Reusing a Multi-lingual Setup to Bootstrap a Grammar Checker for a Very Low Resource Language without Data

Inga Lill Sigga Mikkelsen **Linda Wiechetek** **Flammie A Pirinen**
UiT Norgga árktalaš universitehta
inga.l.mikkelsen@uit.no Divvun tommi.pirinen@uit.no
Norway
linda.wiechetek@uit.no

Abstract

Grammar checkers (GEC) are needed for digital language survival. Very low resource languages like Lule Sámi with less than 3,000 speakers need to hurry to build these tools, but do not have the big corpus data that are required for the construction of machine learning tools. We present a rule-based tool and a workflow where the work done for a related language can speed up the process. We use an existing grammar to infer rules for the new language, and we do not need a large gold corpus of annotated grammar errors, but a smaller corpus of regression tests is built while developing the tool. We present a test case for Lule Sámi reusing resources from North Sámi, show how we achieve a categorisation of the most frequent errors, and present a preliminary evaluation of the system. We hope this serves as an inspiration for small languages that need advanced tools in a limited amount of time, but do not have big data.

1 Introduction

Language tools for very low resource languages are urgently needed to support language maintenance, but also it takes a long time to develop them. An existing multilingual infrastructure and existing tools that can be reused can speed up the process. In this article, we describe the process of making a Lule Sámi GEC together with a preliminary categorization of frequent Lule Sámi errors. Lule Sámi is on the lower end of lower resource language. It can benefit from North Sámi which is closely related and has a well-functioning grammar checker.

The reuse of existing knowledge is an important concept in effective development of new grammar checkers in multilingual infrastructures. With this work we would like to set an example of how high-end complex NLP tools can be made, in less time, by taking existing tools as a frame. The following tools were already ready-made: an FST-based morphological analyser, a morpho-syntactic disambiguator developed for correct text, and a multi-lingual infrastructure that contains scripts to build the grammar checker (among other applications). Our work took altogether 120 hours, (40 hours of meetings of two linguists (one of them native speaker) and 40 hours of work of one native speaker linguist).

For related languages we can even reuse rules and sets (prenominal modifiers, sentence barriers). But for example, lexemes have to be translated. This article will show in detail what can be reused, and which factors need special focus as they are language specific – many times it is systematic homonymies, and definitely idiosyncratic homonymies. In addition, we will evaluate the Lule Sámi grammar checker and point out future steps for improvement.

2 Background

2.1 Language and resources

Lule Sámi is spoken in northern Sweden and Norway, with an estimated 800-3,000 speakers (Sammallahti, 1998; Kuoljok, 2002; Svonni, 2008; Rydving, 2013; Moseley, 2010). The Lule Sámi written language was approved in 1983 (Magga, 1994). The first Lule Sámi spell checker was launched in 2007. Lule Sámi is a morphologically complex language, for more details see Ylikoski (2022).

In 2013 the Lule Sámi gold corpus of writing errors was built.[1] The gold corpus consists of 32,202 words with 3,772 marked writing errors. The goal of this error marked-up corpus was to test if the spellchecker corresponds to relevant quality requirements, by running the spell checker

[1] https://github.com/giellalt/lang-smj/

Proceedings of the Fifth Workshop on the Use of Computational Methods in the Study of Endangered Languages, pages 149 - 158

on an error corpus, where spelling errors were manually marked and corrected. It was supposed to be usable for testing grammar checkers with some processing, and therefore also marked syntactic, morpho-syntactic and lexical errors. The texts gathered for the gold corpus were written by native Lule Sámi speakers and had neither been spellchecked nor proofread.

Speakers of Lule Sámi do not have a long written tradition, this amount of errors in the gold corpus show that native speakers of Lule Sámi are in need of tools helping them in the writing process. 1,774 of the errors in the gold corpus are non-word errors (i.e. misspellings that result in a non-existent form, non-word error, as opposed to real word errors where the misspelling results in an existing 'wrong' form), found by the spellchecker, the remaining 1,998 errors are morpho-syntactic, syntactic, word choice and formatting errors, which only a grammar checker can detect and correct. Lule Sámi is by UNESCO classified as a severely endangered language. For the (re)vitalisation of a language, it is important that the language is actually being used. With a (re)vitalisation perspective, a grammar checker for Lule Sámi will make it easier for people to use Lule Sámi in writing, which will increase the use of written Lule Sámi.

The marking and correcting of errors for the gold corpus is the first systematic work on Lule Sámi writing errors. So far, this gold corpus has not been used to analyse and describe error types characteristic for Lule Sámi. Our own experiences from proofreading and from the work with North Sámi were therefore the starting point for developing grammar rules.

