Attila Novák (2004): Creating a Morphological Analyzer and Generator for the Komi language.
Proceedings of the SALTMIL Workshop at LREC-2004: First Steps in Language Documentation for Minority Languages. Lisbon, 64-67.

Creating a Morphological Analyzer and Generator for the Komi language

Attila Novák

MorphoLogic Ltd.
Budapest, Orbánhegyi út 5., 1126 Hungary
novak@morphologic.hu

Abstract

In this paper a set of tools for morphological analysis and generation is presented along with its application to Komi-Zyryan, a small Finno-Ugric language spoken in Northeastern Europe. This endeavor is part of a project which aims to create annotated corpora and other electronically available linguistic resources for a number of small members of the Uralic language family. A morphological grammar development environment is also introduced which facilitates a rapid development of the morphological descriptions used by the tools presented.

1. Introduction

Various Hungarian research groups specialized in Finno-Ugric linguistics and a Hungarian language technology company, MorphoLogic have initiated a project with the goal of producing annotated electronic corpora for small Uralic languages. This paper describes the current state of the subproject on Komi.

2. The Tools

In the project, we use a morphological analyzer engine called Humor ('High speed Unification MORphology') developed at MorphoLogic (Prószéky and Kis, 1999). We supplemented the analyzer with two additional tools: a lemmatizer and a morphological generator.

2.1. The Morphological Analyzer

Features of the Analyzer The Humor analyzer performs a classical 'item-and-arrangement' (IA) style analysis (Hockett, 1954). The input word is analyzed as a sequence of morphs. It is segmented into parts which have (i) *a surface form* (that appears as part of the input string), (ii) *a lexical form* (the 'quotation form' of the morpheme) and (iii) *a category label* (which may contain some structured information or simply be an unstructured label). The lexical form and the category label together more or less well identify the morpheme of which the surface form is an allomorph.

Although the 'item-and-arrangement' approach to morphology has been criticized, mainly on theoretical grounds, by a number of authors (c.f. e.g. Hockett, 1954; Hoeksema and Janda, 1988; Matthews, 1991), the Humor formalism had been in practice successfully applied to languages like Hungarian, Polish, German, Rumanian and Spanish so we decided to use it in this project as well. The 'slicing-up' approach of the analyzer we use seemed suitable to the agglutinating type of languages to which Komi belongs. On the other hand, we avoided segmenting any 'portemanteau' morphemes the segmentation of which would have been purely stipulated.

Another feature of the analyzer is that it produces flat morph lists as possible analyses, i.e. it does not assign any internal constituent structure to the words it analyzes. The reason for this is that it contains a regular word grammar, which is represented as a finite-state automaton. This is clearly much more efficient than having a context-free parser and it also avoids most of the irrelevant ambiguities a CF parser would produce.

The following is a sample output of the Humor analyzer for the Komi word form *kolanla*.

```
analyzer>kolanla
kov[S_V]=kol+an[D=A_PImpPs]+la[I_CNS]
kov[S_V]=kol+an[D=N_Tool]+la[I_CNS]
```

Morphs are separated by + signs from each other. The representation morphs is `lexical form[category label]=surface form`. The surface form is output only if it differs from the lexical form. A prefix in category labels identifies the morphological category of the morpheme (`S`: stem, `D`: derivational suffix, `I`: inflectional suffix). In the case of derivational affixes, the syntactic category of the derived word is also given.

In the example above, *kov* is identified as the lexical form of a verb stem (`S_V`). The stem undergoes a stem alternation the result of which is that its surface form end in *-l* instead of *-v*. A derivational suffix *-an* is attached to it, the surface and lexical form of which is identical. The morph is ambiguous: it is either a noun forming suffix or a suffix forming a passive participle. This is followed by an inflectional suffix: the consecutive case marker *-la*.

