Twol at work

Sjur Moshagen
The Saami Parliament
Norway

Pekka Sammallahti
Giellagas Institute
University of Oulu

Trond Trosterud
Faculty of the humanities
University of Tromsø

1 Introduction

In this article, we will show two-level morphology at work. In sections 2 and 3, we will lay out the foundation for the work, by presenting the philosophy behind the Northern Saami two-level parser. In sections 4.1, 4.2 and 4.3 we then have a look at different applications, pedagogical programs, spell checking and terminology management.

2 Two-level morphology

Languages with ample morphophonological variation pose a problem for automatic analysis. A recapitulation of historical changes (or phonological rules) in two-level rules may be an elegant solution from the point of view of an overall grasp of the language in question but one soon runs into difficulties in dealing with products of analogical levellings and other exceptions, especially when text words consist of several morphemes each interacting with the other phonologically. One such language is Saami where word stems interacting with affixes can have over 20 phonological variants and derivational and inflectional morphemes interacting with word stems or each other more than five.

After unsuccesfully trying different morphophonological rule approaches to the variation stemming from morpheme interaction an **indexed concatenation model** was devised. This model provides **phonological/graphemic morpheme variants** with indexes (or abstract features) according to their continuation categories, the sets of suffixes it predAs. In practice a phonological/graphemic variant of a word stem occurring before a certain set of suffixes receives an index different from that of another variant occurring before a different set of suffixes.

Since the distribution of stem variants in relation to suffix sets vary from one stem type to another, a single phonological/graphemic variant of a word stem may belong to two or more morphophonological variants if a different stem has two or more phonological/graphemic variants before the same set of suffixes. Accordingly the suffixes are indexed according to their relations with morphophonological word stem variants.

The interaction with the word stems*giehta* 'hand; arm' and*njunni* 'nose' with the nominative plural suffix *t* and the second person singular accusative possessive suffix *t* may serve as an example. Both suffixes call for **weak grade** in the stem consonant center (the consonants between a stressed and a stressless vowel): (*njunni* →) *njuni-t* 'noses' and (*giehta* →) *gieđa-t* 'hands/arms'. However, the two suffixes call for different stem vowel alternants in i-stems but not in a-stems: *njuni-t* 'noses' ≠ *njuná-t* 'your nose' but *gieđa-t* 'hands/arms' = *gieđa-t* 'your hand/arm'. The stems *njuni-* and *njuná-* receive different indexes or abstract features (such as *njuni-N1* and *njuná-N2*) because they call for different sets of suffixes but so do the corresponding instances of the stem *gieđa-* (*gieđa-N1* and *gieđa-N2*) because the suffixes it precedes belong to two sets.

Since nouns have partly the same mophophonological variants as verbs, the morphophonological variants receive three kinds of indexes: *N* for the morphophonological variants of nouns, *V* for the morphophonological variants of verbs, and *X* for the morphophonological variants shared by nouns and verbs. To illustrate the use of indexes, a selection of noun and verb inflectional forms from three different parisyllabic stems are given. The examples represent types which show maximal morphophonological variation. Those with monophthongs in the first syllable (such as *ruhta* 'money', *lohti* 'wedge', *rohtu* 'grove'; *sihtat* 'want', *bihtit* 'to be strong enough for something', *vihkut* 'to suspect') or with other kinds of consonant centers (such as *gálgat* 'to undo', *máhttit* 'to know how', *riggut* 'to become rich') have a smaller number of morphophonological variants because some word stem accommodations such as diphthong simplification or extra strong grade do not manifest themselves in them. Cf. tables 1 and 2.

