Next to nothing – a cheap South Saami disambiguator

Lene Antonsen
University of Tromsø
Norway
lene.antonsen
@uit.no

Trond Trosterud
University of Tromsø
Norway
trond.trosterud
@uit.no

Abstract

The goal of this article was to show that even a small constraint grammar may achieve results good enough to be used as a lemmatiser. The result shows that a rule set of 115 CG rules is efficient enough to give a lemmatisation accuracy (lemma + POS identification) of 1.056 for open POS.

1 Introduction

Lemmatising is important for a whole range of language technology applications. Morphology-rich languages get better word alignment, and both dictionary and terminology work need lemmatisation in order to be able to search for words in texts in reliable ways.

Constraint grammars are widely recognised for achieving deep syntactic analyses very close to the gold standard, but at the expense of requiring carefully crafted rule sets of several thousand rules (Karlsson et. al, 1995). The goal of this article is to investigate whether a small rule set may achieve a more restricted task, namely POS and lemma disambiguation.

1.1 Lemmatising

Deciding whether two word forms belong to the same lemma or not might be problematic. In order to do that, we first define the parts of speech of the language by morphosyntactic means. Which lexeme a given word form belongs to, will then follow from the overall POS structure. For us, lemmatising thus means finding the correct lexeme for each word form. Our research shows that even a small constraint grammar may achieve results good enough to be used as a lemmatiser.

Homonymy in the Uralic languages is more often than not confined to paradigm-internal homonymy. Two homonym word forms usually express different grammatical words of the same lexeme, and not homonym word forms of different lexemes. This means that even a partial disambiguation may be helpful for lemmatising, even though it fails in resolving all the grammatical ambiguities.

2 Relevant features of South Saami grammar

South Saami is, like the other Saami languages, a Uralic language. Typologically, it has a medium-size morphology, with 8 cases, 2 numbers for nouns, and 9 person-number values, 2 moods and 2 tenses for verbs, in addition to several infinite verbforms and a productive derivational morphology. The relatively agglutinative morphology is combined with a rather complex morphophonology (Sammallahti, 1998).

The most important morphophonological process is an Umlaut system consisting of 7 different vowel series and 6 different morphophonologically defined contexts. Other processes include diphthong simplification processes and suffix alternations depending upon the underlying foot structure.

Compared to the other Saami languages, South Saami has relatively little morphological ambiguity. On average, each reading receives 1.6 analyses, as compared to 2.6 analyses for North Saami.

3 Derivations

In the Saami languages there is much derivation, for all the open word classes. In our transducer lexicon (at http://giellatekno.uit.no), many of the derivations are lexicalized. Since more work has been done for North Saami than for the other languages, there are more lexicalisations in the North Saami lexicon than in the Lule and South Saami ones. In the output from the morphological analyser, there are dynamic analyses, in addition to the possibly lexicalized

one, as shown in Figure 1.

bájkálattjat (Lule Saami) ('locally')
`bájkke N Der1 Der/lasj A Der2 Der/at Adv`

báikkálaččat (North Saami) ('locally')
`báiki N Der1 Der/laš A Der2 Der/at Adv`
`báikkálaš A Der2 Der/at Adv`
`báikkálaččat Adv`

Figure 1: The morphological analysis of derived words may differ for the *sme* and *smj* analysers.

When extracting term pairs from parallel corpora, the challenge is to extract the lemmas in one language against the non-lexicalised lemma + derivation affix series in the other.

The algorithm is as follows:

1. Choose the lexicalized reading if there is one
2. If there is no lexicalised reading, choose the derived one with the fewest number of derivational affixes.

The Lule Saami word *bájkálattjat* means 'locally', and is derived from the adjective meaning 'local' which is derived from the noun meaning 'place'. In this case, word alignment between Lule Saami and North Saami gives the following alignment: *bájkke* 'place' = *báikkálaččat* 'locally'.

A better solution is to glue the derivation tags to the lemma, so the word alignment process will align *bájkke N Der1 Der/lasj A Der2 Der/at Adv* to *báikkálaččat Adv*. Figure **??**, Matt. 9.8., gives an example of lemmatised text with derivation tags.

original text:
`Muhto olbmot ballagohte go oidne dán, ja sii máidno Ipmila gii lea addán olbmuide dakkár fámu.`

lemmatised text:
`muhto olmmoš ballat+V+Tv+Der3+Der/goahti go oaidnit dát , ja son máidnut Ipmil gii leat addit olmmoš dakkár fápmu .`

'But people began to be afraid when they saw it, and they prised God which had given the people such a power.'

