How to Create Treebanks without Human Annotators – An Indigenous Language Grammar Checker for Treebank Construction

Linda Wiechetek **Flammie A Pirinen** **Maja Lisa Kappfjell**
UiT—Norgga árktalaš universitehta
Tromsø, Norway
first.last@uit.no first.last@uit.no first.last@uit.no

Abstract

Creating treebanks for low resource languages is an important task. However, low resource Indigenous language contexts have not only limited resources in terms of text data, but also limited human resources that are available for linguistic annotation. We suggest a work-around by applying a Constraint Grammar operated rule-based dependency parser to do the work of creating a marked-up treebank. However, due to a lot of noise, meaning spelling and grammatical errors in South Sámi written texts, this tool often fails to create complete and correct trees. As a fix to this, we created a grammar checking tool for the most common South Sámi grammatical error types, which improves the quality of the dependency parser significantly. As both literacy and normative standards for most Indigenous languages are much more recent than for majority languages, spelling and grammatical variation and errors are a common source of noise, and the application of a correction tool like ours can be useful in the construction of treebanks for these languages.

1 Introduction

In an extremely low resource language context, treebanks are an important link to developing high level tools that other languages consider standard. Machine-learning based language technology can utilise the treebanks for training and testing new models, and rule-based systems can use them as a gold standard to strive for. In addition they can be used for language comparative tasks, evaluation, etc. Low resource languages like South Sámi, however, are not only low resource in terms of data ($<$ 2 million words) but also lack human resources, which makes manual linguistic annotation of big text corpora impossible. For creating a South Sámi treebank, we therefore applied a Constraint Grammar based dependency annotation tool that can annotate unlimited amounts of text automatically using existing morphological and syntactic tools as their basis. When dealing with low resource Indigenous languages we need to keep in mind that language standards are often still in the process of being developed, and language contact with the majority language influences the way people use their language. South Sámi texts contain a lot of noise in each sentence in terms of typos and non-standard forms, code-switching and sentence structures that resemble literal translations from the majority language rather than using authentic South Sámi syntax. This type of noise is not comparable to the noise in a majority language corpus. It rather reflects the relatively large amount of L2 writers (second language users) in the South Sámi text corpus. As we want a treebank that can also be used for teaching purposes, we would like it to represent mostly L1 language.

Some of these errors and non-standard forms disrupt the sentential dependency structures and prevent our tool from working properly. Especially noun phrase internal errors, case errors and agreement errors lead to broken dependency trees. We therefore suggest the usage of a spelling and *grammar error correction* (GEC) tool as part of the pipeline to create a treebank. All our tools are part of a multi-lingual language resource platform (*GiellaLT*) which provides a common infrastructure for over 150 languages, most of them low-resource and/or Indigenous languages.[1] We manually marked-up error corpora, which we used to identify relevant and frequent errors and created a grammar checking tool that corrects these morpho-syntactic structures. The corrected sentences are then fed into the dependency tool, which create our treebank for South Sámi. South Sámi is an Indigenous language with about 500 speakers, and about 10 percent of these writes the language. This work has been made within a language technology group that started as an initiative of the Sámi Parliament

[1] https://giellalt.github.io and https://giellalt.github.io/LanguageModels.html

Proceedings of the 23rd International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2025), pages 119–128

20 years ago, which is why we combine both native language and engineering competence. Our main goal is to develop tools for and together with the language community, especially those that are needed in administration and education. This is self-determination in practice, which is also central principle in Sámi endeavors. South Sámi is a Uralic language with interesting syntactic features, such as copula drop, which leaves many sentences without a finite verb, an interesting matter for dependency parsing.

This work is a contribution to creating both proofing tools and a treebank for further research and tool creation. South Sámi did not previously have an annotated treebank, thus our contribution in this work is also that of a new treebank. Our goal was to create in the most efficient way given limited resources, also making sure that language presented therein is authentic but error free. The treebank follows the written standard that is backed by the South Sámi standardization body *Gïelegaaltije* creating a valuable annotated corpus resource. We will in the following present the grammar checking tool, and show how it is integrated into automatic treebank construction of South Sámi.