Table 1: Noun Stems: *giehta* 'hand; arm', *goahti* 'hut; Saami tent', *niehku* 'dream'

NomSg	giehta-Ø	goahti-Ø	niehku-Ø
IllSg	gihti-i	goahtá-i	nihku-i
LocSg	gieđa-s	goađi-s	niegu-s
ComSg	gieđa-in	gođi-in	niegu-in
Ess	giehta-n	goahti-n	niehku-n
NomPl	gieđa-t	goađi-t	niegu-t
NomSg+Sg2Px	giehta-t	goahtá-t	nihko-t
GenSg+Sg2Px	gieđa-t	goađá-t	nigo-t

Table 2: Verb Stems: *viehkat* 'to run', *boahtit* 'to come', *biehkut* 'to complain'

Inf	viehka-t	boahti-t	biehku-t
IndPrsSg1	viega-n	boađá-n	biegu-n
IndPrsSg3	viehká-Ø	boahtá-Ø	biehku-Ø
IndPrt Du3	viega-iga	bođi-iga	biegu-iga
IndPrtPl3	vihke-t	bohte-t	bihko-t
PotPrsSg1	viega-žan	bođe-žan	bigo-žan
CondPrsSg1	viega-šin	boađá-šin	bigo-šin
ImprtPrsSg2	viega-Ø	boađe-Ø	biego-Ø
ImprtPrsSg3	vihk-os	boht-os	bihk-os
ImprtPrsDu2	viehkki-Ø	boahtti-Ø	biehkku-Ø
ImprtPrsPl1	vihk-ot	boht-ot	bihko-t
VrbGen	viega-Ø	boađi-Ø	biegu-Ø
PrsPtcComSg	vihkki-in	bohtti-in	biehkku-in
PrfPrc	viehka-n	boahtá-n	bihko-n
PassiveStem	vihkk-o-	bohtt-o-	bihkk-o-

The inflectional morphemes are accompanied with a number of word stem accommodations, the ones relevant with regard to the examples in the paradigms are shown in table 3:

We can see that the distribution of these accommodations, each of which corresponds to a rule in the two-level analysis program, is not the same in the paradigms representing different stem types. Grade Alternation, Consonant Center Strengthening, Second Syllable Allegro Shortening and Second Syllable Vowel Loss Before Suffixal Vowel occur in the same paradigmatic forms for all types of parisyllabic stems but Diphthong Simplification and the different Second Syllable Vowel Accommodations do not. On closer inspection, however, it becomes clear that DS and SSVAs are conditioned by the phonological properties of the stems and that their distribution differences depend on their applicability to different kinds of stems.

The stem variants can now be grouped according to the combinations of accommodations in the suffixes the continuation categories call for; if Diphthong Simplification is restricted to a stem type, it will be indicated with the stem vowel in braces. Out of the 15 groups of stem variants requiring different sets of suffixes in table 4, one is specific to nouns (nr. 4), eight are specific to

Table 3: Word Stem Accomodatoins

(a)	Weak Grade (WG): hk → g, ht → đ
(b)	Extra Strong Grade (ESG): hk → hkk, ht → htt
(c)	Diphtong Simplification (DS): ie → i, oa → o, uo → o
(d)	Second Syllable Vowel Accommodation I (SSVAI): i → á, u → o
(e)	Second Syllable Vowel Accommodation II (SSVAII): i → e, u → o
(f)	Second Syllable Vowel Accommodation III (SSVAIII): a → i
(g)	Second Syllable Vowel Accommodation IV (SSVAIV): a → á, i → á
(h)	Second Syllable Vowel Accommodation V (SSVAV): a → i, i → á
(i)	Second Syllable Allegro Shortening (SSAS): i → e, u → o (+ á → a)
(j)	Second Syllable Vowel Loss Before Suffixal Vowel (SSVL): -a → -Ø, -i → -Ø, -u → -Ø

verbs (nrs 3, 5, 6, 9, 10, 12-15) and 6 are shared by nouns and verbs (nrs 1, 2, 6, 7, 8, 11). This means that in the presented examples, noun stems have seven variants which require different continuation categories and verb stems have 14, as in table 4:

The continuation categories presented here are considerably simplified for the purposes of this paper; those in the actual program take various kinds of redundances as well as sequences of continuation categories into account and are far more complex. Furthermore, most of the continuation categories contain a number of suffixes in addition to those in the examples.

