

Apertium-sme-eng for WMT2026 shared task—creating new pair by crossing dictionaries and translating rules

Flammie A Pirinen

UiT Norgga Árkktalaš universitehta

Divvun

Tromsø, Norway

first.last@uit.no

Abstract

In this article I describe an experiment on creation of a new rule-based machine translation model build on two existing models. I built Apertium system between North Sámi and English based on existing translation models between North Sámi and Finnish, and Finnish and English. There were two expert-driven processes involved in building of the system: firstly crossing and validating the bilingual dictionaries and secondly translating and modifying the grammatical and structural rules. I briefly explain why there are no pre-existing systems for this language pair and what limitations and ethical issues are related to it. I will also evaluate some of the training and development corpora provided in the shared task to elaborate on the problem that automatic low quality machine translation can cause to vulnerable minority languages.

1 Introduction

North Sámi (ISO-639: se, sme)¹ is an Indigenous low resource language from Uralic language family. There is a long tradition of rule-based natural language processing within Uralic languages and also specifically with North Sámi. This includes several rule-based machine translation systems between North Sámi and languages that are closely related, e.g. other Sámi languages, related and relevant majority languages, e.g. Finnish (ISO-639: fi, fin) in North Sámi speaking areas of Finland, or just related majority languages like the Norwegian Bokmål (ISO-639: nb, nob) language in North Sámi areas of Norway. There was no pre-existing rule-based machine translation that we know of between North Sámi and English (ISO-639: en, eng). Apertium is a rule-based machine translation system that is most popularly used for low-resourced languages

including North Sámi and medium-resourced languages like Finnish. This article describes an experiment for building a new Apertium language pair from existing language pairs. Two practical approaches or workflows were used to build the necessary resources for a baseline translator: the North Sámi—English dictionary was created by semi-automatically combining North Sámi—Finnish and Finnish—English dictionaries, and the grammatical and structural transfer rules were created by translating and adapting the rules from the Finnish—English pair.

The shared tasks in WMT² are nowadays mainly directed towards data-driven methods like AI, rather than knowledge-driven methods like rule-based machine translation. However, high quality gold data is also crucial for rule-based machine translation. Instead of learning rules from data, the gold translations are used as tests when creating dictionaries and rules. I will describe the process of using corpus data for development phase of rule-based machine translation to draw a parallel towards data-driven approaches. In this article I also briefly go through some problems I faced when using the WMT 2026 data during the testing and development face of my system.

Finally I discuss about the ethical issues in creating machine translations, especially into a low resourced language that is more vulnerable to disruption.

2 Background

North Sámi is an Indigenous language in Uralic language family. It is spoken by approx. 25,000 people according to Wikipedia citing Ethnologue, most of the speakers are bilingual with most common majority language being Norwegian or Finnish. Like all Uralic languages, North Sámi is more morphologically complex than Indo-European lan-

¹we use the ISO-639 codes as shorthand throughout the article in tables etc.

²<https://www2.statmt.org/wmt26/index.html>

guages. North Sámi is low resource language; the available corpora are in range of few millions of tokens (SIKOR, 2018) and with the number of speakers it is not feasible to expect large corpora of human-written high quality texts in next few years (Steingrímsson et al., 2024). English is an Indo-European language with high resources and number of speakers. It is morphologically simpler than Uralic languages and has different syntactical features. The linguistic distance between languages makes it more tricky for automatic machine translation, especially linguistic rule-based system.

Apertium is a rule-based machine translation system, that is especially popular within low resource languages and also various minority language contexts. (Khanna et al., 2021) Apertium is based on modular system, where the translation is build on pipelines that support different technologies and implementations. For the North Sámi—English experiment, I use standard modern Apertium pipeline with finite-state morphological analysis and generation, lexical transfer, and recursive descent transfer, I did not implement anything extra for the other components (such as anaphora resolution, gapped multiword detection etc.) of the system within the context of the shared task. The resulting system I have built, works for both English-to-North Sámi and North Sámi-to-English directions, but for this paper, I concentrate on the shared task direction English-to-North Sámi. The background for this is elaborated in the discussion section.

The concrete technical implementation of the pipeline at the time of writing was as follows:

1. Finite-State morphology analysis, using HFST (Lindén et al., 2011)
2. Morphological disambiguation, using VISL CG 3 (Bick and Didriksen, 2015)
3. Dependency syntactic disambiguation, using VISL CG 3
4. Apertium pre-transfer
5. Lexical translation using Apertium Ittoolbox
6. Lexical selection using Apertium lextools
7. Recursive descent syntactic transfer using Apertium recursive
8. Finite-State morphological generation using Apertium Ittoolbox

9. Post-processing using Apertium Ittoolbox

For morphological generation (and analysis) of North Sámi I have used the North Sámi language model from GiellaLT repository (Pirinen et al., 2023) called *giella-sme*³. This is a mature morphological analyser and disambiguation also used in other apertium-based North Sámi language pairs. For morphological analysis (and generation) of English I have used Apertium’s standard *apertium-eng*⁴ model.

There are several existing North Sámi translation models in apertium repository:

- *apertium-sme-nob*⁵: Norwegian—North Sámi (Trosterud and Unhammer, 2012),
- *apertium-sme-fin*⁶: Finnish—North Sámi (Johnson et al., 2017), and
- *apertium-sme-sma*⁷: South Sámi—North Sámi (Antonsen et al., 2016),

but there is no existing English—North Sámi pair. I have made use of related high-resource language Finnish and Finnish—English (Pirinen, 2019) resources to help start the development of North Sámi—English.