2 Background

2.1 Language background

South Sámi is an official language in altogether four municipalities in Norway and six municipalities in Sweden. There are approximately 300-600 South Sámi speakers. South Sámi is a morphologically complex language with similar grammatical structures as other Sámi languages. The Sámi languages belong to the Uralic language family, which is unrelated to the Indo-European languages. South Sámi has a number of features that clearly distinguish it from other Sámi languages. South Sámi has even stronger SOV word order than Lule Sámi, and both distinguish between elative and inessive case, which are replaced by locative case in North Sámi. South Sámi typically drops the copula in sentences without pro-drop. It also has nominative plural noun phrases in definite object position, which influences syntactic disambiguation. Negation is more complex than in North and Lule Sámi as South Sámi has a specific paradigm for past tense copula negation verbs that agree with the negation forms. The South Sámi written standard or according to the term of the time, The South Sámi textbook standard, was recommended by the Sámi Language Council in 1976 and was adopted in 1978. (Bergsland and Mattsson Magga, 1993) Some grammatical variants and paradigms have not yet been standardized explicitly by the standardization organ (*Gïelegaaltije*). However, there are written grammars that serve as a basis for teaching and for proofreading. A few grammatical matters are not described in grammars yet, and the grammatical authority lays with the native speaker elders. This knowledge remains to be formalized and presented in a way such that newer speakers that are less exposed to the language can receive the guiding they need to be confident speakers and writers.

Language contact with the Scandinavian majority languages Norwegian and Swedish are further leading to a lot of interference in South Sámi written text. These are clearly marked because they deviate significantly from both Sámi and Uralic morpho-syntax. A clear South Sámi standard is essential for the survival of the language. Without a clear standard new learners lack the confidence to use the language in speech and script and typically chose the safer alternative, the majority language. This means that language planning requires clear choices as regard orthography, lexicon, idioms and grammar to ensure a future for South Sámi and discontinue the colonialization process.

2.2 Technical background

The core pieces of this work are a rule-based dependency analyzer and a grammar checker module. The dependency analyzer is written for the three Sámi languages, North Sámi, Lule Sámi and South Sámi, which is based on a full morphological analysis that is followed by morpho-syntactic disambiguation and syntactic parsing. The syntactic parsing includes only function labels, but no explicit dependencies. Until this step the different Sámi languages have their separate language modules. The dependency structure, however, is added in a common module for all the languages, based on the flat syntactic function tags from the previous module. This work is thoroughly described in Antonsen et al. (2010). The automatic dependency annotation is created bottom up, so that even partial dependency trees can be created if some parts of the sentence contain errors or could not be fully disambiguated. Dependencies build on the same syntactic structure as the grammar checker. They use a specific rule format, which maps dependents to their parents and the other way around based on previously mapped morpho-syntactic labels and

```
SETPARENT:SetObjToRightMv OBJ> TO (*1
(<mv>) BARRIER S-BOUNDARY OR @-FSUBJ>)
;
```

Figure 1: Example rule mapping objects to their right handed verbal mothers

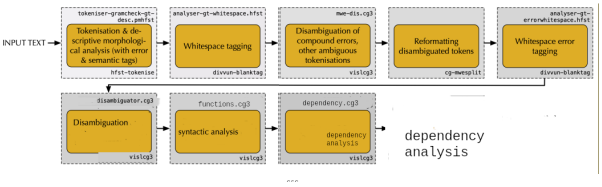


Figure 2: Modular structure of the dependency analysis

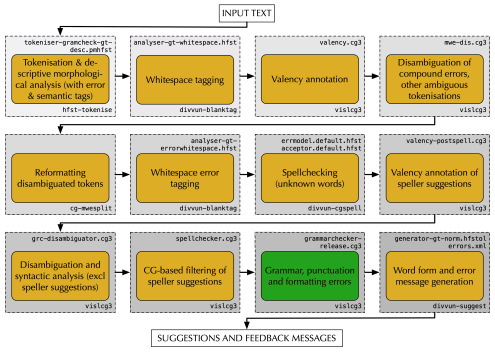


Figure 3: Modular structure of the grammar checkers

word order. The parsing of dependencies is based on rules of the type shown in Figure 1, for example where we map the object to a transitive main verb to its right.

The grammar checker module uses the same technology and a similar pipeline. It is specifically written for South Sámi, although some of the error types exist in North and Lule Sámi as well.