Figure 2: The lemmatised text contains derivation tags.

4 South Saami as part of a larger Saami analyser

The Saami languages have different morphological and morphophonological processes, and therefore separate morphological transducers are built for each language.

The output of the morphological analysers is then disambiguated in separate modules for each language. Due to different homonymy patterns of the languages, different rules apply. North Saami needs many rules in order to resolve the homonymy between accusative and genitive case. In Lule Saami, this type of homonymy is restricted to the personal pronouns, and in South Saami it does not exist at all.

The mapping of syntactic tags to conjunctions, subjunctions and finite and non-finite verbs is done at an early stage in the North and Lule Saami disambiguation files because these tags are used for sentence boundary detection, which is crucial for disambiguation of e.g. case forms.

However, the mapping of most of the syntactic tags is done in a common module shared by all three Saami languages, as shown in Table 1. The annotation is based on 49 syntactic tags.[1]. Due to the relatively free word order in Saami, a fairly large number of tags is needed.

The rules in the syntactic analyser refer to morphological tags and sets of lemmas (e.g. the `TIME` set contains lemmas that denote time adverbials), which are language specific. The disambiguator adds language tags (<sme>, <smj>, <sma> for North, Lule and South Saami, respectively) to all morphological analyses. When a lemma is identified as belonging to a certain language, language-specific rules and language-specific exceptions are triggered. E.g., in South Saami, the copula is often omitted in existential and habitive sentences, which means there is no finite verb in the sentence. In North Saami, a sentence without a finite verb is analysed as a fragment or an elliptic sentence, which is not appropriate for South Saami. Furthermore, the habitive function is expressed by different cases in North Saami (locative), Lule Saami (inessive) and South Saami (genitive). Nevertheless, `@HAB-tag` is assigned to all of them. The integration of the different disambiguation rule sets is presented in (Antonssen et al, 2010).

The mapping of dependency tags is done in a Constraint Grammar module common to all the Saami languages, and the rule set is compiled with the Visl CG3 compiler ((visl, 2008)) On the dependency level, syntactic tags for verbs are substituted

[1] `http://giellatekno.uit.no/doc/lang/sme/docu-sme-syntaxtags.html`

by other tags (according to clause-type) in order to make it easier to annotate dependency across clauses.[2]

4.1 Disambiguation

In order to test the disambiguator, we took a South Saami corpus of 142.500 words (55% Bible texts and 45% administrative texts). Our South Saami morphological analyser accepts substandard lemma and inflection forms. For frequent typographical errors we have a correction procedure. Despite of this, 12.395 words, or 8,7% of the corpus, were not recognized by our morphological analyser. The unknown words are partly due to the immature status of our morphological analyser, and partly due to the high degree of errors and non-normative forms in South Saami texts. The texts in the corpus were written at a time when there was no spellchecker available for South Saami. The written norm is new and unstable, and rules for writing loanwords are not established. The texts also contain upper cased headlines, which the analyser is not able to analyse, and there are proper nouns and some Norwegian words, which are not recognized by the analyser.

We made two versions of the corpus, one where the unknown words were removed, and one where all the sentences containing at least one unknown word were removed. Unknown words are uninteresting for disambiguation, with no analysis they trivially have no ambiguous cohorts either. Sentences with unknown words are also problematic, since the unknown words may influence upon the analysis of the remaining sentence. In order to look at disambiguation of sentences without unanalysed words, we removed all sentences with unknown words. In our test corpus, we have a missing rate of 8.7% words, and by removing all the affected sentences we lose 64% of the corpus. We are therefore also interested in looking at to what extent the unknown words influence the lemmatising.

The results may be seen in Table 2. The table shows the results for the whole corpus (left column), for the whole corpus analysed with a guesser (central column) and the subcorpus with fully analysed sentences (right column). For each corpus is shown the degree of homonymy (analyses per 1000 words) before and after disambiguation. We then show the result for lemma + PoS (lemmatising), first for all PoS, and then for a reduced PoS set, containing just 4 PoS's (N, V, A, other).

The results improve as we reduce the level of precision, from full analysis, PoS only, to a reduced 4-membered PoS set. For many lemmatisation purposes, distinguishing between different closed classes is not that interesting, and the relevant level of disambiguation is thus 1.056-1.058.

