Porting morphological analysis and disambiguation to new languages

Trond Trosterud
University of Tromsø

Department of Linguistics
N-9037 Universitety of Tromsø, Norway
trond.trosterud@hum.uit.no
http://giellatekno.uit.no/

Abstract

The paper presents a parser and disambiguator for North and Lule Sámi, and effort aimed at porting the work on Sámi to other Uralic languages.

1 Introduction

This poster presents morphological parsers and disambiguators for North and Lule Sámi, Uralic languages spoken in the Northern parts of Norway, Sweden and Finland (the project's home page is *http://giellatekno.uit.no/*. The parsers use Xerox tools (*www.fsmbook.com*) for morphological analysis, and constraint grammar for disambiguation (*sourceforge.net/projects/vislcg/*). It also report from the experiences with porting the system for Sámi to other Uralic languages.

2 The Tromsø disambiguation project

2.1 Morphological analysis

The morphological analyses of the project are based upon a two-level analysis with finite state automata, cf. [3]. We use Xerox software, (*lexc* for lexical analysis and segmental morphology, *twolc* for morphophonological processes, *perl* for preprocessing and *xfst* for case conversion and integrating the parts into a whole (cf. e.g. [1] and *http://www.fsmbook.com*). The lexical analysis and the segmental morphology operate on two levels, one surface level for roots and affixes, and one underlying level for lexemes and grammatical properties. The surface level again becomes the underlying level for the morphophonological rule set, taking a root- and suffix string as input, enriched with morphophonological information, and transforms it to the wordform we know from the written language (cf. [2]).

2.2 Suffixes

The lexicon contains all the roots of the language. The roots are classified according to part of speech (POS) and stem class, directing words taking the same suffixes and undergoing the same morphophonological processes to the same continuation lexica.

2.3 Morphophonology

Morphophonological processes are taken care of in a different component. The rules invoked in generating the forms shown in the float diagram are shown below. In the figure the lexicon MUORRA contains words undergoing consonant gradation, and NOADE contains words that do not. The case forms that show weak grade are kept in a separate lexicon, and the MUORRA nouns get a weak grade mark $Q1$ before they enter this lexicon. This means that the root consonants (here: sst and jvv undergo gradation, change to st, jv, their accusative forms are $bastiv$, $biejvev$, and not $*basstiv$, $*biejvvev$, although we have $girkkov$, $barggev$. They undergo the following rules (using the $twolc$ formalism):

3 Disambiguating Sámi

As an example, let us take the sentence *Mii eat leat dan muitalan* 'We haven't told it', with the verbs *leat* 'to be' and *muitalit* 'to tell'. The sentence is given the following analysis, prior to disambiguation:

```
``<Mii>''
        ``mun'' Pron Pers Pl1 Nom
        ``mii'' Pron Interr Sg Nom
``<eat>''
        ``ii'' V Neg Ind Pl1
``<leat>''
        ``leat'' V Ind Prs Pl1
        ``leat'' V Ind Prs Pl3
        ``leat'' V Ind Prs Sg2
        ``leat'' V Inf
        ``leat'' V Ind Prs ConNeg
``<dan>''
        ``dat'' Pron Dem Sg Acc
        ``dat'' Pron Dem Sg Gen
``<muitalan>''
        ``muitalit'' V PrfPrc
        ``muitalit'' V Act
        ``muitalit'' V Ind Prs Sg1
``<.>''
```

The only unambiguous word is *eat*, first person plural of the negation verb. In reality the sentence does not have 60 readings (2 x 5 x 2 x 3), but one:

```
``<Mii>''
        ``mun'' Pron Pers Pl1 Nom
``<eat>''
        ``ii'' V Neg Ind Pl1
``<leat>''
        ``leat'' V Ind Prs ConNeg
``<dan>''
        ``dat'' Pron Dem Sg Acc
``<muitalan>''
        ``muitalit'' V PrfPrc
``<.>''
```

Here are the rules that were used to arrive at the correct reading (the rules are given according to constraint grammar conventions, the numbers identify positions in the clause, 0 is the wordform to be disambiguated, 1 is the first word to the right and -2 the word two positions to the left, *-2 to a word two or more positions to the left (for an introduction to the rule formalism, see [6]).

```
SELECT Pers IF (0 ("mii"))(*1 V-Pl1 BARRIER NON-ADV);
SELECT ConNeg IF (*-1 Neg BARRIER VFIN);
SELECT Acc IF (*-1 LEAT-FIN-NON-IMP BARRIER NON-PRE-N)(1 PrfPrc);
SELECT PrfPrc IF (*-1 Neg BARRIER CONFRA);
```

