

# Rules Ruling Neural Networks – Neural vs. Rule-Based Grammar Checking for a Low Resource Language

**Linda Wiechetek**  
UiT Norgga árktalaš  
universitehta  
Norway

**Flammie A Pirinen**  
UiT Norgga árktalaš  
universitehta  
Norway

**Mika Hämäläinen**  
University of Helsinki,  
Rootroo Ltd  
Finland

**Chiara Argese**  
UiT Norgga árktalaš  
universitehta  
Norway

## Abstract

We investigate both rule-based and machine learning methods for the task of compound error correction and evaluate their efficiency for North Sámi, a low resource language. The lack of error-free data needed for a neural approach is a challenge to the development of these tools, which is not shared by bigger languages. In order to compensate for that, we used a rule-based grammar checker to remove erroneous sentences and insert compound errors by splitting correct compounds. We describe how we set up the error detection rules, and how we train a bi-RNN based neural network. The precision of the rule-based model tested on a corpus with real errors (81.0%) is slightly better than the neural model (79.4%). The rule-based model is also more flexible with regard to fixing specific errors requested by the user community. However, the neural model has a better recall (98%). The results suggest that an approach that combines the advantages of both models would be desirable in the future. Our tools and data sets are open-source and freely available on GitHub and Zenodo.

## 1 Introduction

This paper presents our work on automatically correcting compound errors in real world text of North Sámi and exploring both rule-based and neural network methods. We chose this error type as it is the most frequent grammatical error type (after spelling and punctuation errors) and twice as frequent as the second most frequent grammatical error (agreement error). It also regards both spelling and grammar as the error is a space between two words, but its correction requires grammatical context.

A grammar checker is a writer's tool and particularly relevant to improve writing skills of a minority language in a bilingual context, as is the case

for North Sámi. According to UNESCO (Moseley, 2010), North Sámi, spoken in the North of Norway, Sweden and Finland, has around 30,000 speakers. It is a low resource language in a bilingual setting, and language users frequently face bigger challenges to writing proficiency as there is always a competing language. (Outakoski, 2013) Developing a reliable grammar checker with a high precision that at the same time covers a lot of errors has therefore been our main focus. Good precision (i.e. avoiding false alarms) is a priority because users get easily frustrated if a grammar checker gives false alarms and underlines correct sentences.

In this paper we focus on the correction of compound errors. This type of errors is easy to generate artificially in the absence of large amounts of error marked-up text, and we have a good amount of manually marked-up corpus for evaluation for this error type. Compound errors (i.e. one-word compounds that are erroneously written as two words) can be automatically inserted by using a rule-based morphological analyser on the corpus and splitting the word wherever we get a compound analysis. Unlike other error types (like e.g. real word errors) they are easily inserted, and existing compounds are seldom errors. In addition, they are interesting from a linguistic point of view as they are proper (complex) syntactic errors and not just spelling errors and serve as an example for higher level tools. Two adjacent words can either be syntactically related or erroneous compounds, depending on the syntax. In North Sámi orthography, as in the majority languages spoken in the region (Norwegian, Swedish and Finnish), nouns that form a new concept are usually written together. For example, the North Sámi word *boazodoalloguovlu* 'reindeer herding area' consists of three words *boazu* 'reindeer', *doallu* 'industry' and *guovlu* 'area', and thus it is written together as a single compound. The task of our methods is to correct spellings such as

*boazodoallu guovlu* into *boazodoalloguovlu* in case the words have been written separately in error.

We develop both a rule-based and a neural model for the correction of compound errors. The rule-based model (*GramDivvun*) is based on finite-state technology and Constraint Grammar. The neural model is bi-directional recurrent (BiRNN). While the rule-based model has earlier produced good precision, it did not handle unknown compounds well, which is why we were interested in a neural approach. However, neural models depend on large amounts of ‘clean’ data and synthetic error generation (or alternatively marked-up data). Typical for low-resource languages and also North Sámi, the corpora are not clean and contain a fair amount of a variety of different spelling and grammatical errors (see Antonsen 2013). Therefore, efficiently preparing data as to making it available for neural model training is an important part of this paper. In our case, we make use of the existing rule-based tools to both, generate synthetic error data and clean the original data for training. For evaluation, on the other hand, we use real world error data.