How the analyzer works The program performs a search on the input word form for possible analyses. It looks up morphs in the lexicon the surface form of which matches the beginning of the input word (and later the beginning of the yet unanalyzed part of it). The lexicon may contain not only single morphs but also morph sequences. These are ready-made analyses for irregular forms of stems or suffix sequences, which can thus be identified by the analyzer in a single step, which makes its operation more efficient.

In addition to assuring that the requirement that the surface form of the next morpheme must match the beginning of the yet unanalyzed part of the word (uppercase-lowercase conversions may be possible) is met, two kinds of checks are performed by the analyzer at every step, which make an early pruning of the search space possible.

On the one hand, it is checked whether the morph being considered as the next one is locally compatible with the previous one. On the other hand, it is examined whether the candidate morph is of a category which, together with the already analyzed part, is the beginning of a possible word construction in the given language (e.g. suffixes may not appear as the first morph of a word etc.). The global word structure check is performed on candidate morphs

for which the local compatibility check has succeeded. Possible word structures are described by an extended finite-state automaton (EFSA) in the Humor analyzer.

2.2. The Lemmatizer

The lemmatizer tool, built around the analyzer core, does more than just identifying lemmas of word forms: it also identifies the exposed morphosyntactic features. In contrast to the more verbose analyses produced by the core analyzer, compound members and derivational suffixes do not appear as independent items in the output of the lemmatizer, so the internal structure of words is not revealed. [1]
The analyses produced by the lemmatizer are well suited for such tasks as corpus tagging, indexing and parsing. The output of the lemmatizer and the analyzer is compared in the example below:

```
analyzer>kolanla
kov[S_V]=kol+an[D=A_PImpPs]+la[I_CNS]
kov[S_V]=kol+an[D=N_Tool]+la[I_CNS]

lemmatizer>kolanla
kolan[N][CNS]
kolan[A][CNS]
```

The lemmatizer identifies the word form *kolanla* as the consecutive case of the noun or adjective (in fact: participle) *kolan*.

2.3. The Generator

Originally, MorphoLogic did not have a morphological generator, so another new tool we created using the analyzer engine was a morphological generator. The generator produces all word forms that could be realizations of a given morpheme sequence. The input for the generator is normally very much like the output of the lemmatizer: a lemma followed by a sequence of category labels which express the morphosyntactic features the word form should expose.
The generator is not a simple inverse of the corresponding analyzer, thus it can generate the inflected and derived forms of any multiply derived and/or compound stem without explicitly referring to compound boundaries and derivational suffixes in the input even if the whole complex stem is not in the lexicon of the analyzer.
The following examples show how the generator produces an inflected form of the derived nominal stem *kolan*, which is not part of the stem lexicon, and the explicit application of the derivational suffix (and the same inflectional suffix) to the absolute verbal root of the word.

```
generator>kolan[N][CNS]
kolanla
generator>kov[V][_Tool][CNS]
kolanla
```

The development environment also makes it possible for the linguist to describe preferences for the cases when a certain set of morphosyntactic features may have more than one possible realization. This can be useful for such applications of the generator as text generation in machine translation applications, where a single word form must be generated.
We also created a version of the generator which accepts the morphosyntactic category labels in any order (as if it were just an unordered set of morphosyntactic features) and produces the corresponding word forms.

3. The Morphological Database

Various versions of the Humor morphological analyzer have been in use for over a decade now. Although the analyzer itself proved to be an efficient tool, the format of the original database turned out to be problematic. The operations that the analyzer uses when analyzing the input word must be very simple so that processing could be very efficient. This requires that the data structures it uses contain redundant data (so that they do not have to be calculated on the fly during analysis).
The most important problem with the Humor analyzer was that MorphoLogic had no tools for creating and maintaining these redundant data structures, which were hard to read for humans, and to modify in a consistent way. This resulted in errors and inconsistencies in the descriptions, which were difficult to find and correct.