2.2 Framework

The technological implementation of our grammar checker is based on well-established technologies in the rule-based natural language processing: finite-state automata for morphological analysis (Beesley and Karttunen, 2003; Lindén et al., 2013) and constraint grammar (Karlsson, 1990b; Didriksen, 2010) for syntactic and semantic as well as other sentence-level processing. The Lule Sámi has an existing morphological analyser and lexicon publicly available[2], which were originally imported from North Sámi with all rules and set specifications and then adapted to Lule Sámi. Antonsen et al. (2010) report F-scores of 0.95 for part-of-speech (PoS) disambiguation, 0.88 for disambiguation of inflection and derivation, and 0.86 for assignment of grammatical functions (syntax) for the Lule Sámi analyser.

The system is built on a pipeline of modules: we process the input text with morphological analysers and tokenisers to get annotated texts, then disambiguate and then apply grammar rules on the disambiguated sentences, c.f. Figure 1.

It is noteworthy, that the system is part of a multilingual infrastructure *GiellaLT*, which includes numerous languages — 130 altogether.

The grammar checker takes input from the finite-state transducer (*FST*) to a number of other modules, the core of which are several Constraint Grammar modules for tokenisation disambiguation, morpho-syntactic disambiguation and a module for error detection and correction. The full modular structure (Figure 1) is described in Wiechetek (2019). We are using finite-state morphology (Beesley and Karttunen, 2003) to model word formation processes. The technology behind our *FSTs* is described in Pirinen (2014). Constraint Grammar is a rule-based formalism for writing disambiguation and syntactic annotation grammars (Karlsson, 1990a; Karlsson et al., 1995). In our work, we use the free open source implementation VISLCG-3 (Bick and Didriksen, 2015). All components are compiled and built using the *GiellaLT* infrastructure (Moshagen et al., 2013). The code and data for the model is available for download [3].

The syntactic context is specified in handwritten Constraint Grammar rules. The ADD-rule below adds an error tag (identified by the tag `&real-negSg3-negSg2`) to the negation verb *ij* '(to) not' as in example (1) if it is a 3rd person singular verb and to its left there is a 2nd person singular pronoun in nominative case. The context condition further specifies that there cannot be any tokens specifying a sentence barrier, a subjunction, conjunction or a finite verb in between for the rule to apply.

(1) Dån **ittjij** boade guossáj.
you NEG.PAST.SG3 come guest.ILL
'You didn't visit.'

```
ADD (&real-negSg3-negSg2) TARGET ("ij")
IF (0 (Sg3))
```