Our free and open-source rule-based tools can be found on GiellaLT GitHub.<sup>1</sup> The training data and the neural models are freely available on Zenodo.<sup>2</sup> We hereby want to promote a wider academic interest in conducting NLP research for the North Sámi.

## 2 Background

Sámi open source rule-based language tools have a long and successful tradition (nearly 20 years) (Trosterud, 2004; Moshagen, 2011; Antonsen and Trosterud, 2011; Rueter and Hämäläinen, 2020). North Sámi is a low-resource language in terms of available corpus data (32.24M tokens raw data). Although there is a fair amount of data, it contains many real errors and only a small amount is marked up for errors. Applying neural approaches for high-level language tasks to low resource languages is an interesting research question due the various limitations of minority language corpora, versus the existing research in the topic in well-resourced, majority languages and artificially constrained setups (Nekoto et al., 2020). Rules have been used and are in a wide-spread use in the context of endangered Uralic languages. There is recent work on grammar checking for North Sámi (Wiechetek

et al., 2019a) and spell checking for Skolt Sámi (Trosterud and Moshagen, 2021). Other rule-based approaches to grammar checking are extensively described in Wiechetek (2017).

Before the era of neural models, it was common to use statistical machine translation (SMT) as a method for grammar error correction (Behera and Bhattacharyya, 2013; Kunchukuttan et al., 2014; Hoang et al., 2016). Many recent papers on grammar checking use bi-directional LSTM models that are trained to tag errors in an input sentence. Such methods have been proposed for Latvian (Deksne, 2019), English (Rei and Yannakoudakis, 2016) and Chinese (Huang and Wang, 2016). Similar LSTM based approaches have also been applied for error correction (Yuan and Briscoe, 2016; Ge et al., 2019; Jahan et al., 2021). Other recent approaches (Kantor et al., 2019; Omelianchuk et al., 2020) use methods that take advantage of BERT (Devlin et al., 2019) and other data-hungry models. While such rich sentence embeddings can be used for English and a few other languages with a large amount of data, their use is not viable for North Sámi.

## 3 Data

For evaluation and training the neural model we use the **SIKOR** (2018) (the Sámi International Korpus), which is a collection of texts in different Sámi languages compiled by UiT The Arctic University of Norway and the Norwegian Sámi Parliament. It consists of two subcorpora: *GT-Bound*<sup>3</sup> (texts limited by a copyright which are available only by request) and *GT-Free*<sup>4</sup> (the publicly available texts). As a preprocessing step, we run a rule-based grammar checker (Wiechetek, 2012) and remove sentences with potential compound errors, as we cannot automatically ensure whether these errors are real or not. This is needed as we want this data to be fully free of any compound errors as it serves as the target side of our neural model.

Thereafter, we take in each sentence in this error free data and analyse it by a rule-based morphological analyser<sup>5</sup>. When the analyser sees a potential compound word, it indicates the word boundary with a compound (+Cmp#) tag. We use this information to automatically split all compounds identified by the rule-based analyser. This results in a

<sup>1</sup><https://github.com/giellalt/>

<sup>2</sup><https://zenodo.org/record/5172095>

<sup>3</sup><https://gtsvn.uit.no/boundcorpus/orig/sme/>

<sup>4</sup><https://gtsvn.uit.no/freecorpus/orig/sme/>

<sup>5</sup><https://github.com/giellalt/lang-sme>

parallel corpus of the original sentences as the prediction target and their corresponding versions with synthetically introduced compound errors. Many of the compound boundaries are ambiguous, and the algorithm decides the one used in training data based on heuristics: maximum number of compound boundaries where the splitting will not cause any other modifications of the word stems or other content.

As an additional data source, we use the North Sámi Universal Dependencies treebank (Tyers and Sheyanova, 2017). We parse the corpus with Uralic-NLP (Hämäläinen, 2019) and split the compounds the rule-based morphological analyser identifies as consisting of two or more words in order to synthetically introduce errors. We also run the rule-based morphological analyser and morpho-syntactic disambiguator to add *part-of-speech* (POS) information to produce an additional data set with POS tags. For the Universal Dependencies data, we use the POS tags provided in the data set.

We then make sure that all sentences have at least one generated compound error and that the only type of error the sentences have is the compound error (no other changes introduced by the rule-based models). We shuffle this data randomly and split it on a sentence level into 70% training, 15% validation and 15% testing. The size of the data set can be seen in Table 1, the sentences were tokenized based on punctuation marks.