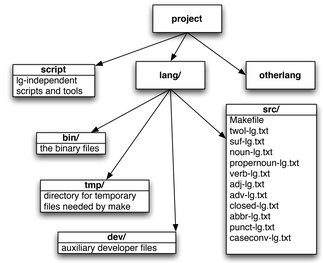
The pronoun *mii* may be a personal or interrogative. The rule states that if there is a PL1 verb to the left, with no other words than adverbs between the two, then the personal pronoun reading is selected. In order to get the correct reading for copula, the ConNeg form (the form connected to negative verbs) is chosen if a negation verb may be found somewhere to the left, before we find any other finite verb. The rule for perfect participles is similar, but here the barrier is a set of words cancelling negation, like the word *muhto* 'but'. This set has been listed earlier, and is labelled CONTRA. The rule for accusative demands a finite copula to the left, and with nothing but NP-internal pre-modifiers intervening, and a perfect participle to the right. In order to disambiguate running text, approximately 1500 to 2500 such rules are needed.

4 Extending the work on Sámi to other languages

4.1 Sharing infrastructure

The North Sámi transducer has been ported to several Uralic languages. In this process, much of the content was reused: Partly or fully language-independent files like lists of punctuation marks and letters for case conversion. Linguistic resources like person names and place names may be reused (with different continuation lexica for different languages), the same goes for productive loan-words. Thus, certain classes of Norwegian verbs and nouns can potentially be borrowed into the different Sámi languages, with slightly different adjustment strategies. The same goes for Russian loan-words in Udmurt, Mari, etc. A way of re-using such resources is to create a common pool of potential loan-words, with language-specific adjustment strategies. This strategy also makes it easy to remove these words from certain applications, e.g. puristic spell-checkers.

A major advantage was found in replicating the basic file structure for each new language. The initial development phase was shortened, and work on multilingual projects was easier with a uniform file structure.



4.2 Localisation

North Sámi has 7 letters outside ascii (for example *áčšž*). The origin of the project dates back to 1994, so it is localised in Latin 1, using digraphs for the 6 non-Latin1-letters: *á, c1, s1, etc.*. The current versions of the Xerox tools support UTF-8. We have made UTF-8-based parser prototypes (for Komi and Hindi). The source code was written in TextEdit on Mac OS X. TextEdit is clearly not an optional text editor. Unicode support on Linux is still not the default option (cf. *http://www.cl.cam.ac.uk/ mgk25/unicode.html*), but once the technical problems are solved, there are several benefits of writing the source code in UTF-8: dictionary files may be imported directly, they are easier to proofread, text may be analysed without conversion, and the morphophonological (actually morphographemical) rules may generalise directly over the actual letter sequences.

The following is an extract from the Komi analysis.

```
LEXICON NOUNSUF
+N+Sg: NOUNSUF-2 ;
+N+Pl:%>яс NOUNSUF-2 ;

LEXICON NOUNSUF-2
 PREPX ;
 PRECX ;
 ACCLEX ;
 LOCLEX ;
 CARLEX ;
 ELALEX ;

LEXICON PREPX
PREPXSG ;
PREPXPL ;

LEXICON PREPXSG
POSTCX ;
+PxSg1:%>ой POSTCX;
+PxSg2:%>ыд POSTCX;
+PxSg3:%>ыз POSTCX;

LEXICON PREPXPL
POSTCX ;
+PxPl1:%>ным POSTCX;
+PxPl2:%>ныд POSTCX;
+PxPl3:%>ныз POSTCX;

LEXICON POSTCX
+Nom: #;
+Gen:%>йӧн #;
+Abl:%>лысь #;
+Dat:%>лы #;
+Com:%>код К ;
+Cns:%>ла К ;
LEXICON ACCLEX
+Acc:%>ос #;

LEXICON LOCLEX
+Ill: LOCPX ;
+Ill:%>ӧ #;
+Ine: LOCPX ;
+Ine:%>ын #;
+Ins:%>н LOCPX ;
+Ins:%>ӧн #;

LEXICON LOCPX
+PxSg1:>ам К ;
+PxSg2:>ад К ;
+PxSg3:>ас К ;
 :%>а POSTPX ;

LEXICON ELALEX
+Ela:%>ысь #;
+Ela:%>сь POSTPX ;

LEXICON CARLEX
+Car:%>тӧг #; ! caritive
+Car+PxSg1:%>йӧтӧг К ;
+Car+PxSg2:%>тӧгыд К ;
+Car+PxSg3:%>тӧгыс К ;
+Car:%>тӧг POSTPXPL ;

LEXICON PRECX
+Apr:%>лань POSTPX;
+Egr:%>сянь POSTPX;
+Tra:%>ті POSTPX;
+Tra:%>од POSTPX;
+Ter:%>одз POSTPX;

LEXICON POSTPX
POSTPXSG ;
POSTPXPL ;

LEXICON POSTPXSG
+PxSg1:%>ым К ;
+PxSg2:%>ыд К ;
+PxSg3:%>ыз К ;

LEXICON POSTPXPL
+PxPl1:%>ным К ;
+PxPl2:%>ныд К ;
+PxPl3:%>ныз К ;
```