[2] https://github.com/giellalt/lang-smj/

[3] https://github.com/giellalt/lang-smj/

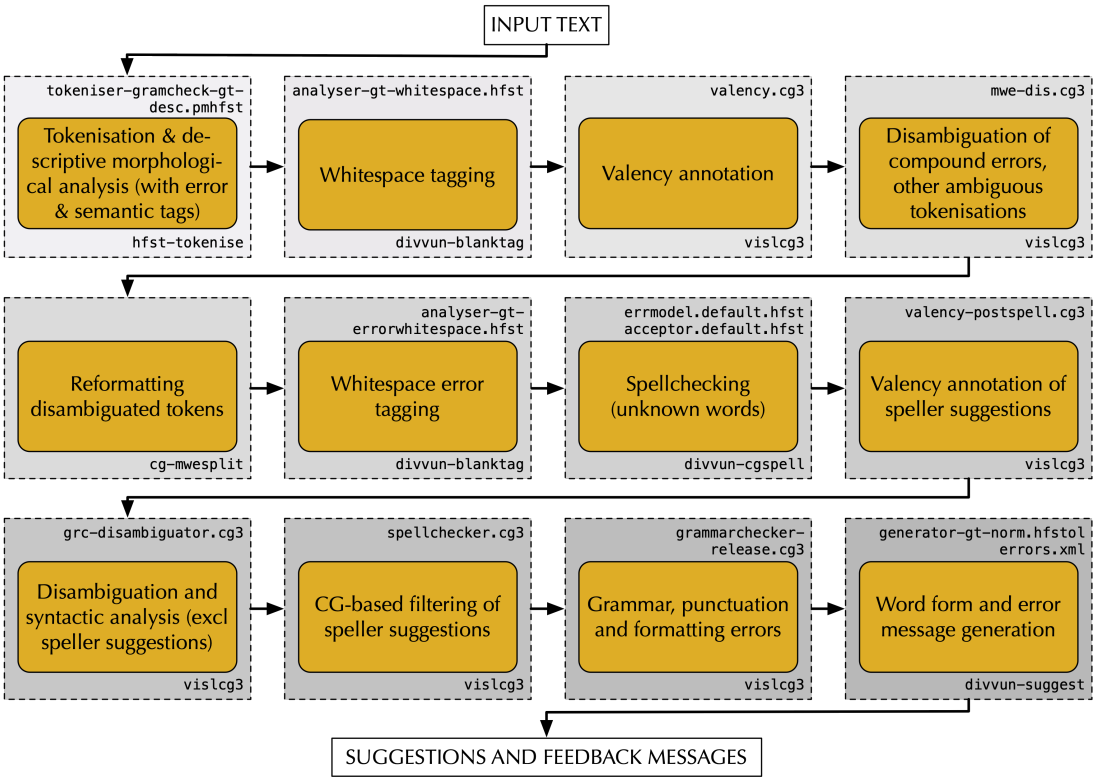

Figure 1: Structure of a grammar checker

```
(*-1 (Pron Nom Sg2)
BARRIER S-BOUNDARY OR
CS OR CC OR VFIN) ;
```

3 Setup

In this section, we answer the question of how to set up a grammar checker for a new language in *GiellaLT*. The resources we need are:

1. Word-based tools:
   - a tokeniser / handling of multiword entities etc.
   - an FST-based morphological analyser
   - a spellchecker
2. Sentence-based tools:
   - a disambiguator (that can deal with erroneous input)
   - a syntactic analyser
   - a number of phonological or morpho-syntactic sets to categorise groups of words
   - error detection/correction rules for a set of frequent errors
3. A set of frequent error types
4. Regression tests (error-marked up test sentences)

Unlike machine learning, this approach is not dependent on a large amount of text data or a gold corpus. To develop a grammar checker, we only need several test sentences containing the errors in question. (Wiechetek et al., 2021) However, in the absence of a fully error marked-up text corpus, finding frequent errors is a challenge. We therefore provide a scheme based on our experience with finding common errors (for the North Sámi grammar checker) as a guideline for work on new languages. This scheme serves any language, but our experience is based on morphologically richer languages.

Error types can be divided into three main categories:

1. phonology-/typography-based errors
2. (morpho-)syntactic errors
3. writing convention-based errors

Phonology-/typography-based errors can be based on diacritics, vowel/consonant length, silent endings in certain contexts (*-ij* pronounced *-i*),

divergence pronunciation/writing and homophone words.

Writing/formatting conventions apply to compounding (one vs. several words, hyphen), quotation marks, comma and punctuation in general. Morpho-syntactic and syntactic errors can be subdivided into verb-, NP-internal and VP-internal issues. NP-internal issues can be about prepositions and postpositions and their case restrictions, adjective agreement /forms in attributive/predicative positions, and relative pronoun agreement with its anaphora in number, gender and animacy.

Verb internal issues concern the auxiliary construction, negation phrases (where negation is expressed by a verb) and other periphrastic verb constructions.

VP internal issues, on the other hand, are more global and concern subject-verb agreement, subclauses formation, subcategorisation in general and case marking of object/adverbial and word order.

In addition to that, the choice of error types will depend on efficiency as well, that means which error types can rules generalize over, and which error types are very word specific. Very word specific work that cannot be generalized may not be so efficient.

3.1 Reuse of resources

Reusing (particularly North Sámi) resources to create Lule Sámi tools goes back as far as 2005, where the North Sámi descriptive morpho-syntactic analyser/disambiguator was used to disambiguate Lule Sámi text and adapting work started. A disambiguator is a tool that resolves homonymy in a given syntactic context, and is an essential tool in sentence-level text processing. This tool was already available when we started our work. However, the initial goal of sentence analysis is based on correct input. We therefore had to adapt the tool to fit error input, e.g. by removing rules that were too strict and paying closer attention to misspelled word forms that can be confused with correct forms. In the course of time, other tools or modules have been copied over to Lule Sámi and been reused with or without adaptations, thereby creating lower-cost tools for Lule Sámi, cf. Table 1. Another tool that was already available when we started to build our GEC was the Lule Sámi morphological analyser. It had previously been constructed from scratch, starting from a common template used in the *GiellaLT* infrastructure.

