Supporting Language Users - Releasing a Full-fledged Lule Sámi Grammar Checker

Anonymous Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Anonymouser Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

Releasing a Lule Sámi grammar checker has direct consequences for language revitalization. Our rule-based L1 grammar checker is a tool with the primary intention to support language users in their writing and their confidence to use the language. We release a version of the tool that corrects seven error types, including copula forms, lexical errors, different types of agreement errors of reflexive pronouns, relative pronouns and subject-verb, adjective form confusions, and interference errors from the majority languages. The selection of these error types is based on frequency of the errors and success of our tool, keeping in mind the usefulness for our users. Phonetically and syntactically copula errors turn out to be the most frequent error type in our evaluation corpus and are also most successfully corrected by our tool with a precision of 96%.

1 Introduction

We release a new grammar tool for Lule Sámi with the purpose of giving language users the security that their language is right in the absence of a strict norm - a paradox we face in our daily work. Speakers and writers of a language are confident and carefree when they feel secure in their language use. However, minority languages often face loss of language arenas and at the same time have less resources for language teaching than majority languages. The consequence is that (new) language users get insecure in their use of language and are often left to criticism by the language experts when speaking. This can lead to frustration and resistance to use the language among the ones that are not considered language experts. The notion of the language barrier - where older generations take the role of the language police - has also been reported in other minority language contexts.

The ones that know the language have a clear feeling of how the language should be even if there is not a written norm So there is a big breach between these experts and the ones that learn the language now. At the same time there are little contexts/opportunities to “improve” ones grammar skills and avoid being critcized so that speaking can lead to frustrations. Especially in writing, Lule Sámi text production differs from their coexisting majority language text production. Even official texts and texts written by highly proficient users contain a lot of spelling and grammar errors (Wiechetek et al., 2022). This is due to lesser written language proficiency in minority languages, and also unclear written norms.

A written norm and someone enforcing this norm is necessary to teach language competence to the younger generation and pass on expert language knowledge in all its richness. In the absence of enough L1 teachers, now many L2 speakers become teachers that need support to teach the language in all its details. There are no books that explain the language norm in all its details, including contrastive examples and frequent mistakes. Existing grammar books do not exceed correct text book sentences with a focus on morphology, rather than syntax. To get sufficient feedback one ones one language production, we need either constant human input or a tool that can evaluate our language on the fly and give feedback about its correctness.

The first Lule Sámi grammar checker for L1 users is ready to be released to the public in May 2023[1]. Building a Lule Sámi grammar checker started in October 2020 with a general error categorization and smaller experiments with rules. In

[1] divvun.no

2022 we did intensive work to collect regression tests and reported first results (Mikkelsen et al., 2022). NLP tools are made by linguists and software developers and they are evaluated by them as well. However the main motivation for these tools are the language users and the usability of the tools for them. That means that we want to make the tools available at an early stage, even if they do not include all the functionalities yet, and at the same time ensure their quality (i.e. especially good precision). Ensuring the quality means that only those error types that give a certain precision are included. The tools are meant to support teachers, proof readers and individuals by finding errors that are hard to detect because of orthographic similarities. They are also meant to help enforcing the (mostly orthographic) language norm proposed by the normative organ Giellagálldo[2] in a consistent way.

2 Language situation for Lule Sámi

Lule Sámi is an indigenous language spoken in Northern Norway and Sweden. The language is classified as a severely endangered language by UNESCO and has an estimated 800-3,000 speakers (Sammallahti, 1998; Kuoljok, 2002; Svonni, 2008; Rydving, 2013; Moseley, 2010). Lule Sámi is a morphologically complex language, for more details see Ylikoski (2022).

The current written form of Lule Sámi was approved in 1983, and the first spell checker for the language was launched in 2007. Lule Sámi lacks a long written tradition. According to Kuoljok (1997) most of the speakers can barely read and even fewer write. This situation has changed since 1997. In the education system, Lule Sámi is taught and used as the language of instruction. In Norway, Lule Sámi was for the first time taught as first language in primary school in 1992, and in 2012 it was for the first time possible to take a bachelor's degree in Lule Sámi at Nord University. Lule Sámi is also to a greater extend used in public administration, in 2000 Jåhkåmåhkke/Jokkmokk municipality became a part of the Sámi language administration municipalities in Sweden, and i 2006 the municipality Divtasvuona/Tysfjord became a part of the Sámi language administration municipalities i Norway. This development means that Lule Sámi is also used in writing to a greater extent than before. However, the written tradition is not very established, and the elderly heritage speakers master the written language only to a smaller extent.