	Sentences	Source tokens
<b>Train</b>	43,658	388,167
<b>Test</b>	9,356	83,107
<b>Validation</b>	9,355	82,566
<b>Real-world errors</b>	3,291	26,565

Table 1: Training, testing and validation sizes for the **neural model** (corpus with synthetic errors)

For the rule-based model *GramDivvun* we do not generate synthetic errors. We have hand-selected a large corpus for rule development and as regression tests, consisting of representative sentences from *GT-Free*. The current selection for syntactic compound errors includes 3,291 sentences with real world compound errors (and possibly other errors in addition).

## 4 Methods

We use a neural models and a rule-based model for compound error correction.

### 4.1 Neural Model

We model the problem at a character instead of word level in NMT (neural machine translation). The reason for using a character-level model instead of a word-level model is that, this way, the model can work better with out-of-vocabulary words. This is important due to the low-resourced nature of North Sámi, although there are other deep learning methods for endangered languages that do not utilize character level models (Alnajjar, 2021). In practice, we split words into characters separated by white spaces and mark actual spaces between words with an underscore (\_). We train the model to predict from text with compound errors into text without compound errors. As previous research (Partanen et al., 2019; Alnajjar et al., 2020) has found that using chunks of words instead of full sentences at a time improves the results in character level models, we will be training different models with different chunk sizes. This means that we will train a model to predict two words at a time, three words at a time, all the way to five words at a time.

We train the models with and without POS tags. For the models with POS tags, we surround each word with a token indicating the beginning and the end of the POS tag. The POS tags are included only on the source side, not on the target side. They are separated from the word with a white space.

An example of the data can be seen in Table 2. Even though every sentence in the training data has a compound error, this does not mean that every input chunk the model sees would have a compound error. This way, the model will also learn to leave the input unchanged if no compound errors are detected.

We train all models using a bi-directional long short-term memory (LSTM) based model (Hochreiter and Schmidhuber, 1997) by using OpenNMT-py (Klein et al., 2017) with the default settings except for the encoder where we use a BiRNN (Schuster and Paliwal, 1997) instead of the default RNN (recurrent neural network), since BiRNN based models have been shown to provide better results in character-level models (Hämäläinen et al., 2019). We use the default of two layers for both the encoder and the decoder and the default attention model, which is the general global attention presented by Luong et al. (2015). The models are trained for the default of 100,000 steps. All models are trained with the same random seed (3,435) to ensure reproducibility.

n	Input	Output
2	geahččaladdan_prošeaktan	geahččaladdanprošeaktan
3	geahččaladdan_prošeaktan_prošeaktan	geahččaladdanprošeaktan_prošeaktan
2	V>geahččaladdan<V_N>prošeaktan<N	geahččaladdanprošeaktan
3	V>geahččaladdan<V_N>prošeaktan<N_N>jagi<N	geahččaladdanprošeaktan_prošeaktan

Table 2: Examples of the character-level input and output, where  $n$  indicates the chunk size. The first examples are without POS tags and the last with POS tags

During the training of the neural models, we evaluate the models using simple sentence level scores. There we look only at full-sentence matches and evaluate their accuracy, precision and recall, as opposed to the evaluations in Section 5, where we study them more carefully at the word-level. The results of the neural models for the generated corpus (where errors were introduced by splitting compounds) can be seen in Table 3. The results indicate that both of the models receiving a chunk of two words at a time reached to the highest accuracy, and the model without the POS tags also reached to the highest precision.

Chunk	POS	Accuracy	Precision	Recall
2	no	<b>0.925</b>	<b>0.949</b>	0.974
3	no	0.847	0.883	0.955
4	no	0.852	0.892	0.950
5	no	0.869	0.909	0.952
2	yes	<b>0.925</b>	0.948	<b>0.976</b>
3	yes	0.906	0.934	0.968
4	yes	0.856	0.896	0.951
5	yes	0.857	0.895	0.953

Table 3: Sentence level scores for different neural models tested on a corpus with artificially introduced errors

The POS tags were not important for the models, as the results with and without them are fairly similar. The largest gain was when the compound error correction was done for three words at a time. As this performance gain only occurred for that specific model, it suggests that it is more of an artefact of the training data and how it is fed into the model than any actual improvement.

