Constraint Grammar based Correction of Grammatical Errors for North Sámi

Linda Wiechetek

Romssa Universitehta, Norway
linda.wiechetek @ uit.no

Abstract

The article describes a grammar checker prototype for North Sámi, a language with agglutinative and inflective features. The grammar checker has been constructed using the rule-based Constraint Grammar formalism. The focus is on the setup of a prototype and diagnosing and correcting grammatical case errors, mostly those that appear with adpositions. Case errors in writing are typical even for native speakers as case errors can result from spelling mistakes. Typical candidates for spelling mistakes are forms containing the letter *á* and those with double consonants. Alternating double and single consonants is a possible case marker. Case errors in an adpositional phrase are common mistakes. Adpositions are typically homonymous (preposition, postposition, adverb) and ask for a genitive case to the left or right of it. Therefore, finding case errors requires a disambiguation of the adposition itself, a correct dependency mapping between the adposition and its dependent and a diagnosis of the case error, which can require homonymy disambiguation of the dependent itself. A deep linguistic analysis including a module for disambiguation, syntactic analysis and dependency annotation is necessary for correcting case errors in adpositional phrases.

1. Introduction

One of the challenges in grammar checking is the diagnosis and correction of morphosyntactic case. The difficulty lies in unambiguously identifying not only local, but partly global context in which a given form of a word available for e.g. case marking is erroneous. Not only needs one to identify relationships over distance (by means of dependency and valency), but also needs one to disambiguate homonymous items by means of morphosyntactic and semantic constraints. Grammar checkers differ in their approach. There are both statistically based (Atwell, 1987), and machine learning based (Izumi et al., 2003) and rule-based (Naber, 2003)[1] approaches.

While a number of grammar checkers exist for majority languages such as English, Spanish etc. a number of people have also developed grammar checking tools for minority languages. Kevin Scannell has the open-source grammar checking tools *Gramadóir* for Irish (Gaeilge), Afrikaans, Akan, Cornish, Esperanto, French, Hiligaynon, Icelandic, Igbo, Languedocien, Scottish Gaelic, Tagalog, Walloon, and Welsh. [2].

The prototype of the grammar checker for North Sámi, a morphologically complex language is written in Constraint Grammar (Karlsson, 2006), using the CG-3 version (`http://visl.sdu.dk/constraint_grammar.html`) which allows for dependency annotation and a number of other features making the analysis more efficient. There are already a number of grammar checkers written in the constraint grammar formalism, for example grammars for the Scandinavian languages, e.g. Swedish (Arppe, 2000), Danish (Bick, 2006) and Norwegian (Bokmål) (Johannessen et al., 2002) and Norwegian (Nynorsk) is being worked on `http://kaldera.no`, additionally one for Esperanto (Lundberg, 2009) and a grammar-checker module for Basque (Oronoz, 2008) dealing with postposition errors and agreement erros.

[1] He is using the opene source language tool approach (`http://languagetool.wikidot.com/papers`).

[2] http://borel.slu.edu/gramadoir/manual/index.html

2. Errors - definition and classification

What is a grammatical error? A grammar checker usually marks errors that can be resolved by means of the morphosyntactic context. That does not only include purely grammatical errors, but also so-called physical errors, i.e. typos which are not caught by a spellchecker, but result in real wordforms, the following errors are "real-word errors". In some cases morphosyntactic context is not sufficient for the resolution of syntactic errors and semantic and lexical information is necessary. In many cases a grammar checker and a spellchecker work together. The grammar checker generally diagnoses an error and suggests a correct alternative. Depending on the target group of the grammar checker, different kinds of errors are being diagnosed by a grammar checker. While language learners (L2) usually make a lot of grammatical errors, first language (L1) users do not, and their errors are more likely to be mechanical errors (vs. cognitive errors (Miłkowski, 2010)) which result in real-word errors and morphosyntactic errors. Other typical errors are copy-paste errors, where the process of copying part of a sentence in one place and inserting it in another place results in erroneous sentence structure (Johannessen et al., 2002). The target group of the North Sámi grammar checker are first language users.

