 'if, when' and *nagu* 'as, like', 13%. No decision is made whether they are adverbs or conjunctions.
5. The words *mis* 'what' and *kes* 'who', 13%. No decision is made whether these words are in the singular or plural.
6. The words that have different base forms but their part of speech and inflectional form are identical, for example, *mandri – manner/mander* 'continent', *lõi – looma* 'create'/*lööma* 'hit'; 4%. In this case, the output has a number of variants with different base forms. Actually, morphological disambiguation is of no help here because the problem has a lexical or semantic nature.
7. The words *üks* 'one' and *teine* 'other, second', 4%. No decision is made whether they are numerals or pronouns.
8. All other cases make up 12% of the words that retain their ambiguity.

At present about 3% of the morphologically analyzed words get a wrong analysis because of disambiguation (it concerns the stem, part of speech, or some other morphological category). The overwhelming majority of errors, one third, arise because in the case of the noun the wrong variant is selected from among homonymous case forms (the nominative, genitive, partitive, or short illative). If we are interested in the part of speech only, then the error rate is 1.7%; if we are interested only in the base form, where no word components or upper-case and lower-case letters are distinguished, then the correct version is absent in 1.5% of the disambiguated text.

8. Some Problems

Previously we described those tools that are at the disposal of a linguist. However, it is a peculiarity of linguistics as one of the humanities that its basic concepts and categories are not defined in the same way as in the sciences. It concerns morphological analysis as well – both the system of the used categories as well as what is the base form of the word after all.

At present two category systems with different degrees of attention to detail are used for the computerized morphological analysis of Estonian. One of them is based on the 'Concise Morphological Dictionary' by Ülle Viks (Viks 1992). A slightly modified version of this work o.html) is used also in the morphological analyzer under discussion; let us call it the fs-tag system. The other one bears more resemblance to such grammars as (Valgma, Remmel 1970) and (EKG 1995) as well as the categories of the international standardization project EAGLES (Monachini, Calzolari 1995), and it has been used in the manual disambiguation of texts: let us call it the kym-system.

The CG-disambiguator (Puolokainen 1998) and the syntactic analyzer (Müürisep 2000) require that a text should be tagged in the kym-system. In contrast, our disambiguator requires that it should be tagged using the fs-system.

If a text has been tagged according to one system, then its conversion to the other is fully automatic. At the same time, one has to take into account that the use of different systems, even in the existence of an automatic conversion program, may cause problems in linking program modules.

Across languages, practice has shown that the tagging system that is used for disambiguation is even more important from the point of view of accuracy than the algorithm or program itself. In case of a "bad" tagging system, a human (not to speak of the program) is unable to decide how to tag a concrete word in the text. This will result in inconsistencies in the tagged text, and the user does not know how reliable it may be.

Thus, the selection of the suitable tagging system is an important problem in morphological disambiguation. It may sound strange because Estonian morphology is a well-researched area. Actually, one has to distinguish between the morphological and syntactic categories that are applicable to Estonian in principle and the categories that can be identified uniformly in texts. The latter are much less in number. The problems involved have been described in detail in the articles by (Kaalep *et al.* 2000) and (Puolakainen 2000). It is sufficient to point out that such theoretical works as (Valgma, Remmel) and (EKG) classify Estonian words in such detail into syntactic and semantic classes that even an educated linguist is unable to establish them uniformly and consistently. In this case, one had better abandon detailed tagging, which is theoretically possible, rather than carry it out inconsistently.

Another problem in morphological analysis and lemmatization is that different linguistic tasks require different base forms. According to the Estonian grammatical tradition, the base form of a word form with regular derivation is a form that includes derivation, e.g. the base form of *minemise* ‘going, singular genitive’ is *minemine* ‘going, singular nominative’. On the other hand, according to the traditions of dictionary-making, dictionaries do not list regular derivatives as independent words. Therefore, the base form is an underived form, e.g. *minema* ‘to go’ for *minemise* ‘going, singular genitive’. The different approaches to the concept of the base form in two branches of linguistics may give rise to various problems when we wish to link automatically the linguistic data, for example, text corpora and dictionaries. Our intuition requires that a morphological analyzer should provide only one analysis for each word form that is suitable in the given context, but it appears that this is in conflict with the multitude of linguistic traditions.

References

Ken Church "Unix for Poets". MS

Eesti Keele Grammatika 1995. 1. M. Erelt (ed.); Eesti TA EKI, Tallinn.

Kaalep, H-J. 1997. An Estonian Morphological Analyser and the Impact of a Corpus on its Development. Computers and Humanities. 31: 115-133, 1997.

Kaalep, H-J. 1998. Tekstikorpuse abil loodud eesti keele morfoloogiaanalüsaator. Keel ja Kirjandus 1/1998, lk 22-29

Kaalep, H-J., Vaino, T. 1998. Kas vale meetodiga õiged tulemused? Statistkale tuginev eesti keele morfoloogiline ühestamine. Keel ja Kirjandus 1/1998, lk 30-38

Kaalep jt. 2000 Keel ja kirjandus (in press).

Koskenniemi, K. 1983. Two-level Morphology: A General Computational Model for Wordform Recognition and Production. Publications of the Dept. Of General Linguistics, University of Helsinki, 11

Merialdo, B. 1994. “Tagging English text with a probabilistic model.” Computational Linguistics, 20(2), 155-171.

Monachini M., Calzolari, N. 1995. Synopsis and Comparison of Morphosyntactic Phenomena Encoded in Lexicons and in Corpora and Application to European Languages. EAGLES document EAG-LSG-T4.6/CSG-T3.2, Pisa.

Müürisep, K. 2000 (in the present collection of articles)

Puolakainen, T. 1998. “Eesti keele kitsenduste grammatika morfoloogiline ühestaja.” Keel ja Kirjandus, 1, lk. 37-46

Valgma, J., Remmel, N. 1970. Eesti Keele Grammatika. Valgus, Tallinn

Viks, Ü. 1992. Väike vormisõnastik I. Sissejuhatus & grammatika; II. Sõnastik & lisad. Tallinn.

Voutilainen, A., Heikkilä, J., Anttila, A. 1992. Constraint Grammar of English. A Performance-Oriented Introduction. Univ. of Helsinki, Dept. of General Linguistics, No. 21