With the indexed concatenation model outlined here, it is possible to deal with all kinds of complex morphophonologies in a straightforward manner. It is also relatively easy to add new morpho¬phon¬ological accommodations into the analyser.

3 Disambiguation

Disambiguation may be done in several ways. One is Finite-state intersection grammar, as suggested by [Kos97]. In our Saami project, we have chosen a different path, and use constraint grammar, as presented by [Tap96], here in Eckhard Bick's open-source version *vislcg* (cf. *sourceforge.net/projects/vislcg/*. This component is being written by Trond Trosterud and Marit Julien. Although still under development, it is already good enough to match the level of statistically-based POS taggers. At its present stage, it contains approximately 1300 disambiguation rules.

4 Twol in use

A grammatical analysator can be used for many purposes. We will here have a look at some areas where the Saami analysator has been put to use.

4.1 Pedagogical programs

The Saami analysator has been used to analyse sentences for interactive pedagogical syntax learning in the so-called visl project (Visual Syntax Learning) at Syddansk Universitet. The format in itself is not dependent upon having a grammatical analysator, but the analysator makes it possible to add sentences automatically. The process behind the analysis in Figure 3 consists of three parts:

1. The morphological analyser gives all possible analyses
2. A morphological disambiguator removes the incorrect ones, and add syntactic functions
3. A phrase structure grammar gives the linear representation a hierarchical structure

Table 4: Stem variants

(1) **X1**: *No Accommodation*			
N+Sg+Nom: Ø	giehta-Ø	goahti-Ø	niehku-Ø
N+Ess: n	giehta-n	goahti-n	niehku-n
V+Inf: t	viehka-t	boahti-t	biehku-t
(2) **X2**: *SSVAI*			
N+Sg+NomSg+Sg2Px: t	giehta-t	goahtá-t	nihko-t
V+PrfPrc: n	viehka-n	boahtá-n	bihko-n
(3) **V1**: *SSVAIV*			
V+Ind+Prs+Sg3: Ø	viehká-Ø	boahtá-Ø	biehku-Ø
(4) **N1**: *DS(a,u) + SSVAV*			
N+Sg+Ill: i	gihti-i	goahtá-i	nihku-i
(5) **V2**: *DS + SSVAII*			
V+Ind+Prt+Pl3: t	vihke-t	bohte-t	bihko-t
(6) **V3**: *DS + SSVL*			
V+Imprt+Prs+Sg3: os	vihk-os	boht-os	bihk-os
V+Imprt+Prs+Pl1: ot	vihk-ot	boht-ot	bihk-ot
(7) **X3**: *WG*			
N+Sg+Loc: s	gieđa-s	goađi-s	niegu-s
N+Pl+Nom: t	gieđa-t	goađi-t	niegu-t
V+VrbGen: Ø	viega-Ø	boađi-Ø	niegu-Ø
(8) **X4**: *WG + SSVAI*			
N+Sg+Gen+Sg2Px: t	gieđa-t	goađá-t	nigo-t
V+Cond+Prs+Sg1: šin	viega-šin	boađá-šin	bigo-šin
(9) **V4**: *WG + SSAS*			
V+Imprt+Prs+Sg2: Ø	viega-Ø	boađe-Ø	biego-
(10) **V5**: *WG + SSVAIV*			
V+Ind+Prs+Sg1: n	viega-n	boađá-n	biegu-n
(11) **X5**: *WG + DS(i)*			
N+Sg+Com: in	gieđa-in	gođi-in	niegu-in
V+Ind+Prt+Du3: iga	viega-iga	bođi-iga	biegu-iga
(12) **V6**: *WG + SSVAII*			
V+Pot+Prs+Sg1: žan	viega-žan	bođe-žan	bigo-žan
(13) **V7**: *ESG + SSVAIII*			
V+Imprt+Prs+Du2: Ø	viehkki-Ø	boahtti-Ø	biehkku-Ø
(14) **V8**: *ESG + DS(a,i) + SSVAIII*			
N+Sg+Com: in	vihkki-in	bohtti-in	biehkku-in
(15) **V9**: *ESG + DS + SSVL*			
V+Passive: o	vihkk-o-	bohtt-o-	bihkk-o-

At present, beta versions of the first two components are in place. The third component is still missing. For a sentence like *Áhčči lea oastán munnje divrras sabehiid* 'Father has bought me a an expensive pair of skis', the morphological analyser gives the representation in Figure 1.