We distinguish four main errortypes, each of which has many sub-errortags. Those are lexical errors, morphosyntactical errors, syntactical errors, and real-word errors.

type	# rules	# tags
lexical	13	12
morphosyntactic	44	18
syntactic	25	18
real word	89	55
altogether	181	103

Table 1: Ruletypes

Real word errors are a common source of errors for first language users. In North Sámi, consonant and vowel lengths

expressed by double consonants and diacritics (a vs. á) can lead to typos resulting in real word errors. The average homonymy between word forms in North Sámi is 2.6, but some word forms get 10 or more different analyses. Sometimes a real-word error can lead to a morphosyntactic error, e.g. where a missing consonant can result in another morphological case of a word. The line is not always easy to draw between error-types as it can be difficult to identify the "intention" behind the error. Mixing up **a** and **á** as in example (1) makes *vuosttáš* (diminutive of vuostá - cheese) 'little cheese' out of *vuosttaš* 'first'. This can be disambiguated fairly easily in a given context.

(1) a. *Otne lei mis vuosttáš logaldallan.
Today was us cheese.diminutive lecture.
'Today we had our little cheese lecture.'

b. Otne lei mis vuosttaš logaldallan.
Today was us first lecture.
'Today we had our first lecture.'

Typical syntactic errors in North Sámi are incorrect verb forms after auxiliaries, errors regarding subordinate clauses, and use of personal instead of possessive pronouns. Lexical errors can resemble syntactic errors, they involve the use of erroneous lexemes in a certain construction, e.g. the use of the wrong postposition, e.g. *badjel* 'over' instead of *bokte* 'via, by means of' as in example (2).

(2) Dat máksá telefuvnna bokte/*badjel.
He pay telephone via/*over.
'He pays via telephone.'

Morphosyntactic errors refer to the morphological structure of a word. They include compound errors (in North Sámi compounds are written in one word), missing hyphens in constructions where compounds are coordinated, use of wrong morphological case in certain syntactic constructions such as in an adpositional phrase, case-number agreement errors, use of wrong case in coordination, use of wrong tenses in subordinate clauses.

3. Corpus

For developing rules a corpus of 193 sentences (4,349 tokens) has been constructed manually from erroneous sentences found in the Giellatekno corpus of North Sámi (18,142,181 tokens, mostly newspaper text) [3], and online blogs (`http://indigenoustweets.com/blogs/se/`).

Constructing an error corpus takes a lot of time and is rather a matter of coincidence than systematic searching. Others use Wikipedia (Miłkowski, 2007) for automatically constructing an errorcorpus. The North Sámi Wikipedia is written by many L2 speakers, which makes it inadequate as a test corpus for a L1 grammar checker. Additionally it is fairly limited in size (110,000 words).

Case errors are fairly "rule-based", they depend on certain fairly straightforward error patterns.

Real-word errors can be induced automatically by e.g. changing the character *á* to *a* vice versa. Typically, verbs ending in *-it/-at* such as *speadjalastit* 'to mirror, reflect' are a rich source for errors. The participle of the verb *speadjalastit* 'to mirror, reflect' is *speadjalastán*. When *á* is replaced by *a* on the last syllable, it becomes *speadjalastan* which is a compound of *speadjal* 'mirror' and *astat* 'have time', which is a possible but unlikely word.

Changing consonant clusters (single to double consonants vice versa) is another way to induce real-word errors automatically. Good candidates are genitive, nominative, locative, genitive possessive forms of nouns such as *várri* 'mountain' (nominative) vs. *vári* (genitive, accusative).

The testcorpus for postpositional case errors consists of 2000 sentences taken from the Giellatekno corpus for North Sámi. It contains 1000 correct sentences and 1000 sentences where a case error has been inserted.

Given that case errors are fairly straightforward (choice of 6 possible morphological cases) the error has been inserted manually, i.e. the correct case has been changed to an incorrect one. No other parts of the sentence have been changed.

4. How to set up a grammar checker - general architecture

4.1. Existing tools

The North Sámi rule-based grammar checker is written in Constraint Grammar formalism (Karlsson, 2006) and makes use of and enhances existing resources.