Tool	**Reuse**	**Adaption**
Analysis tools		
FST	existing	NONE
disambiguator	from sme	set specs
		rules
tokeniser	from sme	NONE
Error detection/correction tools		
disambiguator	from sme	to fit err input
real w err rules	NEW	-
congr rules	from sme	sets
	from sme	homonymies
Other		
regression tests	NEW	-
corpus mark-up	from sme	applied to
		smj text

Table 1: Reuse of resources for Lule Sámi (sme= North Sámi, smj= Lule Sámi)

Based on our experience, we have found a following workflow to be very effective in creating a new grammar checker: We use the normative morphological analyser and a tokeniser with grammatical tokenisation disambiguation. This is relevant when deciding if two words written apart have a syntactic relation or are simple compound errors. In addition, there, we use a FST-based spellchecker. The descriptive disambiguator/syntactic analyser was first taken as it is to be included in the Lule Sámi grammar checker. However, we found that the need for adaptions was urgent, and we needed a separate version of it specifically for potentially erroneous input. The difference to the descriptive disambiguator lays in the objective. The descriptive disambiguator aims at a reduction of homonymy (risking to some degree that correct analyses get lost). The grammar checker disambiguator, on the other hand, needs disambiguation only to get an idea of the sentence to find the error, but is dependent on finding error-analyses even if they do not make sense in the context, so homonymy is not to be reduced to a point where error readings disappear. The descriptive disambiguator is adapted on the fly, so basically every time testing runs into problems, the respective rules are traced and either eliminated or adapted to erroneous input. In some cases, we also noticed general errors in the rules that lead to an improvement of the descriptive disambiguator.

The error detection/correction module needed to be written from scratch at first glance. However, at second glance, there are parts that could be reused as well. Simple sets and lists were copied over from the Lule Sámi descriptive disambiguator. Semantic groupings of words developed in the process of North Sámi grammar checking were directly copied over from the North Sámi grammar checker, and lexical items translated to Lule Sámi as in the case of the following set *DOPPE* (the first of which is the North Sámi original, and the second of which is the translated Lule Sámi one), which generalises over static place-adverbs:

```
LIST DOPPE = "badjin" "bajil"
"dakko" "dá" "dákko" "dáppe" "dás"
"diekko" "dieppe" "do" "dokko"
"doppe" "duo" "duokko" "duoppe"
"olgun" ;

LIST DOPPE = "badjen" "dáppe"
"duoppe" "dåppe" "dággu" "daggu"
"duoggu" "dåggu" "dánna" "danna"
"duonna" "dånna" "dåhku" "duohku"
"ålggon" ;
```

As regards rules, the error types based on orthographic or phonetic similarity needed to be written from scratch, as they differ in North Sámi and Lule Sámi, as do possible contexts of errors that need to pay attention to homonymies. Especially systematic homonymies are partly different to North Sámi. However, some of them are the same in North Sámi and Lule Sámi, cf. Table 2. One of them is the homonymy between plural inessive (Lule Sámi) /locative (North Sámi) and singular comitative nouns, and between singular elative (Lule Sámi) /locative (North Sámi) and 3rd person singular possessive accusative singular nouns.

Not all rules needed to be written from scratch, certain rule types were reused from North Sámi. Subject-verb agreement rules are well-suited to be copy-pasted from North Sámi to Lule Sámi. With some tag adaptations, they were included into the Lule Sámi grammar checker.

3.2 Errors in Lule Sámi

When working with the Lule Sámi grammar checker, we wanted to start with errors made by high proficiency writers rather than language learners. That way we can have a functioning grammar checker for texts with very few errors and introduce more complex errors along the way.

Homonymy	**Lule S.**	**North S.**
Verbs		
PRS PL3 – PRT SG2	sjaddi	–
INF – PRS PL1	-	šaddat
PRS SG2 – PRS SG3	la	-
PRS SG2 – INF	–	leat
PRS CONNEG		
Nouns		
PL NOM – SG GEN	dile/mánno	–
PL INE – SG COM	gielajn	gielain
SG ELA –	girkus	girkkos
SG ACC PXSG3		

Table 2: Homonymies comparison between Lule Sámi and North Sámi

Texts written by second language learners or students generally have more and other types of errors and more complex errors, which will require a different grammar checker.

Typical errors of high proficiency writers happen when the written norm deviates from the spoken dialectal variation. One example for that is the negation paradigm, which in some dialects resembles the North Sámi paradigm rather than the norm of written Lule Sámi.

In the Lule Sámi written norm, the negation verb is inflected for both person, number and tense (present and past) followed by the main verb in connegative form, which is always the same, whilst in North Sámi only person and number is marked on the negation verb. Tense is marked on the main verb with two different connegative forms, see Table 3.