Our framework is based on rule-based natural language processing: finite-state morphology (Beesley and Karttunen, 2003) and constraint grammar (Karlsson, 1990). We use the free/ open source VISL CG 3 *constraint grammar* (CG) compiler (Bick and Didriksen, 2015). The linguistic analyses made by the systems include morphological, syntactic and semantic analyses, both on word-level as well as on a dependency graph level. The VISL CG 3 -based dependency analysis has been used in various applications including grammar checking, machine translation, semantic role annotation for various languages like Greenlandic, Danish, Spanish, Portuguese. (Bick, 2019; Rademaker et al., 2017; Bick, 2022)

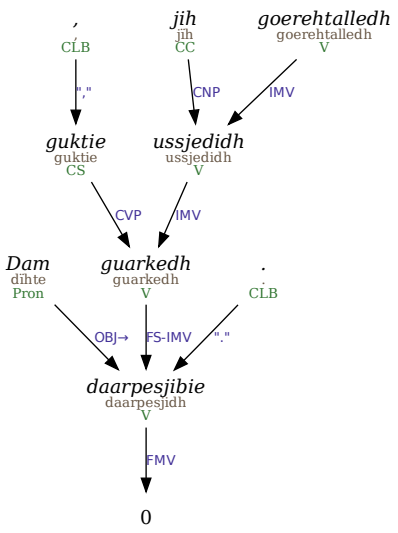
The VISL CG 3 dependency analysis' foremost goal is not to build a treebank with complete trees, but primarily create another linguistic layer that facilitates the above mentioned tasks when building applications for specific language communities. As trees are created bottom-up, which can leave them partly disconnected, they are not instantly convertible to even better known standards such as Universal Dependencies (UD) (De Marneffe et al., 2021). However, there are previous work that is based on conversion from our annotation system to UD, see for example (Sheyanova and Tyers, 2017; Antonsen et al., 2010) for a North Sámi UD treebank. Automatically generated treebanks need to be verified and fixed by human annotators skilled in the langauge, this is both by UD guidelines and of course makes a reasonable way to create goldstandards.

The system performing the grammar analysis and correction is built of modules, see Figure 3 for the structure of the grammar checker. The pipeline used for grammatical error corrections includes a syntactical analysis, and the overall system can be used for dependency-based syntactic analysis as well, with slightly different module structure than the one pictured for grammar checking and correction. (Wiechetek and Kappfjell, 2023)

All text data in this work is taken from Sámi international corpus SIKOR (SIKOR, 2025). It contains texts in Sámi languages including South Sámi.

3 A treebank for South Sámi

Our VISL CG 3 dependency analyzer for South Sámi (Wiechetek and Kappfjell, 2023) maps dependencies between word forms that have received a morphological analysis and a syntactical label. Each of these rules builds a partial tree, and combined with each other ideally a full tree is created. However, the tool is also able to construct partial trees, which is useful for atypical sentences, ellipses, headlines, in particular sentences without finite verbs. This is also relevant for South Sámi as copula-drop is a typical feature of the language. (Magga and Mattsson Magga, 2012) It also means that the tool can construct partial trees for sentences that contain spelling and grammatical errors or ommitted words. We ran the dependency parsing tool on 481 sentences and 7,266-token sample corpus to see how many complete trees it is able to construct. 188 of 481 sentences produce complete parse trees. One of these complete trees is displayed in Figure 4 showing the dependency structure of ex. (1). It includes a finite verb and three coordinated infinitives. The vislcg3 output of



Dam daarpesjibie guktie guarkedh , ussjedidh jih goerehtalledh .

Figure 4: Dependency tree for ex. (1)

the dependency analysis is displayed as graphical trees for the purpose of visualization. The original output can be seen in Figure 5, where dependency structures are expressed by absolute numbers after the hashtag for the position of each word pointing to the number of the word they are dependent on. In the case of the finite verb *daarpesjibie* its position in the sentence is *2* and it points to the root *0* (#2->0) It creates a full tree despite the orthographical error in *jih* (should be: *jïh*) as this the morphological analyzer accounts for some of the typical orthographical errors. The object *Dam* should be analyzed as dependent on the infinitive *guarkedh* 'understand' instead of *daarpesjibie* 'need'.

(1) Dam daarpesjibie guktie guarkedh,
that.ACC.SG need.PRS.1.PL for understand,
ussjedidh jih goerehtalledh.
think and investigate
'We need that to understand, think and investigate.'