The morphological analyzers are implemented with finite-state transducers and compiled with the Xerox compilers `twolc` and `lexc` (Beesley and Karttunen, 2003). The other option is HFST[4].

The morphosyntactic disambiguators analyze the text syntactically, and at the same time disambiguate morphological and syntactic readings by means of context rules ideally leaving only one correct analysis. They are implemented in the CG-framework (Karlsson, 2006) and are based on manually written morphosyntactic rules that select and discard syntactic analyses. They further add grammatical functions and add dependency relations to the analysis. The rules are compiled with vislcg3 [5].

4.2. New modules and adaptions

In addition to the use of existing resources, the following tools have been created and adapted: The noun lexicon has been enriched by means of semantic tags, grammar-checker-specific rules and modification of rules in the disambiguator, a separate grammar-checker grammar including errortag mapping, dependency, valency and semantic role annotation.

4.2.1. Lexicon

The lexicon is enhanced by a number of semantic categories inspired by Bick's 150-200 semantic pro-

[3] `http://giellatekno.uit.no/doc/lang/corp/corpus-sme.html`

[4] `http://www.ling.helsinki.fi/kieliteknologia/tutkimus/hfst/`

[5] `http://beta.visl.sdu.dk/constraint_grammar.html`

totypes (`http://gramtrans.com/deepdict/semantic-prototypes`). The categorization is not complete as categories are created as they are needed. Also the Basque grammar checker makes use of semantic categories for the disambiguation of postpositions and considers them necessary.
Currently, there are the following categories: *Masculine, Feminine, Surname, Place, Organisation, Object, Animate, Plant, Human, Group, Time, Text, Route, Measure, Weather, Building, Education, Clothes*.

4.2.2. Disambiguator

The disambiguator needs adaptations in order to be used for grammar checking. As mentioned by Bick (2006) and Johannessen et al. (2002), correct readings sometimes get discarded because of the erroneous context. This is due to contrary philosophies of grammar checking and regular disambiguation. In regular disambiguation, the assumption is made that input text is correct, and based on that assumption words are analyzed syntactically. The grammar checker on the other hand gets both correct and erroneous text as an input. The disambiguator needs to be adapted to those conditions. It needs to output a correct analysis despite errors in the context. The focus does not lie on removing as much ambiguity as possible, but on not discarding correct readings. Therefore adaptions have been made. As Johannessen et al. (2002), we are using a relaxed grammar checker with specific rules. Bick (2006) on the other hand runs the rules twice, some of them before and others only after applying the grammar checker.
Rules that remove correct readings are removed or changed in favor of leaving more ambiguity. Typical modifications consist in negating a typical context in which a certain reading should not be discarded. Rules for specific errortypes such as case errors in adpositional phrases are added. Instead of basing their whole disambiguation on morphosyntactic constraints, lexical and semantic constraints are being used.

4.2.3. Grammar checker module

The grammar-checker module is a separate module and constructed in the same way as the disambiguation grammar. The pedagaogical programs (*Oahpa*) for North Sámi use a basic grammar checker (Antonsen et al., 2009). There are several reasons for starting from scratch: the target group is a different one (L1 users vs. L2 users), the type of text serving as an input. While the input to *Oahpa* consists in very simple sentences, the grammar checker has to deal with complex sentences. In constraint grammar, two types of rules can add tags to words. MAP rules only allow a single error tag to be added per word, where ADD rules permit more than one to be added. The Sámi grammar checker uses ADD rules, while the Basque one uses MAP.
The idea behind that is that a word can have multiple errors which can be added onto each other. However, in some cases, rules do block each other. The architecture within the grammar checker is the following: The first rule determines the grammaticality of a sentence based on the existence of a finite verb (unless there is a given context for leaving it out, e.g. headers, answers in a dialogue etc.). The missing-verb tag blocks the application of certain other rules.
The grammar checker contains both errortags and correct tags. Correct tags have the function of analyzing complex conditions for certain grammatically correct constructions, which form an exception to other error rules.

```
ADD:corr-not-compound (&corr-not-compound)
 TARGET CNOUN IF
 (0C (N Sg Nom) LINK 1 CNOUN LINK p V) ;
```