Even thou language speakers are getting education in Lule Sámi language they seem to struggle when writing the language. In 2013, the Lule Sámi gold corpus of writing errors was created to test the spell checker's effectiveness. It then consisted of 29,527 words, written by native Lule Sámi speakers, with 2,827 marked writing errors. Of these errors, 1,505 were non-word errors identified by the spell checker, while the remaining 1,322 errors are morpho-syntactic, syntactic and lexical errors that only a grammar checker can detect and correct (Wiechetek et al., 2022). The goldcorpus shows a ***severe amount*** of errors in written texts.

To fully master a written language one must read a lot (Trosterud, 2021), minority language users therefor have a greater need for help in the writing process, since they don't experience their language in written form as much as majority language speakers. With Lule Sámi classified as a severely endangered language by UNESCO, it is important to increase the use of Lule Sámi to vitalize the language. A grammar checker for Lule Sámi would make it easier for people to write in the language, thus increasing its written use.

To develop a functional Lule Sámi grammar checker, we opted to focus on errors made by proficient writers instead of language learners. This approach allows us to create a checker that can handle texts with very few errors and gradually introduce more complex errors. A grammar checker for texts written by second language learners or students would require a different approach as they tend to have more and different types of errors, including more complex errors.

Errors made by high proficiency writers often arise when the written norm deviates from the spoken dialectal variation or the errors might express an ongoing language change.

3 Technical background

All tools described here are part of a multilingual infrastructure for 130 languages (Moshagen et al., 2013).[3]

The technological implementation of our grammar checker is based on finite-state automata for morphological analysis (Beesley and Karttunen, 2003; Lindén et al., 2013) and constraint gram-

[2] http://www.giella.org

[3] *https://github.com/giellaltGiellaLT*

mar (Karlsson, 1990b; Didriksen, 2010) for syntactic and semantic as well as other sentence-level processing. The Lule Sámi has a morphological analyser and lexicon that are both publicly available[4]. The morphological analyser was originally imported from North Sámi with all rules and set specifications and then adapted to Lule Sámi.

The grammar checker is a system consisting of 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.

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 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).

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.

Each ADD-rule is accompanied by a COPY-rule that exchanges relevant morphological tags in order to produce the correct sequence for the FST morphological generator to generate the correct form. In this case Sg3 is exchanged for Sg2. At the same time, we add a tag, *&SUGGEST* to mark that this is not the erroneous form anymore, but the correction.

(1) Dån **ittjij** boade guossáj.
you.SG2.NOM NEG.PAST.SG3 come guest.ILL
'You didn't visit.'

```
ADD (&real-negSg3-negSg2) TARGET ("ij")
IF (0 (Sg3))
(*-1 (Pron Nom Sg2)
BARRIER S-BOUNDARY OR
CS OR CC OR VFIN) ;

COPY (Sg2 &SUGGEST) EXCEPT (Sg3)
TARGET  (&real-NegSg3-NegSg2) ;
```

4 Lule Sámi Grammar checker

4.1 Testset

Having a set of example sentences that show the natural context for a grammatical error is essential for the construction of a grammar checker. We want to correct errors that are actually made by users of the language.

We have collected an error corpus of representative errors in *Yaml*-formatted[5] files specific to each error type. (Wiechetek et al., 2021) Typically, each regression file contains several hundred sentences. Our standard has been to have yaml tests of at least 50 test sentences. There should be a balance of correct and erroneous sentences covering the same phenomena so that one can test for false positives and false negatives. Test sentences should cover a variety of syntactic contexts and pay attention to long-distance relationships between syntactic functions. The collected errors are designed to cover a maximally large amount of real-world errors that people make when writing texts, in order to keep the grammar checker usable for people. The file naming is now error-specific,[6] but as they come from an authentic corpus, they can contain multiple errors per sentence including other types of errors and nested errors.

At first, we did write test sentences for yaml test ourself and also searched SIKOR manually for sentences with similar errors. After having written rules, we automatically harvested test sentences that get error tags from the developer-corpus [7], and used these to improve the rules.