The dependency tree for ex. (2-a) is also complete. However, the dependency structure in Figure 6 shows several errors. The adjective *veaksehke* and the demonstrative pronoun *gaajhkh* should be dependent on the noun *gielen* instead of the finite verb *leah*.

The reason for the partial errors in the dependency structure is one grammatical error in the ad-

```
"<Dam>"
    "dïhte" Pron Pers Sg3 Acc <W:0.0> @OBJ> #1->2
"<daarpesjibie>"
    "daarpesjidh" <mv> V TV Ind Prs Pl1 <W:0.0> @FMV #2->0
"<guktie>"
    "guktie" CS <W:0.0> @CVP #3->4
"<guarkedh>"
    "guarkedh" <mv> V TV Inf <W:0.0> @FS-IMV #4->2
"<,>"
    "," CLB <W:0.0> #5->3
"<ussjedidh>"
    "ussjedidh" <mv> V TV Inf <W:0.0> @IMV #6->4
"<jih>"
    "jïh" CC <W:0.0> @CNP #7->6
"<goerehtalledh>"
    "goerehtalledh" <mv> V TV Inf <W:0.0> @IMV #8->6
"<.>"
    "." CLB <W:0.0> #9->2
```

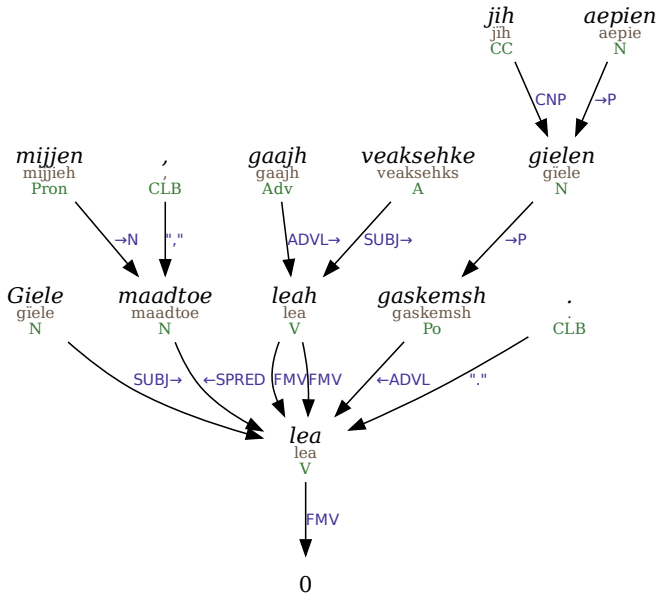
Figure 5: VISL CG3 dependency output

jective form *veaksehke* (correct: *veaksehks*) makes it appear a subject in nominative singular instead of an attribute to *gielen*. *Gaajhkh* can therefore not be identified as adverb dependent on the adjective. The morphological analyzer is robust enough to compensate for several spelling errors as the long 'i' in three words and misspelled *aepien* (correct: *aerpien*). They still receive a morphological and syntactical analysis.

(2) a. *Giele lea mijjen maadtoe,
language be.PRS.3.SG our foundation,
gaajh veaksehke
incredibly strong.NOM.SG
gielen jih aepien
language.GEN.SG and heritage.GEN.SG
gaskemsh leah.
between be.PRS.3.SG
'Language is our foundation, there is an incredibly strong connection between language and heritage.'

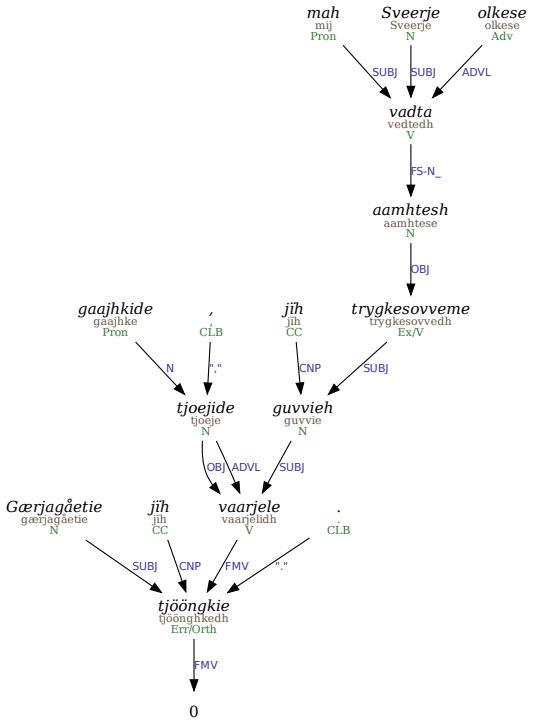
b. Gïele lea mijjen maadtoe, gaajh veaksehks gïelen jïh aerpien gaskemsh leah.