Lule Sámi		North Sámi	
Present	**Past**	**Present**	**Past**
iv vuolge	ittjiv vuolge	in vuolgge	in vuolgán
i vuolge	ittji vuolge	it vuolgge	it vuolgán
ij vuolge	ittjij vuolge	ii vuolgge	ii vuolgán

Table 3: Negation comparison for ‘not leave’

There is no full consensus on the exact border between North Sámi and Lule Sámi (Ylikoski, 2016), so in Lule Sámi text one can find variation regarding negation that reflects dialectical variation. In Lule Sámi text both the North Sámi negation system, as ex. (2), and a system with ‘double’ past marking on both the negation verb and with the main verb (3) are used.

(2) Aktak **ij** **vuolggám**
someone not.NEG.PRES.3SG go.PASTP
nuorráj dan biejve.
sea.SG.ILL that day
'No one went on the sea that day.'

(3) Gå ålgus vuolggi, de
when outside go.PAST.2SG, then
ittji **vuojnnám** åvvå majdik.
not.NEG.PAST.2SG see.PASTP all nothing
'When you went outside, you didn't see anything at all'

Most of the systematic morpho-syntactic errors made by high proficiency writers reflect ongoing languages changes and might not even be corrected by a proofreader. A grammar checker is a good way of making people aware of such changes.

Soajttet is a modal verb meaning '(to) maybe' and usually stands with the infinitive form of the main verb. However, the present singular third-person form *soajttá* '(s/he) maybe' is by many writers being used as an adverb, not as a modal verb, as example (4) shows. The modal auxiliary is not followed by an infinitive as it should, but a finite verb in third-person singular.

(4) EU **soajttá** máhtti mijáv
EU may.PRES.3SG can.PRES.3PL us
viehkedit.
help.INF
'EU might be able to help us'

Within noun phrases, writers frequently make agreement errors. According to the norm the noun should be in singular with numerals and demonstratives agreeing in case and number, according to (Ylikoski, 2022) there is variation in the contemporary language indicating that this agreement system is changing. The errors in the Divvun gold corpus show us that the change has gone further than described in (Ylikoski, 2022), and numerals are handled in the same way as attributive adjectives, see Table 4. Some writers seem to make use of this "new" paradigm, as in ex. (5), while others seem to be somewhere in between, as ex. (6) shows. In this last example, the case of the numeral is correct, but the noun is in plural.

(5) Alvos Státtáv máhtá vuojnnet gåjt
colossal Stáddá can see at.least
gietjav **báhppagieldajs**.
seven.NUM.NOM.SG parish.PL.ELA
'You can see the colossal Stáddá from at least seven parishes'

(6) Suohkana juogeduvvin
municipality divide
gietja sáme
seven.NUM.ILL.ATTR outskirt.area.PL.ILL.
rabdaguovlojda.

'The municipalities got divided into seven outskirt areas.'

	'(these) two cows'	
	Norm	**Systematical errors**
Nom	(dá) guokta gusá	(dá) guokta gusá
Gen	(dán) guovte gusá	(dáj) guokta gusáj
Acc	(dá) guokta gusá	(dájt) guokta gusájt
Ine	(dán) guovten gusán	(dájn) guokta gusájn
Ill	(dán) guovte gussaj	(dájda) guokta gusájda
Ela	(dát) guovtet gusás	(dájs) guokta gusájs
Com	(dájna) guovtijn gusájn	(dáj) guokta gusáj

Table 4: NP with demonstrative pronouns and numerals

Another noun phrase internal error is the use of and adjective in predicative form in an attributive position, as example (7). This is not a very common error, but might be more frequent in texts written by second language learners, since the predicative form is the one in dictionaries and the adjective inflection system is one of the most complex area of the morphology (Ylikoski, 2022). Along with this rule, we also made rules for correcting errors where the attributive form of an adjective is used in a predicative position.

(7) Mij tjuovojma **roaŋkok** bálggáv.
We follow crooked.SG.NOM path.SG.ACC
'We followed a crooked path'

There are also agreement errors where relative pronouns fail to agree with their anaphora in number, as in ex. (8), and not agreeing with its anaphora in animacy, as in ex. (9). A similar error regards the agreement of reflexive pronouns with their anaphora in number.