## 4.2 Rule-based Model

The rule-based grammar checker *GramDivvun* is a full-fledged grammar checker fixing spelling errors, (morpho-)syntactic errors (including real word spelling errors<sup>6</sup>, inflection errors, and compounding errors) and punctuation and spacing errors.

It takes input from the finite-state transducer (*FST*) to a number of other modules, the core

<sup>6</sup>Real word errors are spelling errors where the outcome is an actual word that is not fit for the context.

of which are several Constraint Grammar modules for tokenization disambiguation, morpho-syntactic disambiguation and a module for error detection and correction. The full modular structure (Figure 1) is described in Wiecheteck (2019b). This work regards predominantly the modification of the disambiguation and error detection modules *mwe-dis.cg3*, *grc-disambiguator.cg3*, and *grammarchecker-release.cg3*. 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, 1990; 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<sup>7</sup> with specific version tagged for reproducibility.

The syntactic context is specified in hand-written Constraint Grammar rules. The REMOVE-rule below removes the compound error reading (identified by the tag `Err/SpaceCmp`) if the head is a 3rd person singular verb (cf. 1.2) and the first element of the potential compound is a noun in nominative case (cf. 1.3). The context condition further specifies that there should be a finite verb (VFIN) somewhere in the sentence (cf. 1.4) for the rule to apply.

```
REMOVE (Err/SpaceCmp)
(0/0 (V Sg3))
(0/1 (N Sg Nom))
(*0 VFIN);
```

All possible compounds written apart are considered to be errors by default, unless the lexicon specifies a two or several word compound or a syntactic rule removes the error reading.

<sup>7</sup><https://github.com/giellalt/lang-sme/releases/tag/naacl-2021-ws>



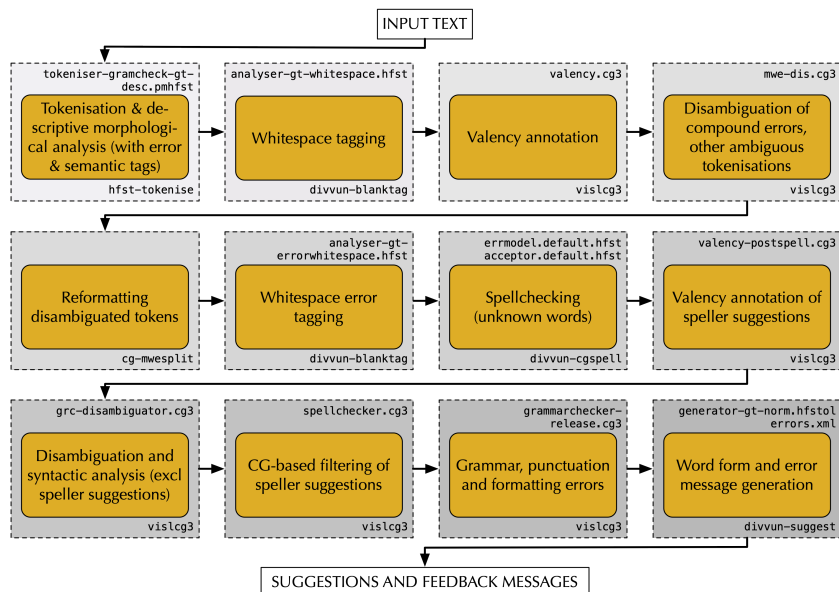


Figure 1: System architecture of the North Sámi grammar checker (*GramDivvun*)

The process of rule writing includes several consecutive steps, and like neural network models they require data. The process is as follows:

1. Modelling an error detection rule based on at least one actual sentence containing the error
2. Adding constraints based on the linguist’s knowledge of possible contexts (remembered data)
3. A corpus search for sentences containing similar forms/errors, testing of the rule and reporting rule mistakes
4. Modification of constraints in the rule based on this data and testing against regression tests so that unfit constraints depending on results for precision and recall (focus on precision)

The basis of rule development is continuous integration. Typical shortcomings and bad errors can be fixed right away with added conditions. Neural models are not usually trained in this way.

The frequent experience of false alarms can decrease the users’ trust in the grammar checker. Typically, full-fledged user oriented grammar checkers, e.g. *DanProof* focus on keeping false alarms low and precision high (Bick, 2015) because users’ experiences have shown that certain experiences will frustrate users and stop them from using the application.