Surprisingly enough, the results for disambiguating the whole corpus is slightly better than the results for disambiguation of the corpus containing fully analysed sentences only. The reason for this is probably that a very large part of the remaining corpus is the Bible, which contains very few words unknown to the analyser, but which has a syntax more demanding for the disambiguator. The administrative texts contain many unknown words, but they are characterized by a more monotone syntax.

We have also tried to improve the result for the specific gold corpus with a word guesser for the unknown words. The word guesser is made with CG, and gives POS and morphosyntactic analysis of the word in question, based upon the word coda. The mid column in Table 2 shows the results of an analysis of the full corpus, where the analysis phase is proceeded by the word guesser. This information is then given as part of the input to the disambiguator. After the disambiguation phase the guessed readings were conflated to one. As can be seen from the table, the guesser component did not give rise to improved results, on the contrary, we see a slight decrease, as compared to the analysis without a guesser.

The main reason for that is this the disambiguator is still in an initial state, where the bulk of the rules are targeted at specific lemma pairs. When input from the morphological guesser is introduced, the picture is completely altered. Now, homonymy across PoS classes is the rule, and not the exception. The disambiguation rules are not written to handle this situation, and the guesser does not improve the results.

Another weakness of the guesser is that it at present gives suggestions on the basis of coda shape only. In a future version, we will add conditional tests to the guesser, and give suggestions based upon syntactic context as well.

[2] http://giellatekno.uit.no/doc/lang/common/docu-deptags.html

Analysers	Languages		
lexicon and morphology	North Saami analyser	Lule Saami analyser	South Saami analyser
disambiguation	North Saami disambiguation	Lule Saami disambiguation	–
syntatic functions	common Saami analyser		
dependency	common Saami analyser		

Table 1: The common Saami analyser infrastructure. The disambiguation of South Saami is the missing link.

Table 2: Homonymy in South Saami

	Whole corpus 8,7% unkn wrds	Whole corpus with guesser	Fully analysed sentences only
Number of words	218.574	218.574	83.530
Analyses per thousand words			
Analyses with homonymy	1.633	1.633	1.792
Present disambiguation	1.112	1.192	1.248
Lemma + PoS disambuguation	1.061	1.141	1.063
Lemma + PoS disambuguation without distinguishing closed PoS	1.056	1.136	1.058

4.2 Precision and recall

For evaluating the accuracy of the disambiguator, we have used two gold standard corpora.

The general gold corpus is a small balanced corpus containing 100 sentences (30 sentences from the Bible, 30 sentences from fictive texts and 40 sentences from newspapers, altogether 1301 words).

The specific gold corpus is closer to the kind of texts, which the disambiguator is meant for. It is an unknown corpus containing 2329 words, 6,7% of them are unknown for our fst. The corpus contains parts from two texts which could be interesting for extracting terminology – one is the *Convention on the Rights of the Child*, and the other one is from a school curriculum about reindeer herding. The results of the analyses are presented in Table 3.

Looking at the results, the disambiguator has a very good recall, as good as 0.98 for full disambiguation and 0.99 for POW disambiguation. As it stands, the program is thus very careful, to the degree that it almost does not remove correct readings. For full morphosyntactic disambiguation , the precision is lower, 0.87 and 0.88, these are poor results in a CG context. Partly, this is the results of some syntactic idiosyncrasies in our special test corpus. But above all it reflects the immature status of the disambiguator. With only 115 disambiguation rules, compared to the 2-3000 rules usually found in standard CG grammars, 0.87 is a good starting point.

For the task at hand, lemmatisation and POS marking, the precision results are much better, 0.93 and 0.94, respectively. Despite the low number of rules, they are efficient enough to carry out POS disambiguation. The remaining degree of homonymy reported for lemma + POS in Table 2 (1.06) thus comes with a precision and recall of 0.94 and 0.99, respectively.

We tried to improve the disambiguation of the known words, by getting more context for the CG-rules in the disambiguator with help of a word guesser. The testing shows however that giving word guesser analysis to the unknown words, does not improve the disambiguation for the known words.