Figure 1: Morphological analysis

```
"<Áhčči>"
         "áhčči" N Sg Nom
"<lea>"
         "leat" V Ind Prs Sg3
"<oastán>"
         "oastit" V PrfPrc
         "oastit" V* N Actor Sg Nom PxSg1
         "oastit" V* N Actor Sg Gen PxSg1
         "oastit" V* N Actor Sg Acc PxSg1
         "oasti" N Sg Nom PxSg1
         "oasti" N Sg Gen PxSg1
         "oasti" N Sg Acc PxSg1
"<munnje>"
         "mun" Pron Pers Sg1 Ill
"<divrras>"
         "divrras" A Attr
         "divrras" A Sg Nom
"<sabehiid>"
         "sabet" N Pl Gen
         "sabet" N Pl Acc
"<.>"
         "." CLB
```

After disambiguation and adding of syntactic functions, the same sentence can be seen in Figure 2.

The underlying representation of the pedagogical program takes the disambiguated analysis as input, and makes a tree structure, as seen in Figure 3. At present, this process is only partly automatised, the bracketing of constituents (denoted with '=') must be done manually, making such a component is the next task ahead.

A pilot set of 200 Saami sentences will shortly be included in the *http://visl.sdu.dk/visl/*. As soon as they become part of that pedagogical platform, the sentences will be reused in a large range of pedagogical programs, ranging from interactive syntax analysis via word-class shooting games to text analysis.

Later, when the analysator becomes better, it will also be possible to offer an open system, where the computer analyses user input and offers an interface for the user to analyse for himself.

4.2 TWOL as generator 1: paradigm generation for a terminological database

The bidirectional nature of the two-level model makes it ideal not only for analysis but also for generation of word forms. An example of this is seen in Figure 4, using the current North Saami transducer.

This feature of the two-level model will be put into use in a terminological database developed by the Saami parliament (at *http://www.risten.no/*) to generate complete paradigms at runtime of any entry in the database. The paradigm generation will first be implemented for North Saami, and later for Julev (Lule) Saami and other Saami languages.

Figure 2: Disambiguated version

```
"<Áhčči>"
        "áhčči" N Sg Nom @SUBJ
"<lea>"
        "leat" V Ind Prs Sg3 @+FAUXV
"<oastán>"
        "oastit" V PrfPrc @-FMAINV
"<munnje>"
        "mun" Pron Pers Sg1 Ill @ADVL
"<divrras>"
         "divrras" A Attr @AN>
"<sabehiid>"
        "sabet" N Pl Acc @OBJ
"<.>"
```

Figure 3: Underlying representation in the pedagogical program

```
S:n('áhčči',sg,nom) Áhčči
P:g
=D:v('leat',ind,pr,3sg) lea
=H:v('oastit',pcp2) oastán
A:pron('mun',<pers>,1sg,ill) munnje
Od:g
=D:adj('divrras',attr)  divrras
=H:n('sabet',pl,acc) sabehiid
```

Figure 4: Analysis and generation of the same word form using the same two-level model in oposite directions

```
xfst[1]: apply up
apply up> máná
mánná+N+Sg+Acc
mánná+N+Sg+Gen

xfst[1]: apply down
apply down> mánná+N+Sg+Acc
máná
```

Unless special attention is paid to homonym entry words with different inflections, the simple application of a two-level model in the oposite direction will overgenerate, leading to wrongly generated word forms. One way of dealing with the overgeneration would be to add a unique string to homonyms as part of their lexical entry, as shown in Figure 5. This would ensure round-trip consistency without over-generation (one will always be able to generate exactly what was analyzed), but would complicate generation of such homonyms if no analysis is available: without knowing that it is a homonym, how would one know that the word requires a special tag to be generated?

```
beassi:beassi_1 GOAHTI ;
beas0s0i:beas'si_2 GOAHTI ;
ára:ára_1 GOAHTI ;
á0ra0:árran2_2 SEAMU ;
```

Figure 5: Homonym lexical entries with an additional differentiator to differentiate them in analysis and generation

With appropriate rules for dealing with homonym indices like *_1* and *_2*, the analysis output (and hence the generator input) would include the necessary info to generate complete and accurate paradigms without overgeneration.