For rule development, regression tests are used. These consist in error-specific YAML<sup>8</sup> tests and

<sup>8</sup><https://yaml.org/spec/1.2/spec.html>

are manually marked up. The regression test for compound errors contains 3,291 sentences (1,368 compound errors, used for development and regression) give the results as shown in Table 4.

Precision	Recall	$F_1$ score
94.95	86.22	90.80

Table 4: The **rule-based model** tested on the developer’s corpus (regression tests)

## 5 Results

We evaluate the models both quantitatively and qualitatively. We evaluate on accuracy, precision and recall, and do a linguistic evaluation. The measurements are defined in this article as follows: Accuracy  $A = \frac{C}{S}$ , where  $C$  is a correct sentence (1:1 string match) and  $S$  is corpus size in sentences, precision  $P = \frac{tp}{tp+fp}$  and recall  $R = \frac{tp}{tp+fn}$ , where  $tp$  is true positive,  $fp$  is false positive and  $fn$  is false negative. The  $F_1$  score is the harmonic mean of precision and recall  $F_1 = 2 \times \frac{P \times R}{P + R}$ . The accuracy is thus sentence level correctness rate—as used in the method section to probe model qualities—whereas precision measures how often corrections were right and recall measures how many errors we found. The word-level errors are counted once per error in the marked-up corpus. Thus, if a three-part compound contains two compounding errors it is counted towards the total as one error, but if a sentence has three separate compounds with wrong splits each, we count three errors.



with false negatives, like *njunuš olbmot* ‘leading people’ that are due to missing entries in the lexicon like in ex. (5).

- (5) Sii leat gieldda **njunuš olbmot.**  
they are municipality.GEN leading people  
‘They are the leading people of the municipality’

## 6 Discussion

In the future, we would like to look into hybrid grammar checking of other error types and other (Sámi) languages.

The neural approach gives us relatively high recall in the real world situation with lower precision, whereas the rule-based model is designed to give us high precision even at the cost of lower recall (user experience), which is why hybrid approaches that combine the best of two worlds are interesting.

Noisy data is to be expected in any endangered language context, as the language norms are to a lesser degree internalized. We will therefore need a way of preparing the data to train neural networks, which can either consist in creating synthetic data or automatically fixing errors and creating a parallel corpus.

When creating synthetic data for neural networks, the amount of data is hardly the main issue. Many generative systems are capable of over-generating data. The main question that arises is the quality and representatives (Hämäläinen and Alnajjar 2019) of the generated data. If the rules used to generate the data are not in line with the real world phenomenon the neural model is meant to solve, we cannot expect very high quality results in real world data.

Generated sentences can easily be less complex ‘text book examples’ that are not representative of real world examples. In the case of agreement errors between subjects and verbs, for example, there are long distance relationships and complex coordinated subjects including personal pronouns that can change the structure of a seemingly straightforward relation. Therefore, we advocate the use of high quality rule-based tools to prepare the data, i.e. fix the errors and create a parallel corpus.

While synthetic error data generation for compound errors is somewhat more straightforward as it only affects adjacent words, the generation of synthetic error corpora for other error types is not as straightforward, in part also because generating synthetic errors of other kind can potentially

create valid and grammatically correct sentences with different meanings. We therefore predict that (hybrid) neural network approaches for other error types that either involve specific morphological forms (of which there are many in North Sámi) or changes in word order will be more difficult to resolve.

## 7 Conclusion

In this paper, we have developed both a neural network and a rule-based grammar checker module for compound errors in North Sámi. We have shown that a neural compound-corrector for a low-resource language can be built based on synthetic error data by introducing the compound errors using a high level rule-based grammar models. It is based on the rule-based tools to both generate errors and clean the data using both part-of-speech analysis, disambiguation and even the error detector.

The rule-based module is embedded in the full-fledged *GramDivvun* grammar checker and achieves a good precision of 81% and a lower recall of 61%. A higher precision, even at the cost of a lower recall, is in line with our objective of keeping false alarms low, so users will be comfortable using our language tools. The neural network achieves a slightly lower precision of 79% and a much higher recall of 98%.

However, the rule-based model has more user-friendly suggestions and some false positives are simply other correct alternatives to the ones in the text, while the neural network’s false positives sometimes introduce new and unrelated errors. On-the-fly fixes that avoid false positives are an advantage of rule-based models. Rule-based models, on the other hand, are not so good at recognizing unknown combinations. Hybrid models that combine the benefits of both approaches are therefore desirable for efficient compound error correction in the future.