Yaml is a mark-up language with a simple syntax that makes writings of the tests convenient and

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

[5] https://yaml.org/spec/1.2/spec.html

[6] https://github.com/giellalt/lang-smj/tree/main/tools/grammarcheckers/tests

[7] https://giellalt.github.io/proof/gramcheck/extracting-precision-sentences.html

co-operation with programmers and linguists easier. We chose to use the Yaml format for grammar testing because of positive experiences with the use of the same format for spell checker testing.[8] The original test framework for morphology testing initiated by Brendan Molloy can be found on GitHub.[9]

4.2 Grammar for error correction

It is challenging to write a prescriptive grammar checker for a language without a long and clear written norm like Lule Sámi. Even written grammar books do not cover all the phenomena of Lule Sámi language. For most languages, a written norm is far away from oral language. Oral Lule Sámi contains a lot of dialectal variations and is subject to ongoing language change. As all speakers of Lule Sámi are bilingual, oral language can include interference and loans from the majority languages, which is not desired in a written norm. For all these reasons it is a challenge to build a grammar checker that corrects this language. We face the question of where to put the boundaries between written and oral Lule Sámi. The decision can have serious consequences since Lule Sámi is an endangered language under revitalisation, and the grammar checker can have a standardising effect on the language of the younger generations. It is positive that speakers receive feedback when they write language that is clearly influenced by Norwegian or Swedish, but at the same time the grammar checker can also thought to give feedback leading to a limitation of dialectal variation.

We do not have the authority to determine the norm, but with the release of the grammar checker, we might have the strongest influence regarding the sentence level norm in the entire Lule Sámi language community. One cannot wait until normative matters are solved before developing tools needed by the language community, the path must be created as we walk. The grammar checker will be further developed and improved after this first version release. Hopefully the releasing of the Lule Sámi grammar checker will facilitate discussions around the norm and discussion around the choices made by us. Upon the release of the grammar checker, we will have presentations for the language community where we inform about the choices regarding the grammar checker and also discuss further development.

We have written 18 rule types, and from the evaluation seven of these are ready to be released.

The words "oahpásmuvvat" and "oahpástuvvat" both meaning "to get to know" are often confused, that decices which one is used is the animacy of what one is *getting to know*. The verb *oahpásmuvvat*, e. (2) is used in inanimate contexts and requires illative case, whilst *oahpástuvvat*, ex. (3) is used in animate context and require comitative case. The rules of the grammar checker corrects both verb according to animacy and the case of the referent.

(2) Oahpásmuváv bijllaj.
get.to.know.PRES.1SG car.SG.ILL
'I get to know the car.'

(3) Oahpástuváv sujna.
get.to.know.PRES.1SG PRON.2SG.COM
'I get to know her/him.'

The modal verb *soajttet* meaning 'maybe' should be paired with the infinitive form of the main verb. However, many writers are using the present singular third-person form *soajttá* as an adverb rather than a modal verb, as shown in ex. (4). In this example the modal auxiliary is not followed by an infinitive as expected, but rather by a finite verb in the first-person singular form. The rules of the grammar checker will replace *soajttá* with the adverb *ihkap*.

(4) ***Soajttá** *tjálláv nágin
maybe.PRES.3SG write.PRES.1SG some
bágojt
word.SG.ACC
'Maybe i will write some words'

For agreement the grammar checker corrects relative pronouns in comitative case, as the incorrect ex. (5), and the reflexive pronouns *iesj* in nominative, as the incorrect ex. (6), when these are not agreeing with their anaphora in number. The grammar checker also corrects agreement errors between subject and verb, this is a quite common error done since indicative verbs are inflected for three numbers and three persons.

(5) Álu l má ålmmåjn ***gænna** l
often is PCLE man.PL.INE who.SG.INE have
fábmo
power

[8] https://giellalt.uit.no/infra/infraremake/AddingMorphologicalTestData.html\#Yaml+tests

[9] https://github.com/apertium/apertium-tgl-ceb/blob/master/dev/verbs/HfstTester.py

‘Often it is men who have power’

(6) Mij hæhttup ***iesj**
we.NOM must.PRES.1PL self.REFL.SG.NOM
jáhkket
believe.
‘We ourselves must believe.’

Another noun phrase internal error corrected by the grammar checker is the use of and attributive adjective in predicative position, as the incorrect ex.(7).

(7) Ássje l ***gássjelis** munji.
matter.sg.nom. is difficult.ADJ.ATTR I.ILL
‘The matter is difficult for me.’