This rule adds a correct tag to a noun (CNOUN) saying there is no compound error in the case where a dependency relation to a verb can be determined (the parent (p) of the noun is a verb (V)).
Errortag-matching rules refer to the correct tag constraining the mapping of an errortag.
The following rules are ordered in the this way:

1. real-word errors rules
2. dependency mapping rules
3. compound rules
4. lexical error rules
5. syntactical error rules
6. morphosyntactical error rules

Real-word errors, compound errors and lexical errors can constrain the other rules, which is why they are applied first. Dependencies help to identify syntactical and morphosyntactical errors, which is why they are mapped before the respective rules for that. CG3 (Bick, 2006) lets us add dependencies in the same grammar. They are used to construct partial trees for adpositional phrases and argument structures of verbs. Those help recognizing errors where the required dependent could not be matched. The dependency trees are very specific and only partial trees are mapped. Using a complete dependency annotations require full disambiguation with a high F-score, but as mentioned before, many times this is prevented by erroneous context and correct readings are discarded/a full disambiguation is not possible. Therefore a full dependency analysis will not give sufficiently good results. Oronoz (2008) uses dependencies to detect agreement errors, but the dependency analysis of erroneous text give such unreliable results that errortag-mapping is not sucessful.

5. Diagnosis and correction of case errors

A grammar checker needs to diagnose an error (identify its cause) and correct it (suggest an alternative the erronous item is substituted for).
Currently, the errortags include both the error and the correction, e.g.: *&msyn-gen-before-postp* (morphosyntactical error, there should be a genitive case before a postposition), *&msyn-gen-after-prep* (morphosyntactical error, there should be a genitive case after a preposition). The correction is still implicit, the correct form is not being generated yet.
Correcting case is one of the more challenging rules for North Sámi grammar checking as it can often involves long-distance relationships (argument structure) and need large contexts to identify the error.

Language learners typically make a lot of case errors, which is partly due to them infering syntactic constructions from their native language. Case errors that are made by native language users are mostly spelling errors in disguise, i.e. spelling errors resulting in a wrong case (similarly to a real-word errors). In the following example (3), the use of the single consonant **d** instead of **dd** makes a possessive form in nominative case out of the locative form that is required by the verb *jearrat* 'ask'.

(3) Mun jearan
I ask girl.sg.loc/*girl.sg.nom.px.sg3
nieiddas/*nieidas.

'I ask the girl.'

(4) Liikon
Like.1.sg.prs reindeer.meat.ill/reindeer.meat.acc
bohccobirgui/*bohccobierggu.

'I like reindeer meat.'

Another source of case errors are improper use of correct valency possibly due to influence from the majority case structures, e.g. *liikot* 'like' asks for illative case in Sámi, but for accusative case in Norwegian.
Case errors can be influenced locally (adpositions ask for genitive case) or globally (verbs asking for a certain case of their arguments).
Locally influenced case errors are errors in adpositional constructions as in example (5) and in partitive constructions as in example (6).

(5) Sii bidje bálgá mielde
They went path along hill.gen/*hill.nom
dievá/*dievvá badjel.
over.
'They went along the path over the hill'

(6) Muhtun osiin
In some parts
álbmotmeahcis/*álbmotmeahccis leat
nationalpark.Loc/nationalpark.Nom.PxSg3 have
ealggat bilistan nuorramuoraid.
elks destroyed young.trees.
'In some parts of the nationalpark, elks have destroyed the young trees.'