4.3 Discussion

A full fledged constraint grammar typically contains several thousand rules. The South Saami disambiguator is still in an embryonic state, and contains only 115 rules. With this small

Table 3: Precision and recall

	Special gold corpus		General gold corpus	
Number of words	2329		1301	
Unknown words	6,7%		0	
	Prec	Rec	Prec	Rec
Lemma + full disambiguation	0.876	0.980	0.884	0.968
Lemma + PoS disambuguation	0.939	0.990	0.938	0.981
Lemma + open PoS disambuguation	0.945	0.992	0.994	0.987
Lemma + full disambiguation w/guess	0.877	0.978	-	-
Lemma + PoS disambuguation w/guess	0.940	0.988	-	-
Lemma + open PoS disambuguation w/guess	0.947	0.991	-	-

rule set, we are still able to disambiguate text down to 1.100 lemma + PoS readings per 1000 word forms. The rules were written with full grammatical disambiguation in mind, and a rule set geared towards lemmatisation only could have been made even smaller. Figure 3 shows the cumulative effect of the CG rules. The 20 most efficient rules account for almost 80% of the disambiguation.

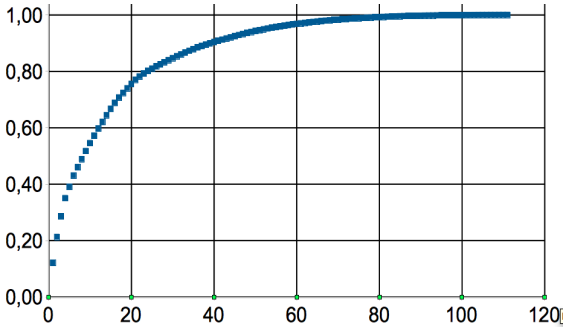

Figure 3: Rule coverage (x = number of rules, y = coverage)

The 10 most efficient CG rules are listed below. For each rule, only the action (select or remove readings) and scope (POS, grammatical feature or lemma), is given. In addition, each rule contains conditional tests for the action in question. For the sake of brevity, these conditions are not given here.

1. IFF: ConNeg if Neg to the left
2. SELECT: Inf is V to the left selects Inf
3. SELECT: A Attr if NP-internal N to the right
4. REMOVE: Imprt if not domain-initial
5. IFF: *goh* is Pcle if in Wackernagel position
6. SELECT: Po, not Pr, if Gen to the left
7. REMOVE: Prefer lexicalised verb to derived
8. REMOVE: *ij* is Periphrastic Neg Prt only if 2nd part of it is present
9. REMOVE: Prefer lexicalised passive to derived
10. REMOVE: Prefer Pers to Dem if no NP-internal N/A/Num to the right

As shown above, the most efficient rules are rules for distinguishing closed PoS. This disambiguation is useful for the rules made for disambiguating open PoS with different lemmas.

Looking now at lexical disambiguation, the 10 most efficient rules for distinguishing between lemmas in open PoS are listed below. The actual word form is given in Italic.

1. SELECT: *Jupmele* – Prefer N Prop to N
2. REMOVE: *Dan* – Prefer Pron Pers to Prop
3. REMOVE: *tjïrrh* – Prefer Po to V
4. REMOVE: Prefer *almetje* N to *elmie* N
5. REMOVE: Prefer *almetje* N to *alma* N
6. REMOVE: Prefer *gïele* N to *gïeledh* V
7. REMOVE: Prefer Adv to A
8. IFF: Interj or other PoS
9. REMOVE: *tjïrrh* – Prefer Po to N
10. SELECT: Prefer V not N

Most of these rules are made specifically for the most frequent lemma pairs having homonym inflectional forms. One improvement strategy might be to make these rules more general and lemma-independent, thereby targeting other lemma-pairs as well.

After disambiguation, there remain 5632 ambiguous word forms, 27.5% of them have the same PoS, and 32.0% of them have the same lemma, as shown in Table 4.