For the copula verb **liehket**, meaning ‘to be’, the grammar checker has rules following the system described in Spiik (1989). In a sentence initial position the copulas different forms form sentence internal forms, as shown for present tense in Table 1 Even if this system is explained in (Spiik, 1989), the sentence internal forms are widely used sentence initially in written texts, and the sentence initial 3.Sg forms in both present and past tense are much used in sentence internal position. The sentence internal present 3.Sg form also varies between “la” or “l”: “la” is used if the preceding word ends on a consonant, and “l” is used if the preceding word ends on a vowel. Even thou there most likely are and have been dialectal variation in regarding the copula system we have made rules according to Spiik (1989). We have fine-tuned the rules for choosing between “la” or “l” since it really is not at straight forward as Spiik explains it. As developers we are not sure of how well the copula rule will be received in the language community; The copula system of the grammar checker is not widely used in texts, for example have the translators of the Lule Sámi New Testament chosen a different approach to the copula “liehket”. However the grammar checker allows users to turn off and on rules they want to have checked, and if some speakers finds it annoying, they can turn the correction for this rule off.

5 Evaluation

For evaluation of our tool, we use a part of *SIKOR*, the Lule Sámi free corpus containing administrative, law, religious, non-fiction, fiction, and science texts. The Lule Sámi corpus SIKOR is divided into three parts: a marked-up goldcorpus for evaluation, an unmarked testing corpus and a development corpus for developing rules.

Morphological form	**Sentence internal**	**Sentence initial**
1Sg	lav	lev
2Sg	la	le
3Sg	la/l	le
1Du	lin	len
2Du	lihppe	læhppe
3Du	libá	læbá
1Pl	lip	lep
2Pl	lihpit	lehpit
3Pl	li	le

Table 1: Paradigm for liehket ‘to be’

SIKOR consists of a freely available corpus, *FREECORPUS* and a corpus that is restricted by copyright, *BOUNDCORPUS*.

The goldcorpus consists of texts from both *FREECORPUS* and *BOUNDCORPUS* and is marked-up for spelling and grammar errors. For simplicity, we will hence refer to these as *FREECORPUS* and *BOUNDCORPUS*. This work includes testing for inconsistencies and improvement of the manual grammar error mark-up the first time. Since the goldcorpus consists of text that has not been proof read there are a lot of grammatical errors. The goldcorpus and its markup is described in Wiechetek et al. (2022).

The testcorpus is not manually marked-up, but put aside for future evaluation and mark-up. As the goldcorpus is still fairly small, we want to make sure that there is enough material that has not been used for rule development to cover sufficient instances of all different types of grammatical errors. This is important keeping quality assurance for our users in mind.

The development corpus, on the other hand, is not marked-up, and being used to develop and improve the grammar checker on the fly.

The results are shown in Table 2. 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.

Table 2 shows that some error types have very few instances in the corpus we checked. Basing the quality of the error rules only on this test is

	Precision	**Recall**	**Positives**
Copula rules	96.13%	83.71%	117
Rel pronoun agreement	72.22%	81.25%	17
Animacy of rel pronouns	33.33%	25%	3
Subject verb agreement	77.42%	25.53%	31
Numeral agreement	60%	100%	10
Passive/Active			5

Table 2: Evaluation of the Lule Sámi grammar checker on *BOUNDCORPUS*

too risky. Therefore, we use regression test results in Table 3 as a second criterion. For the final version of this paper, we will include the second much bigger error marked-up corpus *FREECORPUS* for a more thorough evaluation after fine-tuning the existing mark-up. The work of this article has gone both ways, firstly, rule-development for automatic grammatical error detection, and secondly, improving grammatical error mark-up after running the grammar checker. This shows that manual error mark-up can be difficult and assisted by a grammar checker for consistency.

We also know that *FREECORPUS* includes fiction texts that have more instances of certain error types, e.g. errors for relative pronoun animacy.

Based on the results of both tables, and keeping the quality assurance for the users in mind, we will release functionalities for the following error types. Copula form, relative pronoun agreement, and subject verb agreement rules have a good precision and perform well in regression testing. All of them pass a threshold for precision of 70%.

In addition, we will release error correction for error types with few instances in BOUNDCORPUS based on good regression test results and knowledge about high frequency of the errors from experience as a manual proof reader. These error types are: adverbial use of the modal verb in third person singular, *soajttá* 'maybe s/he does'; use of attributive adjective forms instead of predicative forms; confusion of the two forms *oahpásmuvvat>oahpástuvvat*; and reflexive pronoun errors.