Spelling errors and grammtical non-standard forms are overdimensionally represented in South Sámi written texts. For most majority languages, spelling errors and non-standard forms are filtered out by some kind of proofreading. In addition, writers of majority languages have typically undergone a lot of training and their writing has undergone a lot of proofreading in their respective languages school systems. Figure 7 of a complex sentence including coordinated demonstrative phrases with a relative clause displays a number of these typical errors in South Sámi. Ex. (3-a) shows all errors with their correction in ex. (3-b).



Giele lea mijjen maadtoe , gaajh veaksehke gielen jih aepien gaskemsh leah .

Figure 6: Dependency tree of ex. (2-a)



Gærjagåetie tjööngkie jïh vaarjele gaajhkide tjoejide , guvvieh jïh trygkesovveme aamhtesh mah Sveerje olkese vadta .

Figure 7: Dependency analysis for ex. (3-a)

Morphosntactic errors	334
Syntactic errors	259
Real-word errors	147
Lexical errors	216
Non-word spelling	3,263

Table 1: Error statistics in error annotated text data

(3) a. Gærjagåetie tjöömgkie jïh
library collect.PRS.3.SG and
vaarjele gaajhkide
take.care.PRS.3.SG all.ACC.PL
tjoejide, guvvieh jïh
sound.NOM.PL, picture.NOM.PL and
trygkesovveme aamhtesh
printed item.NOM.PL
mah Sveerje olkese
which.NOM.PL Sweden out
vadta.
give.PRS.SG.3
'The library collects and takes care of all sound, images and printed items which Sweden has published'

b. Gærjagåetie tjöömgkie jïh vaarjele gaajhkide tjoejide, guvvide jïh trygkesovveme aamhtesidie mejtie Sveerje bæjhkohte.

The coordinated demonstrative phrase does not have consequent case agreement, the nominative plural nouns *guvvieh* and *aamhtsesh* should be in accusative case just as their coordinated predecessor *tjoejide*. The parsed tree in Figure 7 therefore interprets *guvvieh* as a new subject to *vaarjele* and does not make it a daughter of *tjoejide* as it should be. In addition, the nominative plural relative pronoun *mah* has a case error. It should be accusative *mejtie* in order to be identified as the object of the finite verb *vadta*.

4 Creating a preprocessing tool for dependency structure

In order to create a smoother dependency analysis for South Sámi and facilitate treebank building, we decided to preprocess the text by means of a hand-written spelling and grammar checker for the most common error types. We added a grammatical error annotation layer to SIKOR (SIKOR, 2025). We chose a 182,759-token part of the corpus that had been marked up for spelling errors already, and classified the grammatical error types on top of those. Table 3 shows that the corpus contains altogether 740 errors.

A demonstrative phrase error as explained in ex. (3-a) is marked as a unit. The error is then classified with its morpho-syntactic properties – in this case the nominative plural noun should be in accusative plural – and then the whole phrase is repeated in its corrected form as below.

wrong phrase:

```
gaajhkide tjoejide, guvvieh
jïh trygkesovveme aamhtesh
```

error classification:

```
demphrase,noun,plnom-placc
```

corrected phrase:

```
gaajhkide tjoejide, guvvide
jïh trygkesovveme aamhtesidie
```

Based on our annotation we decided to write rules for the most frequent error types that would potentially affect the dependency analysis of the sentences. Table 2 shows the selected error types with a few of their subtypes. The most common errors after adjective form errors and general case errors (for example in habitive constructions or as a result of valency violations) are typically agreement errors, both between subject and verb and noun phrase internal agreement (including quantifiers and demonstratives).

South Sámi demonstrative phrase and numeral phrases differ from Germanic structures and follow complex rules, which is why errors are common. In demonstrative (and indefinite) phrases typically pronouns and nouns agree in number and case. In numeral phrases, on the other hand, only nominative agrees in number and case. In all other cases, the noun is in singular after all numbers above *one*.