3.1. Creating a Morphological Description

So the first thing to do was to create an environment which facilitates the creation of the database. In the new environment, the linguist has to create a high level human readable description which contains no redundant information and the system transforms it in a consistent way to the redundant representations which the analyzer uses. The work of the linguist consists of the following tasks:
Identification of the relevant morpheme categories in the language to be described (parts of speech, affix categories).
Description of stem and suffix alternations: an operation must be described which produces each allomorph from the lexical form of the morpheme for each phonological allomorphy class. The morphs or phonological or phonotactic properties which condition the given alternation must be identified.
Identification of features: all features playing a role in the morphology of the language must be identified. These can be of various sorts: they can pertain to the category of the morpheme, to morphologically relevant properties of the shape of a given allomorph, to the idiosyncratic allomorphies triggered by the morpheme or to more than one of these at the same time.
Definition of selectional restrictions between adjacent morphs: selectional restrictions are described in terms of requirements that must be satisfied by the set of properties (features) of any morph adjacent to a morph. Each morph has two sets of properties: one can be seen by morphs adjacent to the left and the other by morphs adjacent to the right. Likewise, any morph can constrain its possible neighbors by defining a formula expressing its requirements on each of its two sides.
Identification of implicational relations between properties of allomorphs and morphemes: these implicational relations must be formulated as rules, which define how redundant properties and requirements of allomorphs can be inferred from their already known (lexically given or previously inferred) properties (including their shape). Rules may also define default properties. relatively simple spe-

[1] The output of our lemmatizer is what is usually expected of a morphological analyzer.

cial-purpose procedural language, which we devised for this task, can be used to define the rules and the patterns producing stem and affix allomorphs.

Creation of stem and affix morpheme lexicons: in contrast to the lexicon used by the morphological analyzer, the lexicons created by the linguist contain the descriptions of morphemes instead of allomorphs. Morphemes are defined by listing their lexical form, category and all unpredictable features and requirements. Irregular affixed forms and suppletive allomorphs should also be listed in the lexicon instead of using very restricted rules to produce them. We implemented a simple inheritance mechanism to facilitate the consistent treatment of complex lexical entries (primarily compounds). Such items inherit the properties of their final element by default.

Creation of a word grammar: restrictions on the internal morphological structure of words (including selectional restrictions between nonadjacent morphemes) are described by the word grammar. The development environment facilitates the creation of the word grammar automaton by providing a powerful macroing facility.

Creation of a suffix grammar (optional): an optional suffix grammar can be defined as a directed graph, and the development environment can produce segmented suffix sequences using this description and the suffix lexicon. Using such preprocessed segmented sequences enhances the performance of the analyzer.

As it can be seen from the description of the tasks above, we encourage the linguist to create a real analysis of the data (within the limits of the model that we provide) instead of just blindly describing each word as one which belongs e.g. to class X23b.

3.2. Conversion of the Morphological Database

Using a description that consists of the information described above, the development environment can produce a lexical representation which already explicitly contains all the allomorphs of each morpheme along with all the properties and requirements of each allomorph. This representation still contains the formulae expressing properties and selectional restrictions in a human-readable form and can thus be easily checked by a linguist. The example below shows a representation of the alternating noun stem *lov* and the plural + second person plural possessive + consecutive case suffix sequence *jasnydla* from the Komi description.

```
lemma:  'lov[N]'
 form:  'lov'
  mcat: 'S_N'
  rp:   'cat_N sfxable mcat_stem'
  rr:   '!V_ini'
 form:  'lol'
  mcat: 'S_N'
  rp:   'cat_N sfxable mcat_stem'
  rr:   'V_ini'

lemma:  'jas[Pl]nyd[PSP2]la[CNS]'
 form:  'jas+nyd+la'
  mcat: 'I_Pl+I_PSP2+I_CNS'
  rp:   'mcat_infl'
  lp:   'Pl'
  lr:   'cat_Nom sfxable'
```

The noun (`N`) stem (`S_`) *lov* has two forms (allomorphs): *lov* and *lol*. Their right-hand side properties (`rp`) are: `cat_N` (syntactic category is noun), `sfxable` (suffixes may be attached) and `mcat_stem` (morphological category is stem). The allomorph *lov* also requires (`rr`) that the following morph should not be vowel-initial, while *lol* requires it to be vowel-initial.