4.3 Spell checker

Using two-level technologies for making spellers was early on a pretty obvious application of it (see **??** for a summary of the different initiatives in Finland), and one way of implementing orthographic correction is briefly described in [BK03]. Commercial implementations have been available since 1986 from Lingsoft Oy[1] for Finnish, then Swedish, later also several other languages. For languages with rich inflectional and/or derivational morphology and free compounding, like Saami languages as well as many others, using a two-level or similar approach is in practice the only possibility.

Two-level technology is not only good for spellers. For languages with free compounding, using two-level technologies can also improve hyphenation, cf. [Kar85]. A commercial implementation is available from Lingsoft Oy, and described at *http://www.lingsoft.fi/doc/d-finhyp9.html.*

Since October 2004 the Norwegian Saami Parliament has been running a project to create proofing tools for North and Julev (Lule) Saami. The project is based on the work by Pekka Sammallahti and the projects at the University of Tromsø, described earlier in this article.

[1]See http://www.lingsoft.fi/

4.3.1 Handling descriptive and normative models at the same time

The proofing tools project is using the same lexicons as other Saami language technology projects at the University of Tromsø. While the university projects by nature are descriptive and wants to be able to analyse both standard orthography and common substandard variants, the proofing tools' goal is to help authors make written text conform to orthographic standards. The language of the proofing tools is thus a subset of the language of the other projects, and to create a two-level model for proofing tools, we need to extract this subset language.

A very simple, but often sufficient, way of doing this, is to add a comment to unofficial variants, and remove these variants using text processing methods in a preprocessing stage before compiling the lexicon, typically using 'grep'. This is how it is implemented in the present version of the North Saami TWOL, cf Figure 6, where the comment *!SUB* marks substandard variants.

```
leans#mánOni:leans#mán'ni VIVVA ;
lens#mánOni:lens#mán'ni VIVVA ; !SUB
kapihtal GAHPIR ; !SUB
sektor GAHPIR ; !SUB
```

Figure 6: Example North Saami entries with comments for substandard forms

Another method would be to use the network operations available in current language technology tools such as the Xerox Finite State tools described in [BK03]. Using network subtraction it would be possible to remove from the full (descriptive) language model the language containing all and only a specified feature, e.g. +Sub. This would also imply that substandard variants that are morphophonological rather than lexical in their nature could easily be removed, which is not necessarily possible with the preprocessing approach discussed above.

Extending both the approaches briefly discussed above would also make it possible to cover dialectal variation. Comparative forms of North Saami adjectives have one standard variant used in western dialects, and another standard form used in eastern dialects. Each of these dialect groups has in addition another variant ending in -u, which is not part of the orthography, but used orally and thus sometimes showing up in print. All four variants are listed in Figure 7, in their *lexc* representation.

```
LEXICON EABBO/EAMOS_CONT
+Comp+W:eabbo EABBU ;  ! Parallel form Standard. West
+Comp+E:ab'bo EABBU ;  ! Parallel form Standard. East
+Comp+W+Sub:eabbu EABBU ;  ! Parallel form Not standard. West
+Comp+E+Sub:ab'bu EABBU ;  ! Parallel form Not standard. East
```

Figure 7: Dialectal variation of Comparative, both standard and substandard variants

We have introduced two more tags in the description above, *+W* and *+E*, to denote western and eastern dialects respectively. We have also used a lexc tag *+Sub* instead of a comment to identify substandard variants. Using any of the methods above we can extract any combination of these parameters, that is, including or excluding substandard variants, for eastern or western dialects. In a speller context, this can be used to e.g. create a more strict speller, by excluding forms not relevant for the user.