(8) Da sáme **gænna** ietjanisá
Those s.PL.NOM who.SG.INE themselves
ællim muorravuovdde
have.not wood.forrest
'Those s without their own wood forrest'

(9) Åhtsåp jådediddjev **mij**:
Search leader.SG.ACC which.NHUM.SG.NOM
'We are looking for a leader who:'

Conditional mood is according to (Ylikoski, 2022) largely missing in Lule Sámi, and instead a periphrastic conditional consisting of the auxil-

iary lulu- 'would' and the infinitive is used. The conditional auxiliary *lulu-* is by some writers handled as if it is a separate verb with present and past tense, not a mood, making errors like (10) and the non-word error (11).

(10) Vuorasulmutja **lulu** huvsov
old.people.PL.NOM be.COND.2SG care
ja sujtov oadtjot.
and nursing get.IF
'Old people would get care and nursing '

(11) ...sávvá ienebu **lulujin** kursajda
...wish more *would.3PL course
oassálasstet.
attend.
'...wishes more people would attend courses.'

Another big group of errors are real-word errors. These are mostly based on phonetic similarity between the confused forms. In this work, we focused on general rules that are not limited to one single word, but rather forms that apply to a group of lemmata. In Table 5 the first error (*álgge-álkke*) is an error limited only to this specific word. When in a hurry of building resources for very low resource languages, one has to make sure to work in an efficient way, and writing rules for correcting specific words does not get us fast-forward. The rest of errors in Table 5 are errors being corrected by rules that generalise over groups of words, or for the frequent negation auxiliary (function words are more efficient).

The errors we have worked with in Table 5 are all real word errors with the ij-sound written 'i', or the other way around, with 'i' written 'ij'. We classified them as real word errors, even though some errors can also be seen as agreement errors. High proficiency writers are typically not insecure about agreement, but errors of this type can still happen when typing fast. Another complicating factor is that the -i sound can also be written -ij. Odd syllable nouns in illative case end in -ij, even though the pronouncation is not -ij. 'To the dog' is spelled *bednagij* even though the actual pronounced more like *bednagi*. However, the spelling error *bednagi* will be picked up by the spell checker since it is a non-word.

Both Lule Sámi and North Sámi verbs are inflected with three persons and three numbers in past and present tense. The subject verb agreement rules were copied from North Sámi to the Lule Sámi grammar checker.

4 Evaluation

The first version of the Lule Sámi grammar checker has 64 rules and 17 rule types, three of which have a regression test of 50 or more test sentences. We also ran an initial evaluation of each regression test, and plan to run the grammar checker on the error-marked up corpus of 32,202 words[4].

Figure 6 shows an evaluation of three error types with a sufficiently large regression tests. The other error types will be evaluated in the final version of the paper. The rules for relative pronoun and numeral/determiner agreement and for modal verb maybe-constructions give good results for both precision and recall. Precision and recall of the modal verb constructions are as good as 98%. We are aware that this still needs to be tested on an independent corpus. The quality is measured using basic precision, recall and f_1 scores, such that recall $R = \frac{t_p}{t_p+f_n}$, precision $P = \frac{t_p}{t_p+f_p}$ and f_1 score as harmonic mean of the two: $F_1 = 2\frac{P \times R}{P+R}$, where t_p is a count of true positives, f_p false positives, t_n true negatives and f_n false negatives.

We also ran a test run of the automatic evaluation on the marked-up gold corpus of Lule Sámi, to see if the grammar checker finds true errors and also to improve the error mark-up of grammatical errors in the corpus, keeping in mind that the corpus had been originally marked up for predominantly spelling errors.

A lot of errors found by the grammar checker are true positives. Many of them were either not marked up or - more frequently - marked up with a different scope. Since the start of marking up the corpora for spelling errors, the mark-up guidelines have been developed further in connection with *GramDivvun*, the North Sámi grammar checker, and adapted to automatic evaluation, where the grammar checker output is tested against the corpus mark-up.

There are examples of when the grammar checker actually found grammatical errors that the human proof-reader missed out. Thirdly, there are examples where the original marking is not consistent with the newer guidelines for how much the scope of the error should be with regard to how much the grammar checker actually marks up. Example (12) is one of the cases where an error in relative pronoun agreement has been identified correctly by the grammar checker. This error type

[4]Can be found on GitHub: https://github.com/giellalt/lang-smj/tools/grammarcheckers