The representation of the inflectional (`I_`) suffix sequence *jasnydla* states that it is composed of the surface forms *jas*, *nyd* and *la*, the category labels of which are `Pl` (plural), `PSP2` (possessive second person plural) and `CNS` (consecutive case), respectively. The properties of this form is `mcat_infl` (morphological category is inflection) and `Pl` (the first member of the sequence is a plural suffix). Its left neighbor must be a morph of a nominal category to which suffixes can be attached.

This representation is then transformed to the format used by the analyzer using an encoding definition description, which defines how each of the features should be encoded for the analyzer. The development environment makes it possible to express that certain properties are in fact mutually exclusive possible values of the same feature (eg. `cat_N` and `cat_V`) by decomposing them to independent binary properties in the encoding definition.

4. The Komi Analyzer

In the subproject on Komi, which concentrates on the standard Komi-Zyryan dialect, we created a Komi morphological description using the development environment described in the previous section. As a result, a working morphological analyzer, a lemmatizer and a generator have been produced.

4.1. The Language

Komi (or Zyryan, Komi-Zyryan) is a Finno-Ugric language spoken in the northeastern part of Europe, West of the Ural Mountains. The number of speakers is about 300 000. Komi has a very closely related language, Komi-Permyak (or Permyak, about 150 000 speakers), which is often called a dialect, but with a standard of its own.

As a language spoken in Russia, Komi is an endangered language. Although it has an official status in the Komi Republic (Komi Respublika), this means hardly anything in practice. The education is in Russian, children attend only a few classes in their mother tongue. A hundred years ago, 93% of the inhabitants of the region were of Komi nationality. Thanks to the artificially generated immigration (industrialization, deportation) their proportion is under 25% today.

Komi is a relatively well documented language. The first texts are from the 14th century, and there is a great collection of dialect texts from the 19th and 20th centuries. There are linguistic descriptions of Komi from the 19th century, but hardly anything is described in any of the modern linguistic frameworks.

4.2. Creating a Komi Morphological Description

Since the annotated corpora we want to create are intended for linguists, we decided to use a quasi-phonological transcription of Komi based on Latin script instead of the Cyrillic orthography of the language. The non-phonemic nature of the Cyrillic orthography results in a number of linguistically irrelevant alternations we did not want to deal with in the first place. On the long run,

however, we plan to produce a Cyrillic version of the analyzer as well.

The first piece of description we created in the Komi subproject was a lexicon of suffix morphemes along with a suffix grammar, which describes possible nominal inflectional suffix sequences. One of the most complicated aspect of Komi morphology is the very intricate interaction between nominal case and possessive suffixes.

Another problem we were faced with was that neither of the linguistic descriptions we had access to describes in detail the distribution of certain morphemes or allomorphs. In some of these cases we managed to get some information by producing the forms in question (along with their intended meaning) with the generator and having a native speaker judge them. In other cases we will try to find out the relevant generalizations from the corpus.

Then we started to work on the stem lexicon along with the formal description of stem alternations triggered by an attached suffix. Fortunately, all of the stem alternations are triggered by a simple phonological feature of the following suffix: that it is vowel initial. The alternations themselves are also very simple (there is an *l~v* alternation class and a number of epenthetic classes).