Altogether these are seven general error types that will be released with functionalities with the first version of the Lule Sámi grammar checker.

	PASS	**FAIL**
Lexical error (*oahpásmuvvat-oahpástuvvat*)	63	1
Number agreement of reflexive pronoun	60	7
Modal verb soajttá used as adverb	78	7
Adjective form (Attr>Pred)	164	5
Copula form	122	4
Number agreement comitative relative pronoun	105	16
Subject-verb agreement	91	19
Past tense negation	46	8
Animacy of rel pronouns	118	83
Copula agreement with subject li > lij (PrsPl3 > PrtSg3)	35	16
Copula agreement with subject lij > li (PrtSg3 > PrsPl3)	31	16
Negation agreement with subject i > ij (NegSg3 > NegSg2)	11	1
Negation agreement with subjec ij > ij (NegSg2 > NegSg3)	13	1
Nomen actionis¿Present Du1 or Pl1 (hábbmima>hábbmijma)	11	0
Adjective form (pred>attr)	48	26
Genitive before postposition	67	24
Number agreement nominative relative pronoun	118	89
Numeral agreement	141	110

Table 3: Regression test results of the Lule Sámi grammar checker (for comparison)

Even thou we seven rules are working fine according to evaluation of regression tests and the goldkorpus, there are still remaining complex issues with these rules.

In ex. (8) and (9), the sentences are more complex than what we tough off when writing rules. In ex. (8) the grammar checker erroneously changes the attributive adjective "buosjes" to predicative "buossje". In this example there are two attributive adjectives connected with the conjunction "ja" meaning "and". When writing rules for the grammar checker we have not thought about coordinate attributives.

(8) Adrian Nystø Mikkelsen gut aj la
Name Name Name who also is
buosjes ja vissjalis
tough.ADJ.ATTR and eager.ADJ.ATTR

bállotjiektje.
soccerplayer.SG.NOM.
‘ Adrian Nystø Mikkelsen who is a tough and eager soccer player.’

Another complex matter is when the pronoun is dropped and the grammar checker therefore gets the subject-verb agreement all wrong, as in ex. (9). The grammar checker erroneously corrects the verb into Pl3 since the Sg1 pronoun “mån” is dropped.

(9) Hådjånav gå **vuojnáv**
Get-upset.PRES.1SG when see.PRES.1SG
mijá galba biejsteduvvi.
our signs destroy.PASSIVE.PRES.3PL
‘ I get upset when I see our signs being destroyed.’

Some of the errors that the grammar checker makes are due to the combination of errors. In ex. (10), the grammar checker erroneously changes “ma” to *mij*. Therefore the subject is singular and the verb *guosski* is also corrected by the grammar checker. Here the grammar checker changes “ma” to singular which is a false positive because of a wrong referent, and then as a follow up false positives it also tries to change the verb “guosski” to singular to correct the agreement with the relative pronoun.

(10) Lav válljim
Have.PRES.1SG choose.pst.ptcp
teoritevstajt kompendijis **ma**
text.PL.ACC compendium that.PL.NOM
guosski
regard.PRES.3PL
álggoálmmukmetodologijav.
indigenous.methodology.ILL
‘I have chosen texts from the compendium that regard indigenous methodology.’

We also have similar examples where the erroneously correction by the grammar checker is due to a combination of errors, but where it is the writer who has done two different errors. In ex.(11) the grammar checker corrects the attributive adjective “váges” to singular “váhke”, but it should be corrected to plural “váge”. The writer has done two errors and written the verb “viertti” in present Sg3, when it actually should be present Pl3 “vierttiji”. The grammar checker misses this agreement error and therefor the adjective attribute form is corrected to predicative singular form. If we improve the grammar checker so that it does not miss the agreement error, it will succeed correcting the second error too.

(11) Moralla subttsasin de máhttá liehket
Moral story then might be
rádna ***viertti** liehket
friend.PL.NOM must.PRES.2SG be
***váges** nubbe nuppijn jus
honest.ADJ.ATTR each other if
rádnastallam galggá bissot.
friendship will remain.
‘The moral of the story might be that friends need to be honest with each other if the friendship is to remain.’

The same happens in ex. (12), where the writer has misspelled the indefinite pronoun *iehtjádijn*, and therefor the grammar checker erroneously corrects *oahpástuvvat* to *oahpásmuvvat*.