Error	Correct form	Type of error
álgge 'beginner'	**álkke** 'easy'	Only for this single word
hábbmima NOMACT SG GEN *'the designing's'*	**hábbmijma** PRT PL1 'we designed'	Systematic for all contracted -it verbs
bælosti PRS PL3 'they defend'	**bælostij** PRT SG3 's/he defended'	Systematic for all odd syllable -it verbs and auxiliary/copula *liehket*
i/ittji PRS/PRT SG2 'you do/did not'	**ij/ittjij** PRS/PRT SG3 's/he do/did not'	Missing "j" for negation verbs Sg2
ij/ittjij PRS/PRT SG3 's/he do/did not'	**i/ittji** PRS/PRT SG2 'you do/did not'	Extra "j" for negation verbs Sg3

Table 5: Real word errors comparison

	Precision	Recall	F_1
Rel pronoun agreement	81.43	83.82	82.61
Modal verb ('maybe')	98.00	98.00	98.00
Num/det agreement	74.14	67.19	70.49

Table 6: Performance of the grammar checker on three error types based on regression tests

had a particularly high number of true positives in our preliminary evaluation, showing that this is a frequent error type. Another very frequent true positive that has not been adapted to current mark-up standards regards numeral error types, as in (13). The old mark-up would have a bigger scope including context for the error, i.e. *daj gålmmå tiemáj birra>dan gålmå tiemá birra*. The current guidelines only mark up the form that is to be corrected, meaning *daj>dan*, *gålmmå>gålmå* and *tiemáj>tiemá* which are corrected in three steps and by three separate rules.

(12) Da ulmutja
Those people.PL.NOM
ma Hamsuna mielas li
which.NHUM.PL.NOM Hamsun mind is
buorre ulmutja Hamsun gåvvi buorak
good people Hamsun describe good
láhkáj.
way.
'Those people who, according to Hamsun, are good, he describes in a good manner'

(13) Tjállagin li artihkkala **daj**
Text is article these.DEM.PL.GEN
gålmmå **tiemáj** birra ma li
three.NUM.SG.NOM theme about which is
ássje majna Árran la barggam ...
topics with Árran is work ...
'In the text there are articles about these three themes, which are topics Árran has worked with'

However, there are also several false positives, as in ex. (14), where *gålmmå* is not an error. The difficulty here is that the subsequent noun form is homonymous between nominative and genitive, and the numeral should have only been corrected if it was a genitive phrase. False positives occurred specifically for this error type (in the case of nominative/genitive nouns), showing that more work with the respective rules is necessary to improve the performance of the grammar checker.

(14) Ja gå Knut lij **gålmmå**
And when Knut was three.SG.NOM
jage vuoras de jåhtin Hábmelij,
year.SG.GEN old then move Hábmel,
sadjáj Hamsund.
place Hamsund
'And when Knut was three years old, they moved to Hábmel, to a place called Hamsund'

In ex. (15), on the other hand, the agreement error finding of the grammar checker in *álgij* 's/he started' and its correction to *álggin* 'they started' is a false positive. This is based on there being two subject candidates, because of singular nominative and plural genitive being homonyms, (*cuhppa*) and the other one plural (*biejve*, which in this sentence is singular genitive). The grammar checker confuses the first of them for a subject and therefore wrongly adapts the verb to it.

(15) Bierjjedagá sjnjilltjamáno 20. biejve
Friday March 20. day.SG.GEN
álgij cuhppa, ja hiejtij
begin.PAST.SG3 cup.SG.NOM, and end
lávvodak iehkeda.
saturday evening
'The cup started Friday on March 20 and ended Saturday evening'

Additionally, we tested the grammar checker on

a manually proofread Lule Sámi corpus used for a new text to speech (TTS) tool. The grammar checker did find errors that the proofreader had missed and was therefore useful in a project where we want the text to be perfect. Most of the responses from the grammar checker on this corpus were however false positives, with the grammar checker marking correct forms as errors. These 'bad' results were in turn used to improve and fine tune the grammar checker rules. We find this a very beneficial way of working - using our tools to double-check a proofread corpus, and at the same time using the results of the corpus to improve our tools.

When running the grammar checker on a university level thesis, the grammar checker found many real errors. It was interesting that some highly frequent repeated errors were due to changes in the language norm.

The overall results show us that the grammar checker actually finds real errors, but the main challenge with making it usable to users is to restrict the rules. At this point there is too much noise with more false positives than true positives.