On the other hand, it does not seem to be predictable from the (quotation) form of a stem whether it belongs to any of the alternation classes. This information must therefore be entered into the stem lexicon. The following is a list of all nominal and verbal alternation classes with an example for each from the stem lexicon.

```
töv[N];stemalt:LV;
lym[N];stemalt:Jep;
mös[N];stemalt:Kep;
un[N];stemalt:Mep;
göp[N];stemalt:Tep;
ov[V];stemalt:LV;
lok[V];stemalt:Tep;
jul[V];stemalt:Yep;
```

These are the actual entries representing these stems in the stem lexicon. The quotation form is followed by a label indicating its syntactic category and its unpredictable idiosyncratic properties (in this case the stem alternation class it belongs to). For regular stems only the lexical form and the category label has to be entered.

Irregular suffixed forms and suppletive or unusual allomorphs can be entered into the lexicon by listing them within the entry for the lemma to which they belong. The following example shows the entry representing a noun which has an irregular plural form.

```
pi[N];rr:!Pl;\
    ++!pi+jan[PL];rr:(Cx|Px);
```

The entry defines the noun *pi*, which requires that the morph following it should not be the regular plural suffix (which is *-jas*) and introduces the irregular plural form *pijan*, which in turn must be followed by either a case or a possessive suffix.

In Komi, personal pronouns are inflected for case and reflexive pronouns are inflected for case, number and person. Locative case suffixes can be attached to postpositions and adverbs. Certain parts of these paradigms are identical to that of regular nominal stems, but there are also idiosyncrasies (especially among the forms of reflexive pronouns there are very many idiosyncratic ones). We handled regular subparadigms by introducing lexical features and having the analyzer process the corresponding word forms like any regular suffixed word. Idiosyncratic forms, on the other hand, were listed the in the lexicon along with their analysis.

It turned out to be extremely difficult to acquire any lexical resources (either dictionaries or corpora) for Komi in an electronic form. We found practically nothing on the Internet. At present, we have a very limited amount of text available. We converted this corpus to the quasi-phonemic Latin transcription we use. The stem lexicon now contains all stems occurring in this corpus.

5. Conclusion

The tests we have performed on the corpus available to us with the morphological tools described above promise that they will be an effective means of producing the annotated corpora we intend to arrive at.[2] The morphological database for Komi could be created rapidly using the high-level description language of the development environment. At present, the Komi database contains the description of most of the morphological processes in the language. On the other hand, the size of the stem lexicon is quite limited due to our limited lexical resources.

One of the remaining tasks is to expand and refine the lexicon of the analyzer and to gather further corpora and to annotate them.

References

Beesley, Kenneth R. and Lauri Karttunen. (2003). Finite State Morphology. Stanford, CA: CSLI Publications.

C. Hockett. (1954). Two models of grammatical description. Word 10 (2): 210-234.

J. Hoeksema and R. Janda. (1988). Implications of process-morphology for categorial grammar. In: R. Oehrle et al. (eds.), Categorial Grammars and Natural Language Structures. Dordrecht: Reidel.

P. Matthews. (1991). Morphology. Second edition. Cambridge, MA: Cambridge University Press.

Prószéky, Gábor and Balázs Kis. (1999). A Unification-based Approach to Morpho-syntactic Parsing of Agglutinative and Other (Highly) Inflectional Languages. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, (pp. 261–268). College Park, Maryland, USA.

[2] We must also add that beside its fortes, the toolset has its limits: we found that the formalism we successfully used to describe Komi (and a number of other languages) does not apply so smoothly to another small member of the Uralic language family, Nganasan, where a quite morphology-independent surface phonology plays an important role in shaping the form of words. The very productive (and quite intricate) gradation processes in Nganasan are governed by a set of constraints on surface syllable structure (both the presence of a coda and an onset and whether the syllable is odd or even play a role). Gradation in Nganasan is difficult to formalize as a set of allomorph adjacency restrictions because phonemes at the opposite edges of syllables may belong to non-adjacent morphemes. We thus turned to the Xerox finite-state toolset (Beesley and Karttunen, 2003), which fortunately became easily accessible for non-commercial purposes last summer, to create an analyzer for Nganasan.