5.0.4. Adpositional phrases

Error detection of case in adpositional phrases is complicated by the extensive homonymy of adpositions themselves. Currently there are 305 adpositions and 1089 possible analyses of those (3.6 possible analyses per postposition).
An incorrect case in front of a postposition is detected in the following way. At first particular rules disambiguate the postposition itself. There are currently 60 rules, mostly rules selecting alternative readings to adpositional readings (e.g. adverbial readings), which are modified as not to discard correct adpositional analyses. *(NEGATE 0 Po LINK -1 Gen)* says that the rule should not apply to potential postpositions with a genitive to its left. Since those rules are assuming correct text as their input and often rely on correct text, the modifications prevent that correct readings in erroneous text are discarded.
51 more rules are disambiguation rules specifically choosing or discarding adpositional readings. Since the morphosyntactic context is not reliable - the main cue for selecting a postposition/preposition is a preceding/following genitive - semantic cues and valency information is used to rule out adpositional or alternative analyses. Especially the postpositions that are homonymous to adverbs and stand alone pose a problem to error tag matching. Nouns are usually rare forms or can be disambiguated by valency information. Verb readings can sometimes be difficult to disambiguate too unless they are rare forms (such as infinite forms that are used in specific contexts only, e.g. the verbgenitive form). Some rare noun forms are possessive suffixed nouns that require a human subject in the same person.
The following rule selects an adverbial reading if the lexeme is *sisa* 'into' and followed by a noun of the category *building* in illative case unless the word to the left of it is in genitive case.

```
SELECT:GramPo Adv IF (0 ("sisa") LINK 1
 BUILDING LINK 0 Ill)(NOT -1 Gen) ;
```

Other rules select a certain reading if the word is part of a multiword expression as *ieš alddis* 'by himself' where the form *alddis* is not a postposition with a possessive suffix ending, but a pronoun, *ieš* 'oneself', in locative case with a possessive suffix ending. This could possibly resolved in a different way.

```
REMOVE:GramPo ("alde") IF (0 PxSg3)
  (-1 ("ieš"));
```

The following rules select an adverbial reading if a certain set of verbs are there and a noun of the category CLOTHES is there.

```
SELECT:GramPo (Adv) IF (0 ("ala") LINK *0
 ("bidjat")  OR ("coggat") OR ("coggalit"))
 (-1 CLOTHES);
```

7 rules of the type SETPARENT/SETCHILD set the dependency relation of genitive nouns/pronouns/numerals to a following postposition that either belongs to the set of postpositions that can only be a postposition or to a disambiguated postposition which is following the genitive. Other rules are dealing with with prepositions.

```
SETPARENT Gen TO (*1C Po BARRIER S-BOUNDARY);
```

This rule depends on good disambiguation. If a secure postposition follows, the genitive is linked via depenedency to the postposition.
The following illustrates the different philosophies of the disambiguation grammars. The regular disambiguation

grammar selects an adverb if the noun preceding the adposition is in nominative case, and a postposition if the preceding noun is in genitive case. But in grammar checking, the case of the preceding noun cannot be used as the only criterium for disambiguation of PoS because it can be erroneous. In order to resolve the adverb-adposition ambiguity, other constraints need to be used. In example (7-a), *gorži* 'waterfall' is a potentially erronous form. The difficulty here lies in that both *gorži vuolde* and *goržži vuolde* are possible bigrams because the nominative form can potentially be a subject/predicative of the sentence making *vuolde* `under, beneath' an adverb. The decisive element that allows us to disambiguate between the adverb and adpositional reading of *vuolde* `under, beneath', is the habitive *mus* 'I.locativ' together with the verb *leat* 'to be'. Without another potential predicative in the sentence, the adpositional reading is discarded in (7-a), and *vuolde* `under, beneath' is analyzed as an adverb.

(7) a. Mus lea gorži vuolde.
I.loc be.Sg3 waterfall.nom beneath.
'I have a waterfall beneath (= under me).'

b. *Mus lea goržži vuolde.
I.loc be.Sg3 waterfall.gen under.
'I have under a waterfall.'

The rule selects the adverbial reading of *vuolde* `under, beneath' if there is a habitive and the verb *leat* 'be' to the left of it.

```
SELECT:GramPo (Adv) IF (0 ("vuolde")
LINK -1 N) (*-1 ("leat"))(*-1 @HAB) ;
```

The final step is the grammar checker error mapping rule itself, which maps an errortag **&msyn-gen-before-postp** to the potential dependent of an adposition (noun, pronoun, numeral, adjective) unless it is in genitive case and a dependent of the adposition.

```
ADD:gen-before-postp (&msyn-gen-before-postp)
TARGET NP-HEAD - ABBR IF (NOT 0 Gen)(1C Po)
(NEGATE 1 N);
```

In some cases, the wrong postposition is selected, e.g. *rastá* 'through' instead of *mielde* 'along', *badjel* 'over' instead of *rastá* 'through' etc. While in the Basque grammar checker, all errors in an adpositional phrase are treated as one type (due to other categorizations of case vs. adpositions), in North Sámi those are considered to be lexical errors, while the previously discussed error type is considered to be a morphosyntactic error.