(12) –Ietja dahki majt hálidi,
Self do.PRES.3PL what want.PRES.3PL,
ja dan båttå máhtá buorebut
and that moment can.PRES.2SG better
*ietjadijn **oahpástuvvat**, javllá Inga Lill.
non.word get.to.know.INF, says Inga Lill
‘Everyone does what they want, and at the same time you can get to know someone better, says Inga Lill.’

Also there are examples where the rules of the grammar checker work fine, but where it erroneously corrects because of problems with disambiguateing homonymies. In ex. (13) the disambiguator believes “jage” to be nominative plural, when it actually is genitive singular. Because of the grammar checker believing “jage” to be the subject of the sentence it corrects sentence initial present form “le” Pl3 form “li” instead of the correct Sg3 form “la”.

(13) Badjel guoktalåk jage
Over twenty years
***le**
is.PRES.SG3.SENT.INIT
duodje munji årrum
Sámi.handcraft.SG.NOM me be
vájmoássjen ja oasse iehtjam identitehtas.
heart.case and part my identity.
‘For over twenty years Sámi handcraft has been close to my heart and a part of my identity.’

The evaluation shows that even though the grammar checker works well with seven rules, there are still complex issues that cause the grammar checker to fail even for these types errors. More errors in the same sentence make it harder for the grammar checker. It is therefore important upon to point out to the users that the grammar checker is meant predominantly for L1 users does not work very well with second learners texts (yet) when releasing it. The evaluation shows that building a grammar checker for L1 users before L2 users is a good way to go, as the tool performs better with only one error in the sentence, and high proficiency writers are assumed to make less errors.

6 Conclusion and future plans

We have developed a tool for grammatical detection and correction of Lule Sámi that is ready to be released and support the Lule Sámi language community in writing. The evaluation of the grammar checker shows that seven error types are ready to be released. These are corrections regarding copula forms, lexical confusion of *oahpásmuvvat-oahpástuvvat*, number agreement for reflexive pronouns, the use of the modal verb *soajttá* as an adverb, confusion of attributive and predicative adjective forms, comitative forms of relative pronouns, and lastly subject-verb agreement. While our evaluation corpus is still a bit too small to have a good representation of all errors, it was evident that especially copula errors are very frequent and also the other error types were represented. They also show the best precision with 96% and recall of 84%. In other error types we rely on our manual proof-reading experience to know about their frequency. This goes hand in hand with our wish to focus on user demands. In the future we will test on bigger corpora and enhance them with error mark-up. We also plan to improve precision and recall for the correction of existing error types by testing on more syntactic contexts. In addition we would like to include more error types, also for L2 users, based on our findings and feedback from the language community.

References

Kenneth R Beesley and Lauri Karttunen. 2003. Finite-state morphology: Xerox tools and techniques. *CSLI, Stanford.*

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. http://visl.sdu.dk/cg3/vislcg3.pdf (Accessed 2017-11-29) *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. 1997. *Nominalavledningar på ahka i lulesamiskan.* Acta Universitatis Upsaliensis.

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.

Inga Lill Sigga Mikkelsen, Linda Wiechetek, and Flammie A Pirinen. 2022. https://doi.org/10.18653/v1/2022.computel-1.19 Reusing a multi-lingual setup to bootstrap a grammar checker for a very low resource language without data. In *Proceedings of the Fifth Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 149–158, Dublin, Ireland. Association for Computational Linguistics.

Christopher Moseley. 2010. http://www.unesco.org/culture/en/endangeredlanguages/atlas *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.

Nils Eric Spiik. 1989. *Lulesamisk grammatik*. Sameskolstyrelsen.

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

Trond Trosterud. 2021. Utan tastatur, ingen tekst: om det språkteknologiske grunnlaget for språka våre. In Karin Kvarfordt Niia, editor, *Framgång för små språk.*, pages 68–73. Små språk i Norden.

Linda Wiechetek, Katri Hiovain-Asikainen, Inga Lill Sigga Mikkelsen, Sjur Moshagen, Flammie Pirinen, Trond Trosterud, and Børre Gaup. 2022. https://aclanthology.org/2022.lrec-1.125 Unmasking the myth of effortless big data - making an open source multi-lingual infrastructure and building language resources from scratch. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1167–1177, Marseille, France. European Language Resources Association.

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. https://aclanthology.org/2021.iwclul-1.6 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. 2022. Lule saami. *The Oxford Guide to the Uralic Languages*, pages 130–146.