This is an extract from the Hindi analysis.

```
Multichar_Symbols

+N
+Sg +Pl
+Nom +Obl +Voc

LEXICON Root
Noun ;

LEXICON Noun
लड़का:लड़क FIRSTLONG ; ! larka, 'boy'
कुआँ:कुआ FIRSTSHORT ; ! kuā, 'well'
घर SECOND ; ! ghar, 'house'
लड़की THIRD ; ! larkii, 'girl'

LEXICON FIRSTLONG
+N+Msc+Sg+Nom:ा # ;
+N+Msc+Sg+Obl:े # ;
+N+Msc+Sg+Voc:े # ;
+N+Msc+Pl+Nom:े # ;
+N+Msc+Pl+Obl:ों # ;
+N+Msc+Pl+Voc:ो # ;

LEXICON FIRSTSHORT
+N+Msc+Sg+Nom:ाँ # ;
+N+Msc+Sg+Obl:ों # ;
+N+Msc+Sg+Voc:ों # ;
+N+Msc+Pl+Nom:ों # ;
+N+Msc+Pl+Obl:ओं # ;
+N+Msc+Pl+Voc:ओ # ;

LEXICON SECOND
+N+Msc+Sg+Nom: # ;
+N+Msc+Sg+Obl: # ;
+N+Msc+Sg+Voc: # ;
+N+Msc+Pl+Nom:ें # ;
+N+Msc+Pl+Obl:ों # ;
+N+Msc+Pl+Voc:ो # ;

LEXICON THIRD
+N+Fem+Sg+Nom: # ;
+N+Fem+Sg+Obl: # ;
+N+Fem+Sg+Voc: # ;
+N+Fem+Pl+Nom:ाँ # ;
+N+Fem+Pl+Obl:ों # ;
+N+Fem+Pl+Voc:ों # ;
```

4.3 Working on similar languages in parallel

Porting grammatical analysis to new languages is especially useful for language continua, or groups of closely related languages, where the linguistic analyses may be reused. Examples include Turkic, Bantu, Dravidian, Indo-Aryan, Slavic, Romance and Scandinavian.

4.4 Grammatical approaches in language technology

Grammar-based disambiguation has been known to provide good results, compared to stochastically-based approaches [5].

Looking at minority languages, the arguments in favour of grammar-based approaches are even stronger. In the cases of the Sámi languages or the Uralic languages of Russia, there is not a choice between using the multimillion electronically available corpus or not. There is no such corpus. Rather, what is available is a grammar, and in most cases a reasonably good dictionary. With these two tools (especially if the dictionary is electronically available, it is possible to build good transducers and disambiguators within a couple of years, or, after a while, within even shorter time. For inflectional languages with hundreds of inflected forms for each lexeme (and sometimes more), transducers based on stem classes and inflectional paradigms are the only way of ensuring good coverage of the language.

Another option than the manual writing of transducers is to apply to a combined version of human elicitation and machine learning, as argued by [4]. This approach should be more suited to families of very similar languages, like the Turkic or Bantu languages. Whether these semiautomatic transducers are as easy to update as hand-made ones, or whether they will look more like a "black box", remains to see.

Most minority languages do not have many and rich enough speakers to attract commercial language technology projects. Linguists still write reference grammars for these languages. For grammar-based language technology, it makes perfectly sense to be integrated in this descriptive work. Making a morphological parser is the best way of checking the coverage of any language description. Practical applications like spell checkers should then come as a side effect of this type of basic descriptive work.

5 Summary

The present poster has given an overview of work with morphological transducers and disambiguators for some related Uralic languages. The work conducted so far shows that the building of transducers and disambiguators will benefit from sharing code written in an as language-independent way as possible.

References

[1] Kenneth R. Beesley and Lauri Karttunen. *Finite State Morphology*. Studies in Computational Linguistics. CSLI Publications, Stanford, California, 2003.

[2] Lauri Karttunen. Constructing lexical transducers. In *15th International Conference on Computational Linguistics (COLING-94)*, pages 406–411, Kyoto, Japan, 1994.

[3] Kimmo Koskenniemi. *Two-level Morphology: A General Computational Model for Word-form Production and Generation*. Publications of the Department of General Linguistics, University of Helsinki. University of Helsinki, Helsinki, 1983.

[4] Kemal Oflazer, Sergei Nirenburg, and Marjorie McShane. Bootstrapping morphological analyzers by combining human elicitation and machine learning. *Computer Engineering Technical Report*, BU-CE-0003, 2000.

[5] Christer Samuelsson and Atro Voutilainen. Comparing a linguistic and a stochastic tagger. In *35th Annual Meeting of the Association for Computational Linguistics*, 1997.

[6] Pasi Tapanainen. *The Constraint Grammar Parser CG-2*, volume 27 of *Publications of the Department of General Linguistics, University of Helsinki*. University of Helsinki, Helsinki, 1996.