Table 4: Remaining homonymies

	Number of analyses	Percentage
Homonomy with same PoS	1551	27.5%
Homonomy with same lemma	1797	32.0%
Total	5632	100%

The remaining homonymies are mainly of the following types:

1. The same lemma, but different PoS, eg. *juktie* N ('carcass') vs. *juktie* CS ('so that').
2. Different lemmas and different PoS, eg. *vihte* N ('wit') vs. *vihth* Adv ('again').
3. Different lemmas, same PoS and inflection eg. *båetedh* V ('to come') vs. *böötedh* V ('to mend, to pay a fine'). These are the really hard ones to disambiguate.
4. Different lemma, same PoS, but inflection is different (one of them may be derived from the other), eg. *utniedidh* V ('to held') vs. *utnedh* V ('to have, to use').
5. The same lemma has one reading as Proper noun and one as common noun – *Saemie* N ('Saami') vs. *saemie* N ('saami').
6. There are two orthographic variants of the same lemma, which should have been subsumed under the same lemma, eg. *ussjiedidh* V vs. *ussjedidh* V ('think').
7. Derivation vs. lexicalisation, eg. like for *ryöjnesjæjja* N vs. *ryöjnesjidh+V+TV+Der1 +Der/NomAg+N* ('shepherd').

The three first types are true instances of homonymy, many of them can only be resolved by lemma specific rules. The fourth type may or may not be resolved, dependent upon the task at hand. The fifth type is found in some very frequent lemmata. In many instances, this distinction is irrelevant and should be ignored, in other instances one might want to disambiguate them. The last two types are irrelevant for any semantic purposes.

Figure 4 shows the cumulative homonymy for word forms not assigned to a single lemma. Some word forms are very frequent, and writing word specific disambiguation rules for, say, the 50 most common words will already reduce the remaining homonymy with one third.

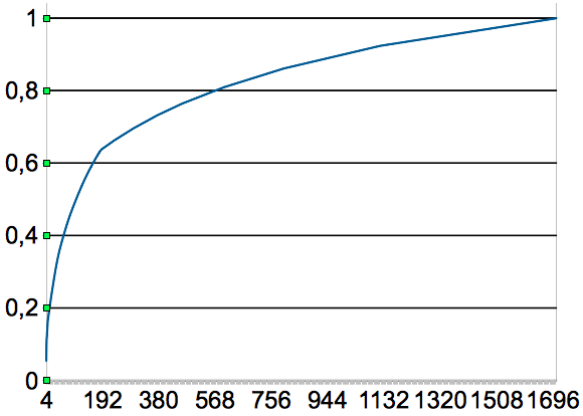

Figure 4: Cumulative homonymy (x = word forms, y = homonymy)

5 Conclusion

The paper has shown that even a tiny set of 115 disambiguation rules is able to achieve quite good results for lemmatising and POS tagging, with a disambiguation rate down at 1.06. In order to disambiguate the full grammatical analysis, a more thorough disambiguation is needed, here the results are about 1.12 even if the corpus contains unknown words. A word guesser doesn't improve the results particularly.

The results also show that the constraint grammar formalism is robust against badly analysed morphological input. As a matter of fact, it scores slightly better on a corpus with an 8.7% error rate, than on a perfect corpus. Even though the difference is probably due to systematic differences in the corpora themselves, it at least shows that constraint grammar is a robust framework for syntactic analysis, capable of dealing with noisy data.

- A small-size CG (115 rules) gives an accuracy of 1.118 - 1.058 readings/word.
- 1/6 of the rule set removes 80% of the homonymy.

- The CG is robust enough to give good disambiguation even with an fst coverage of only 91.3%.
- Adding the results from a morphological guesser did not improve the disambiguation results. More work is needed in order to make use of guesser input.
- The disambiguator's recall is very good, 98.0%. Precision is lower, 87.6-88.6%, and the main focus for improving the South Saami disambiguator will be to improve precision.
- The rule set is a good starting point for a full-fledged disambiguator.

The general conclusion is that even a small-size constraint grammar is able to provide results good enough for POS tagging, lemmatisation, and several other purposes. In order to get a syntactic analysis at the level achieved by other constraint grammars, more work is needed.

References

Lene Antonsen, Trond Trosterud and Linda Wiechetek. 2010. Reusing Grammatical Resources for New Languages *Proceedings of the LREC*. Association for Computational Linguistics, 2782–2789, http://www.lrec-conf.org/proceedings/lrec2010/pdf/254_Paper.pdf

Fred Karlsson, Atro Voutilainen, Juha Heikkilä and Arto Anttila. 1995. *Constraint grammar: a language-independent system for parsing unrestricted text*. Mouton de Gruyter.

Pekka Sammallahti. 1998. *The Saami Languages: an Introduction*. Davvi Girji, Kárášjohka.

VISL-group. 2008. *Constraint Grammar*. http://beta.visl.sdu.dk/constraint_grammar.html Institute of Language and Communication (ISK), University of Southern Denmark