5 Conclusion

We have shown that by using a related language grammar checker as a starting point, we were able to create a basic level grammar checker for Lule Sámi, categorise a fair amount of frequent error types and collect regression tests for each of them in a reasonable amount of time (120 hours between two linguists, one of them a native speaker). The importance for language revitalisation cannot be measured before integrating the tools in the respective text processing programs for the language community to use. But we know from experience with the spell checker, that the tools have a wide group of users, and their importance can usually be felt in the number of complaints that are sent when something is wrong with the distribution or other technical issues. In the future, we want to offer a high-performance tool for the most common error types to the Lule Sámi users. We aim to release a beta version together with the commonly distributed spellchecker in 2022.[5] From the developer side we aim at regression tests of at least 100 examples per error type with at least 90 % precision and 70 % recall, so that the tool will be useful for a wider language community, be used in schools, by the government and for private users on mobile phones.

[5] c.f. https://divvun.no/en/index.html

Acknowledgments

We want to thank Børre Gaup for running the evaluation on the gold corpus and helping with the technical side of error mark-up and automatic evaluation.

References

Lene Antonsen, Linda Wiechetek, and Trond Trosterud. 2010. Reusing grammatical resources for new languages. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 2782–2789, Stroudsburg. The Association for Computational Linguistics.

Kenneth R Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI publications.

Eckhard Bick and Tino Didriksen. 2015. CG-3 – beyond classical Constraint Grammar. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NoDaLiDa 2015)*, pages 31–39. Linköping University Electronic Press, Linköpings universitet.

Tino Didriksen. 2010. *Constraint Grammar Manual: 3rd version of the CG formalism variant*. GrammarSoft ApS, Denmark.

Fred Karlsson. 1990a. Constraint Grammar as a Framework for Parsing Running Text. In *Proceedings of the 13th Conference on Computational Linguistics (COLING 1990)*, volume 3, pages 168–173, Helsinki, Finland. Association for Computational Linguistics.

Fred Karlsson. 1990b. Constraint grammar as a framework for parsing unrestricted text. In *Proceedings of the 13th International Conference of Computational Linguistics*, volume 3, pages 168–173, Helsinki.

Fred Karlsson, Atro Voutilainen, Juha Heikkilä, and Arto Anttila. 1995. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin.

Susanna Angéus Kuoljok. 2002. Julevsámegiella. *Bårjås: Julevsámegiella uddni - ja idet?*, pages 10–18.

Krister Lindén, Erik Axelson, Senka Drobac, Sam Hardwick, Juha Kuokkala, Jyrki Niemi, Tommi A Pirinen, and Miikka Silfverberg. 2013. Hfst—a system for creating nlp tools. In *International workshop on systems and frameworks for computational morphology*, pages 53–71. Springer.

Ole Henrik Magga. 1994. Hvordan den nyeste nordsamiske rettskrivingen ble til. *Festskrift til Ørnulv Vorren*, pages 269–282.

Christopher Moseley. 2010. *Atlas of the World's Languages in Danger*, volume 3. UNESCO.

Sjur N. Moshagen, Tommi A. Pirinen, and Trond Trosterud. 2013. Building an open-source development infrastructure for language technology projects. In *NODALIDA*.

Tommi A. Pirinen and Krister Lindén. 2014. State-of-the-art in weighted finite-state spell-checking. In *Proceedings of the 15th International Conference on Computational Linguistics and Intelligent Text Processing - Volume 8404*, CICLing 2014, pages 519–532, Berlin, Heidelberg. Springer-Verlag.

Håkan Rydving. 2013. *Words and varieties : lexical variation in Saami*. Société Finno-Ougrienne.

Pekka Sammallahti. 1998. *The Saami Languages: an introduction*. Davvi girji.

Mikael Svonni. 2008. Språksituationen för samerna i sverige. *Samiskan i Sverige, rapport från språkkampanjerådet*, pages 22–35.

Linda Wiechetek, Sjur Nørstebø Moshagen, Børre Gaup, and Thomas Omma. 2019. Many shades of grammar checking – launching a constraint grammar tool for North Sámi. In *Proceedings of the NoDaLiDa 2019 Workshop on Constraint Grammar - Methods, Tools and Applications*, NEALT Proceedings Series 33:8, pages 35–44.

Linda Wiechetek, Flammie A Pirinen, Børre Gaup, and Thomas Omma. 2021. No more fumbling in the dark - quality assurance of high-level NLP tools in a multi-lingual infrastructure. In *Proceedings of the Seventh International Workshop on Computational Linguistics of Uralic Languages*, pages 47–56, Syktyvkar, Russia (Online). Association for Computational Linguistics.

Jussi Ylikoski. 2016. Future time reference in lule saami, with some remarks on finnish. *Journal of Estonian and Finno-Ugric Linguistics*, 7(2):209–244.

Jussi Ylikoski. 2022. Lule saami. *The Oxford Guide to the Uralic Languages*, pages 130–146.