6. Evaluation

For the evaluation, the rules for 5 adpositions represented in table (2) are evaluated.

The postpositions can be used in a local sense, many of them have other uses too. *ala* 'on' for example can be used with a number of psychological verbs such as *suhttat* 'get angry at', *dorvvastit* 'rely on', *luohttit* 'trust'. *Bokte* 'via, by means of' is used as via as in *media bokte* 'by means of the media' etc. *Rastá* 'across' as in *rastá cearddaid* 'across ethnicities'. Difficulties are especially there to disambiguate between the verb and the postposition if the case is wrong. but e.g. *boktit* 'wake' usually asks for an animate object.

adposition	translation	homonymy
ala/nala	onto	postp, adv, verb (aldat 'get closer')
alde/nalde	on	postp, adv
badjel	over	postp, prep
bokte	via	postp, verb (boktit 'wake')
rastá	across	postp, prep, adv

Table 2: Adposition homonymy

The testcorpus for evaluation contains 15,968 tokens and consists of 200 sentences for each postposition, 100 with correct case and 100 with incorrect case to evaluate both precision/recall and false alarms. An important task that remains is testing precision on a large corpus (e.g. newspaper corpus).

	err corp	corr corp	complete corp
tokens	16206	16105	18,142,181
errors	1000	0	-
detected errors	825	0	65
false alarms	23	-	14
precision	0.98	-	0.78
recall	0.83	-	-
f-score	0.93	-	-

Table 3: Quantitative evaluation

In the small corpus, there are few false alarms, precision is at 0.98. The recall is at 0.83.

The false alarms are mostly due to errors in disambiguation: adverbial vs. adpositional reading (11), genitive vs. accusative reading (5), verbal vs. nominal reading (1).

The reason for false alarms are exclusively disambiguation problems, typically adverb vs. adposition. But also disambiguation problems of the previous word (the dependent) can cause false alarms, either in terms of wrong part of speech (verb vs. noun) or the wrong case in terms of genitive/accusative homonymy. The disambiguation grammar is responsible for these false alarms. For example *čuovga alde* 'light on', *gákti alde* 'Sámi clothes on' can either be used in an expression where on means something like "turned on" (light) vs. "wearing" (clothes). Here, the grammar checker grammar is causing the false alarms.

The second measure is recall. The reasons for undetected errors are shown in table 4.

Sometimes more than one errortype apply and the reason can be a combination of several reasons. In many cases the adverb instead of the adposition is erroneously disambiguated with a preceding nominative, illative or locative. Another challenge is removing the verb first person dual forms of *aldat* 'come closer' and *boktit* 'wake', which still get selected too often. In some cases pre- vs. postposition

type	number
adverb not adposition	51
grammar-checker rule does not hit	49
pre- vs. postposition	40
verb not adposition	15
erronoeus disambiguation of previous word	6

Table 4: Undetected errors

disambiguation needs to be improved. This is most difficult where both a preceding and a following noun are there, and where the noun not belonging to the adpositional phrase is a genitive.

When the noun phrase is complex, e.g. *badjel min ipmárdusrájit* 'across our understanding-borders,*Vuođđoláhka § 110 a bokte* `by means of constitution law § 110 a' the grammar checker rules do not always hit. This can be improved by using more detailed constraints.

For testing precision a larger 'natural' corpus is needed, where not necessarily many errors can be found. I did a small test, with two of the postpositions 'ala/nala' ('onto') and 'rastá'. 65 errors were detected, 51 of those correctly identified errors, and 14 false alarms, giving a precision of 0.78 %.