## Acknowledgments

Thanks to Børre Gaup for his work on the evaluation script. Some computations were performed on resources provided by UNINETT Sigma2 – the National Infrastructure for High Performance Computing and Data Storage in Norway.

## References

- Khalid Alnajjar. 2021. [When word embeddings become endangered](#). In Mika Hämmäläinen, Niko Partanen, and Khalid Alnajjar, editors, *Multilingual Facilitation*, pages 275–288. Rootroo Ltd, Finland.
- Khalid Alnajjar, Mika Hämmäläinen, Niko Partanen, and Jack Rueter. 2020. Automated prediction of medieval Arabic diacritics. *arXiv preprint arXiv:2010.05269*.
- Lene Antonsen. 2013. Cállinmeattáhusaid guorran. *University of Tromsø*. [English summary: Tracking misspellings.].
- Lene Antonsen and Trond Trosterud. 2011. Next to nothing—a cheap south saami disambiguator. In *Proceedings of the NODALIDA 2011 Workshop Constraint Grammar Applications*, pages 1–7.
- Kenneth R. Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Studies in Computational Linguistics. CSLI Publications, Stanford.
- Bibek Behera and Pushpak Bhattacharyya. 2013. Automated grammar correction using hierarchical phrase-based statistical machine translation. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 937–941.
- Eckhard Bick. 2015. DanProof: Pedagogical spell and grammar checking for Danish. In *Proceedings of the 10th International Conference Recent Advances in Natural Language Processing (RANLP 2015)*, pages 55–62, Hissar, Bulgaria. INCOMA Ltd.
- 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.
- Daiga Deksnė. 2019. Bidirectional lstm tagger for latvian grammatical error detection. In *International Conference on Text, Speech, and Dialogue*, pages 58–68. Springer.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Tao Ge, Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. Automatic grammatical error correction for sequence-to-sequence text generation: An empirical study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6059–6064.
- Mika Hämmäläinen and Khalid Alnajjar. 2019. [A template based approach for training nmt for low-resource uralic languages - a pilot with Finnish](#). In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence, ACAI 2019*, page 520–525, New York, NY, USA. Association for Computing Machinery.
- Mika Hämmäläinen, Tanja Säily, Jack Rueter, Jörg Tiedemann, and Eetu Mäkelä. 2019. Revisiting nmt for normalization of early English letters. In *Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 71–75.
- Duc Tam Hoang, Shamil Chollampatt, and Hwee Tou Ng. 2016. Exploiting n-best hypotheses to improve an smt approach to grammatical error correction. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2803–2809.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Shen Huang and Houfeng Wang. 2016. Bi-lstm neural networks for chinese grammatical error diagnosis. In *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*, pages 148–154.
- Mika Hämmäläinen. 2019. [UralicNLP: An NLP library for Uralic languages](#). *Journal of Open Source Software*, 4(37):1345.
- Mir Noshin Jahan, Anik Sarker, Shubra Tanchangya, and Mohammad Abu Yousuf. 2021. Bangla real-word error detection and correction using bidirectional lstm and bigram hybrid model. In *Proceedings of International Conference on Trends in Computational and Cognitive Engineering*, pages 3–13. Springer.
- Yoav Kantor, Yoav Katz, Leshem Choshen, Edo Cohen-Karlik, Naftali Liberman, Assaf Toledo, Amir Menczel, and Noam Slonim. 2019. Learning to combine grammatical error corrections. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 139–148.
- Fred Karlsson. 1990. 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, Atro Voutilainen, Juha Heikkilä, and Arto Anttila. 1995. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin.



- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senelhart, and Alexander M. Rush. 2017. [OpenNMT: Open-Source Toolkit for Neural Machine Translation](#). In *Proc. ACL*.
- Anoop Kunchukuttan, Sriram Chaudhury, and Pushpak Bhattacharyya. 2014. Tuning a grammar correction system for increased precision. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 60–64.
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- Christopher Moseley, editor. 2010. *Atlas of the World's Languages in Danger*, 3rd edition. UNESCO Publishing. Online version: <http://www.unesco.org/languages-atlas/>.
- Sjur Moshagen. 2011. Tilgjengelegheit for samisk og andre nasjonale minoritetsspråk. In *Språkteknologi for økad tillgänglighet*.
- Sjur N. Moshagen, Tommi A. Pirinen, and Trond Trosterud. 2013. Building an open-source development infrastructure for language technology projects. In *NODALIDA*.
- Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Taiwo Fagbohunbe, Solomon Oluwole Akinola, Shamsuddeen Muhammad, Salomon Kabongo Kabenamualu, Salomey Osei, Freshia Sackey, Rubungo Andre Niyongabo, Ricky Macharm, Perez Ogayo, Orevaoghene Ahia, Musie Meressa Berhe, Mofetoluwa Adeyemi, Masabata Mokgesi-Seling, Lawrence Okegbemi, Laura Martinus, Kolawole Tajudeen, Kevin Degila, Kelechi Ogejeji, Kathleen Siminyu, Julia Kreutzer, Jason Webster, Jamiil Toure Ali, Jade Abbott, Iroro Orife, Ignatius Ezeani, Idris Abdulkadir Dangana, Herman Kamper, Hady Elshahar, Goodness Duru, Ghollah Kioko, Murhabazi Espoir, Elan van Biljon, Daniel Whitenack, Christopher Onyefuluchi, Chris Chinenye Emezue, Bonaventure F. P. Dossou, Blessing Sibanda, Blessing Basse, Ayodele Olabiyi, Arshath Ramkilowan, Alp Öktem, Adewale Akinfaderin, and Abdallah Bashir. 2020. Participatory research for low-resourced machine translation: A case study in African languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2144–2160. Online. Association for Computational Linguistics.
- Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanyskiy. 2020. Gector—grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170.
- Hanna Outakoski. 2013. Davvisámegiela čálamáhtu konteaksta [The context of North Sámi literacy]. *Sámi dieđalaš áigečála*, 1/2015:29–59.
- Niko Partanen, Mika Härmäläinen, and Khalid Alnajjar. 2019. Dialect text normalization to normative standard Finnish. In *The Fifth Workshop on Noisy User-generated Text (W-NUT 2019)*, page 141–146, United States. The Association for Computational Linguistics.
- 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.
- Marek Rei and Helen Yannakoudakis. 2016. [Compositional sequence labeling models for error detection in learner writing](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1181–1191, Berlin, Germany. Association for Computational Linguistics.
- Jack Rueter and Mika Härmäläinen. 2020. Fst morphology for the endangered Skolt Sami language. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced Languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 250–257.
- Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.
- SIKOR. 2018. SIKOR uit norgga árkatalaš universitehta ja norgga sámédikki sámi teakstačoačkáldat, veršuvdna 06.11.2018. <http://gtweb.uit.no/korp>. Accessed: 2018-11-06.
- Trond Trosterud. 2004. Porting morphological analysis and disambiguation to new languages. In *SALTMIL Workshop at LREC 2004: First Steps in Language Documentation for Minority Languages*, pages 90–92. Citeseer.
- Trond Trosterud and Sjur Moshagen. 2021. Soft on errors? the correcting mechanism of a Skolt Sami spell checker. In Mika Härmäläinen, Niko Partanen, and Khalid Alnajjar, editors, *Multilingual Facilitation*, pages 197–207. Rootroo Ltd.
- Francis M. Tyers and Mariya Sheyanova. 2017. [Annotation schemes in North Sámi dependency parsing](#). In *Proceedings of the Third Workshop on Computational Linguistics for Uralic Languages*, pages 66–75, St. Petersburg, Russia. Association for Computational Linguistics.
- Linda Wiecheteck. 2012. Constraint Grammar based correction of grammatical errors for North Sámi. In *Proceedings of the Workshop on Language Technology for Normalisation of Less-Resourced Languages (SALTMIL 8/AFLAT 2012)*, pages 35–40, Istanbul, Turkey. European Language Resources Association (ELRA).

- Linda Wiechetek. 2017. *When grammar can't be trusted – Valency and semantic categories in North Sámi syntactic analysis and error detection*. PhD thesis, UiT The Arctic University of Norway.
- Linda Wiechetek, Sjur Moshagen, and Kevin Brubeck Unhammer. 2019a. Seeing more than white-space—tokenisation and disambiguation in a north Sámi grammar checker. In *Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages Volume 1 (Papers)*, pages 46–55.
- Linda Wiechetek, Sjur Nørstebø Moshagen, Børre Gaup, and Thomas Omma. 2019b. 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.
- Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386.