4.3.2 TWOL as generator 2: full form list generation for LT-weak spellers

The proofing tools project at the Saami parliament will create spellers and hyphenators for several applications on Windows, MacOS X and Linux, using several different speller engines. Some of these engines are dictated by the applications the project needs to make spellers for, for other applications it has been a goal to reduce the dependency on commercial software as much as possible. The most likely such speller engine is Aspell (*http://aspell.net/*), a derivative of iSpell, which employs a simple one-level automaton model for its engine. Aspell is open source, freely available, and runs on all operating systems the project plans to support. In addition, it plugs in to a multitude of applications on all platforms. Aspell's only weakness is its limited descriptive power, that is - the missing *two* level, which makes it quite hard to write spellers for languages with complex morphophonology.

Recreating the linguistic model in the limited format of Aspell is not an attractive choice: we don't want to maintain two sets of source files, let alone make sure they stay in sync. Developing a good linguistic model is in itself a huge task, and it is imperative that we can, with reasonable effort, reuse what has already been done.

Thanks to the bidirectional nature of the two-level model, it is relatively easy today to circumvent the limitations of spellers like Aspell to a large degree. As long as a transducer is non-circular, we can use it as a generator to not only produce paradigms as described in Section 4.2, but to print out the whole language of the transducer. The circular points in our lexicon are marked up, which makes it trivial to extract a subset of it that is non-recursive with a simple *grep* command. This subset still contains all the non-circular word formation, such as derivation, plus all the inflection.

Though quite substantial, such a generated full-form list will of course not have satisfying lexical coverage unless the underlying lexicon source files are themself «complete». A *complete* lexicon does not exist, but to make it as close to it as possible within the scope of the proofing project, we will add to the lexicon all entries found in available written material[2], including all compounds, such that the total lexical coverage should be quite good. It still remains to be seen whether the result will be satisfying. The following criteria will be essential in evaluating whether we are successful:

- size of resulting binary dictionary file
- speed of the speller
- precision and recall of the speller
- relevance of given suggestions

Using a full-form list also defeats another benefit of automatons and transducers, namely the space-efficient construction of regular morphology in continuation lexicons. iSpell[3] and lately Aspell[4] support what is called "affix compression" using "affix lexicons", which is a limited implementation of continuation lexicons[5]. The Divvun project will look into whether it would be possible to generate such affix lexicons automatically from the generated full-form list, or at least with minimal effort maintain such affix lexicons. This will be especially important if the resulting binary lexicons turn out to be very large without affix compression.

5 Summing up

In this article we have tried to describe issues in the development of the North Saami two-level model, and some of the applications for which it has been, or is going to be put to work. With the resent projects for pedagogical software and proofing tools, it is possible to secure a place in the modern, digital world for small languages like North Saami by using language technology rooted in [Kos83].

[2]The proofing project is building a corpus of Saami texts together with the disambiguator project; the corpus will hopefully contain substantial parts of the total body of Saami texts written in modern orthography

[3]http://fmg-www.cs.ucla.edu/fmg-members/geoff/ispell.html

[4]since version 0.60.

[5]see *http://aspell.net/man-html/Affix-Compression.html* for details

References

[BK03] Kenneth R. Beesley and Lauri Karttunen. *Finite State Morphology.* Studies in Computational Linguistics. CSLI Publications, Stanford, California, 2003.

[Kar85] Fred Karlsson. Automatic hyphenation of Finnish. In *Computational Morphosyntax, Report on Research 1981-84*, volume 13 of *Publications of the Department of General Linguistics, University of Helsinki*, 1985.

[Kos83] 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.

[Kos97] Kimmo Koskenniemi. Representations and finite-state components in natural language. In E. Roche and Y. Schabes, editors, *Finite-State Language Processing*, Language, speech and communication, pages 99–116, Massachusetts, 1997. MIT Press.

[Tap96] 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.