7. Conclusion

The construction of a grammar checker for North Sámi includes a number of challenges. The basis for the grammatical sentence analysis is no longer a correct sentence, but a potentially erroneous one. Not knowing where the errors might be together with high degree of homonymy of North Sámi word forms make morphosyntactic information partly unreliable for error detection. Since one cannot trust the grammar, one needs to make use of semantics and lexicon that can unambiguously identify the word, e.g. a postposition. A good disambiguation of the adpositions is the key for detecting case errors in adpositional phrases. Detailed adposition-specific rules that refer to the semantic context require some work, but are fairly successful in identifying the correct reading. The qualitative evaluation has revealed holes in the disambiguation of adpositions, but also potential for improvement.

As a next step, I would like to use the results from the qualitative analysis to improve both disambiguation and grammar checking rules, and make a new thorough analysis of precision on the complete Giellatekno corpus.

With regard to the grammar checker as a whole, a number of tasks remain to be done. After resolving case errors that depend on local context, resolving those depending on a global context (argument structure of the verb) by means of valency information can be attempted. I predict the task to be much more difficult than detecting errors in the adpositional phrase context and it remains interesting to see if effective solutions can be found. The resolution of real word errors and agreement errors are another important field of development.

As previous Constraint-based grammar checkers show, e.g. (Johannessen et al., 2002), Constraint Grammar-based grammar checkers can be integrated in Microsoft Office. The endproduct of the grammar checker for North Sámi should be integrated into Microsoft Office, Open Office, MacOSX, and InDesign as the main Sámi newspapers and publishing houses use InDesign.

8. Acknowledgments

Thomas Omma has constructed the error corpus, and written parts of the grammar. I would like to thank him for his linguistic help and advice on grammatical constructions, and Francis Tyers for his eagle eye for grammatical and formatting errors and his positivity.

9. References

L. Antonsen, S. Huhmarniemi, and T. Trosterud. 2009. Constraint grammar in dialogue systems. In *NEALT Proceedings Series 2009*, volume Volume 8, pages 13–21.

A. Arppe. 2000. Developing a grammar checker for swedish. In *Proceedings from the 12th Nordiske datalingvistikkdager*, Trondheim.

E. S. Atwell. 1987. How to detect grammatical errors in a text without parsing it. In *Proc. 3rd EACL*, pages 38––45, Copenhagen.

K. R. Beesley and L. Karttunen. 2003. *Finite State Morphology*. CSLI publications in Computational Linguistics, USA.

E. Bick. 2006. A constraint grammar based spellchecker for danish with a special focus on dyslexics. *A Man of Measure: Festschrift in Honour of Fred Karlsson on his 60th Birthday. Special Supplement to SKY Jounal of Linguistics*, 19:387–396.

E. Izumi, K. Uchimoto, T. Saiga, T. Supnithi, and H. Isahara. 2003. Automatic error detection in the japanese learners english spoken data. In *Companion Volume to Proc. ACL'03*, pages 145—148, Sapporo, Japan.

J. Bondi Johannessen, K. Hagen, and P. Lane. 2002. The performance of a grammar checker with deviant language input. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1223–1227, Taipei, Taiwan.

F. Karlsson. 2006. *Constraint Grammar - A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin.

S. Petrović Lundberg. 2009. Collecting and processing error samples for a constraint grammar-based language helper for esperanto. Master's thesis, Stockholms Universitet, Department of Linguistics.

M. Miłkowski. 2007. Automated building of error corpora of polish. *Corpus Linguistics, Computer Tools, and Applications – State of the Art. PALC 2007, Peter Lang. Internationaler Verlag der Wissenschaften 2008*, pages 631–639.

M. Miłkowski. 2010. Developing an open-source, rule-based proofreading tool. *Software – Practice and Experience 2010*, 40(7):543–566.

D. Naber. 2003. A rule-based style and grammar checker diploma thesis. Master's thesis, University of Bielefeld.

M. Oronoz. 2008. *Euskarazko errore sintaktikoak detektatzeko eta zuzentzeko baliabideen garapena: datak, postposizio-lokuzioak eta komunztadura*. Ph.D. thesis, Lengoaia eta Sistema Informatikoak Saila. Donostia.