In ex. (4-a), the indefinite pronoun nominative plural *gaajhkh* 'all' needs to be changed to accusative *gaajhkide* ' to all' because of the subsequent accusative noun *maanide* 'children' and its agreement requirements.

(4) a. *Seabradahken dåarjoe
community.SG.INE support
maanasåjhtose edtja
childcare.SG.ILL should.PRS.3.SG
gaajhkh maanide båetedh.
all.PL.NOM child.PL.ILL come.INF
'Community support for childcare should reach all children'

b. Seabradahken dåarjoe maanasåjhtose edtja gaajhkide maanide båetedh.

rule type	error	correction
demonstrative phrase case agreement	Dem Nom	
numeral phrase agreement	Num N.Nom.Sg.	Num N.Nom.Pl.
numeral phrase agreement	Num N.Pl.	Num N.Sg.
habitive constructions	Nom. copula Nom.	Gen. copula Nom.
infinitive after auxiliary	aux vfin	aux Inf
postposition complement	Acc Po	Gen Po
subject verb agreement	1. Du	3. Pl.
subject verb agreement	3. Pl.	3. Sg.
subject verb agreement	2. Sg.	3. Pl.
subject verb agreement	3. Sg.	3. Pl.
subject verb agreement	Inf.	3. Pl.
phrasal verb lex verb	V Adv	V
unidiomatic phrasal verb	V Adv	V Adv
negation past tense agreement		
negation verb phrase	Neg Inf	Neg Conneg
adjective forms	attr	Nom. Sg.
	attr	Nom. Pl.
	Nom. Sg.	attr
	Nom. Sg.	adv

Table 2: Rule types checked in the South Sámi grammar checking tool

We also need to account for exceptional use of numerals such as in the following sentence (5), where *nulle* 'zero' is actually used as part of a compound 'zero-object' and not as a quantifier.

(5) Voestes aejkien manne **nulle objeekten** bïjre govlim utnim luste goerehtidh maam ij våajnoes aktene raajesisnie.
'The first time I heard about the zero object, I thought it was fun, which wasn't in a sentence.'

Apart from demonstrative phrase, numeral phrase and nominal phrases involving adjectives, also postpositional phrases can alter the dependency structure in parts of the tree. Ex. (6-a) displays a typical case error in dependents of postpositions. In South Sámi, the correct form is genitive case. However, a frequent error is to use accusative case as *dam* 'the' instead of genitive *dan* 'the'. These errors can also involve coordinated noun phrases such as in ex. (7-a).

(6) a. Janne åådtje munnjien dam bïjre mænngan soptsestidh.
Janne get.PRS.3.SG I.ILL that.ACC about later talk.INF
Janne can talk to me about it later.

b. Janne åådtje munnjien dan bïjre mænngan soptsestidh.

(7) a. Mijjieh sïjhtebe vuejnedh buarastehtemem staaten, faagesiebrieh jïh barkoevedtijh gaskem juktie destie baalhkajoekehts nyjsenæjjide
we want.PRS.1.PL see.INF handshaking.ACC state.GEN.SG, tradeunion.GEN.PL and employer.GEN.PL between
'We want to see a handshake between the state, the tradeunion and the employers.'

b. Mijjieh sïjhtebe buarastehtemem vuejnedh staaten, faagesïebri jïh barkoevedtiji gaskem

Other frequent case errors regard habitive constructions such as the one in ex. (8-a), where the possessor role needs to be in genitive case (*Gaajhkesi*) instead of nominative case *Gaajhkesh* 'everyone'. Only then can they be correctly identified as part of the habitive structure in a dependency analysis.

(8) a. Gaajhkesh leah
everyone.NOM.PL are.PRS.3.PL

Dataset	Full trees	Partial
Originals	915	1296
GEC	1390	811
Hand-corrected	1259	948

Table 3: Automatically parsed dependency trees in SIKOR

reaktah årromesæjjan.
right.NOM.PL housing.ILL.SG
'Everbody has the right to a place to live.'

b. Gaajhkesi leah reaktah årromesæjjan.

Verb phrase errors typically regard subject-verb agreement as in examples (9-a) and (10-a), where the verb form needs to be in first person dual instead of first person plural since two and no more people are performing the action. In order to match the verb with its subject, it needs to be in its correct person and number.

(9) a. Daan biejjien Manne jïh Janne
Today I and Janne go.PRS.1.PL
vuelkebe Afrikese,
Africa.ILL.SG vacation.ILL.SG
eejehtæmman

'Today I and Janne are going to Africa for vacation.'

b. Daan biejjien Manne jïh Janne vuelkien Afrikese, eejehtæmman

(10) a. Mænngan Janne jïh manne
Later Janne and I
edtjebe tjaetsieskuvterem
will.PRS.1.PL water.scooter.ACC.SG
vuejedh!
drive.INF
'I and Janne will later drive a water scooter.'

b. Mænngan Janne jïh manne edtjien tjaetsieskuvterem vuejedh!

The following constraint grammar rules in Figure 8 add errortags to (multiple) demonstrative/indefinite pronouns noun combinations and relate them to each other (ADDRELATION) to create a unified error that will be visualized as one error.

```
ADD (&msyn-demphrase-congruence-plnom) TARGET
(Pron Sg Nom) IF (0 Dem OR Indef)(*1 (N Pl Nom)
BARRIER (*) - (Dem Nom) LINK NEGATE 0 (N Sg
Nom));

ADD (&msyn-demphrase-congruence-plnom) TARGET
(Pron Pl Nom) IF (-1 (Pron Dem Pl Nom &msyn-
demphrase-congruence-plnom)) (1 (N Pl Nom));

ADD (&msyn-demphrase-congruence-plnom) TARGET
(N Pl Nom) IF (-1 (Dem &msyn-demphrase-
congruence-plnom) OR (Indef &msyn-demphrase-
congruence-plnom));

ADDRELATION ($2 LEFT) (&msyn-demphrase-
congruence-plnom) TO (-1 (Dem &msyn-demphrase-
congruence-plnom) OR (Indef &msyn-demphrase-
congruence-plnom)) ;
```

Figure 8: Constraint grammar rules adding error tags to demonstrative phrases

```
"<Almetjh>"
    "almetje" N Sem/Hum Pl Nom <W:0.0> @SUBJ> #1->5
:
"<gieh>"
    "gie" Pron Rel Pl Nom <W:0.0> @SUBJ> #2->5
:
"<daesnie>"
    "daesnie" Adv <W:0.0> @ADVL> #3->4
:
"<barkeminie>"
    "barkedh" <mv> V TV Ger <W:0.0> @IMV #4->0
:
"<lea>"
    "lea" <mv> V IV Ind Prs Sg3 <W:0.0> @FMV #5->0
:
"<tryjjes>"
    "tryjjes" A Sg Nom <W:0.0> @<SPRED #6->5
"<.>"
    "." CLB <W:0.0> #7->4
```

Figure 9: Copula drop dependency analysis of ex. (11)

5 Evaluation

We chose a 100 sentence test corpus, part of SIKOR, to manually evaluate the post spell- and grammar checking dependency analysis and got the following results. 73 of 100 sentences received a correct dependency analysis (73%). Of 633 dependencies distributed to word forms – excluding punctuation – 55 human edits were needed to fix the dependencies. This means that 91.3% of the dependencies are correct. 24 of these edits were necessary because the sentence contains copula drop as shown in the dependency analysis of example (11) in Figure 9. Both the non-finite verb *barkeminie* 'working' of the relative clause and the finite verb of the main clause *lea* 'is' go to the root of the sentence, where only the latter should do so.

(11) Almetjh gieh daesnie
people.NOM.PL who.NOM.PL here
barkeminie, lea tryjjes.
working.GER be.PRS.1.SG friendly.NOM.SG
'People who are working here are friendly.'

Copula drop is a known issue in South Sámi describe thoroughly in Ylikoski (2022), and it appears in different forms – the sentence can drop the auxiliary in periphrastic verbal constructions as the one in the previous example, leaving only the non-finite verb form (past participle, gerund etc.). It can also be dropped in copula constructions, leaving only the subject and the predicate. When there are complex sentences with main- and subclause, where the mainclause has copula drop, while the subclause has a finite verb form, the automatic analyzer often analyses the finite verb form of the subclause as the daughter of the root, instead of making it the daughter of the non-finite verbform of the main clause. South Sámi syntax poses challenges to machine-based dependency analysis, which languages with required finite verbs do not, and new solutions need to be carefully investigated.

Other reasons for failing dependencies are remaining spelling and grammar errors (6), and shortcomings in the analysis regarding coordination (7) and finding the correct verbal mother (12).

6 Conclusion

Low resource languages like South Sámi need language resources and treebanks like all other languages. Our approach has taken into account that South Sámi lacks human resources to mark up large amounts of texts to create a treebank by applying a rule-based tool to do so. Instead, we have used our human resources to create and improve rule-based grammar checking and dependency tools so that we can post-edit our treebank with much less effort than creating it from scratch. We have further identified one of the causes of noise in the creation of such resources – spelling and grammatical errors. We therefore enhanced a marked-up error corpus to systematically identify the most frequent grammatical errors that can get into the way of automatic dependency annotation. These include both, errors on the noun phrase and the verb phrase level - demonstrative phrases, numeral phrases, adjectival forms, case errors in habitive constructions and postpositional phrase being a few of them. Based on this analysis we have written rules for all the previous error types to automatically identify and correct these errors and preprocess the input text for the dependency analyzer. We can see that the number of full and partial trees increases with the correction of these grammatical errors, and our current dependency tool gives us 91.3% of correct dependency relations. We were also able to identify the main reasons for remaining flaws in our system. They are related to South Sámi being a copula drop language, which makes it more challenging to identify the roots of these sentences, which can either be a non-finite verb or a nominal phrase. This pecularity of South Sámi will also be interesting when comparing its treebank with the one of other languages. As a next step, we plan to improve our dependency tool and with some human post-editing create the first South Sámi treebank.

We have seen that our method is an efficient way of creating a treebank, a dependency tool and a grammar checker at that same time, all of which can be used as language resources and proofing tools by the South Sámi language community.

References

Lene Antonsen, Trond Trosterud, and Linda Wiechetek. 2010. Reusing grammatical resources for new languages. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Kenneth R Beesley and Lauri Karttunen. 2003. Finite-state morphology: Xerox tools and techniques. *CSLI, Stanford*, pages 359–375.

Knut Bergsland and Lajla Mattsson Magga. 1993. *Åarjelsaemien-daaroen baakoegærja*. Iđut, Alta.

Eckhard Bick. 2019. Dependency trees for greenlandic. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*, pages 140–148. German Society for Computational Linguistics & Language Technology.

Eckhard Bick. 2022. A modular machine translation pipeline for greenlandic. In *Proceedings of The International Conference on Agglutinative Language Technologies as a Challenge of Natural Language Processing (ALTNLP 2022). CEUR workshop proceedings, Vol 3315. ISSN 1613-0073*.

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.

Marie-Catherine De Marneffe, Christopher D Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational linguistics*, 47(2):255–308.

Fred Karlsson. 1990. Constraint grammar as a framework for parsing running text. In *COLING 1990 Volume 3: Papers presented to the 13th International Conference on Computational Linguistics*.

Ole Henrik Magga and Lajla Mattsson Magga. 2012. *Sørsamisk grammatikk*. Davvi girji, Kárásjohkka.

Alexandre Rademaker, Fabricio Chalub, Livy Real, Cláudia Freitas, Eckhard Bick, and Valeria De Paiva. 2017. Universal dependencies for portuguese. In *Proceedings of the fourth international conference on dependency linguistics (Depling 2017)*, pages 197–206.

Mariya Sheyanova and Francis M. Tyers. 2017. Annotation schemes in north sámi dependency parsing. In *Proceedings of the 3rd International Workshop for Computational Linguistics of Uralic Languages*, pages 66–75.

SIKOR. 2025. SIKOR UiT Norgga Árktalaš universitehta ja Norgga Sámedikki sámi teakstačoakkáldat, veršuvdna 2025-04-23. http://gtweb.uit.no/korp.

Linda Wiechetek and Maja Lisa Kappfjell. 2023. A South Sámi grammar checker for stopping language change. In *Proceedings of the NoDaLiDa 2023 Workshop on Constraint Grammar - Methods, Tools and Applications*, pages 46–54, Tórshavn, Faroe Islands. Association of Computational Linguistics.

Jussi Ylikoski. 2022. South sámi. *The Oxford Guide to the Uralic Languages*, page 113–129.