3 Data

There are two types of existing data I utilised in this experiment: dictionaries and rules from pre-existing apertium models and corpora for testing and development.

The language models I used are summarised in the Table 1, as well as our resulting language model that I built during this experiment.

For this experiment, I used the corpora available from the shared task. (Goldhahn et al., 2012; Tiedemann, 2012) There were some problems⁸ with the scripts for downloading the corpora that resulted in different machines I used during the testing having slightly different datasets, I have therefore summarised the one that seems most complete in the Table 2.

³<https://github.com/giellalt/lang-sme>

⁴<https://github.com/apertium/apertium-eng>

⁵<https://github.com/apertium/apertium-sme-nob>

⁶<https://github.com/apertium/apertium-sme-fin>

⁷<https://github.com/apertium/apertium-sme-sma>

⁸e.g. [downloads.wortschatz-leipzig.de](https://www.wortschatz-leipzig.de) timed out on 2 of 3 machines I used to run the tests

Language(s)	Word(pair)s
<i>Giella-sme</i>	141,000
<i>Apertium-eng</i>	60,000
<i>Apertium-sme-fin</i>	20,000
<i>Apertium-fin-eng</i>	164,000
Apertium-sme-eng	66,000

Table 1: North Sámi and English language data used for building apertium-sme-eng

Corpus collection	size in MiB
OPUS	2.4
Leipzig	3.2

Table 2: North Sámi corpora in WMT 2026 shared task

While testing the system I have built, I discovered some problems with the dataset, for example, the dataset `OPUS-opensubtitles-v2024-eng-sme.sme` does not appear to have a single word North Sámi, but is fully written in South Sámi (ISO-639: `sma`) instead.⁹ Some other datasets like `OPUS-wikimedia-v20230407-eng-sme.sme` contain some North Sámi parts but also large parts in English, Norwegian and other languages; this is not totally unexpected of Wikipedia to contain large snippets of other languages, but it can become unwieldy and unsuitable if this kind of texts dominate the “training corpus”. While tests in North Sámi as input language are not crucial for the use case of building rule-based machine translation from English to North Sámi, they will be for the other translation direction should we continue with that and also for other approaches.

4 Methods

In this experiment, I set out to test the process and workflow of building a new language pair for apertium based on existing pairs. The two main components of this experiment are the building of the bilingual dictionary by combining lexical data from other related pairs, or *crossing* dictionaries, and building the structural translation rules by translating and adapting rules from the related pairs. The related pairs we used for this experiment are North Sámi—Finnish and Finnish—English, which

are both high coverage dictionaries and have relatively mature structural transfer rules. One of the design goals for this methodology is that it needs to be expert-driven, and semi-automatic, while it would be possible to fully automate this process, in my view, the main value of bilingual dictionaries and grammars in apertium project is that they are human-verified.

Creating new bilingual dictionaries by crossing has been used in context of apertium before. There is an outdated Java programme for this in apertium wiki¹⁰, however, it was faster for me to roll out new interactive dictionary crossing tool in python, given the design requirements, than modify the existing tools in Java. The dictionary crossing logic is relatively straightforward, when crossing `sme-fin` and `fin-eng`, we go through words in all dictionaries, and if there is a lemma match in target side of `sme-fin` and source side of `fin-eng`, we suggest it as bilingual translation `sme-eng`. The linguist gets to accept, discard or give penalty weight for each word pair. The penalty weights in apertium’s lexical translation work so, that if there is ambiguity, the word-translation with lowest weight gets selected. Throughout the process of ranking translation pairs, few heuristics arose to help the semi-automated ranking:

- if parts-of-speech don’t match, default to reject
- if multi-word-entry (i.e. the lemma contains whitespace), add weight
- if either word-pair has existing weight or extra restrictions, add weight

An example of resulting translation set presented to linguist might look like Figure 1. The linguist needs to approve the heuristics or modify the results by accepting, discarding or ranking the word pairs, which generally takes no more than few seconds per word for a person knowing the three languages. For the first ranking run it is recommended to leave hard problems unsolved rather than delving into etymologies and background materials.

The structural rules were created by translating and adapting existing rules, in this case we used English to Finnish rules to create English to North Sámi rules. While languages are closely related, the rules need both translating and adapting, for example the possession in Finnish and North Sámi is expressed by a copula verb and possessor in specific

⁹Despite being related language, the overlap in surface words and mutual intelligibility is limited and therefore the dataset is useless for automated testing or training.

¹⁰<https://wiki.apertium.org/wiki/Crossdics>

```

nahkehit.vblex.tv <=> laittaa.vblex <=> put.vblex
nahkehit.vblex.tv <=> laittaa.vblex <=> set.vblex
nahkehit.vblex.tv <=> laittaa.vblex <=> arrange.vblex
nahkehit.vblex.tv <=> laittaa.vblex <=> lay.vblex
nahkehit.vblex.tv <=> laittaa.vblex <=> place.vblex
nahkehit.vblex.tv <=> laittaa.vblex <=> put in.vblex
nahkehit.vblex.tv <=> laittaa.vblex <=> ready.vblex
nahkehit.vblex.tv <=> laittaa.vblex <=> repose.vblex
nahkehit.vblex.tv <=> laittaa.vblex <=> set up.vblex
nahkehit.vblex.tv <=> laittaa.vblex <=> stick.vblex

```

Figure 1: Tidied up example of a semi-automatic dictionary crossing session by a linguist

case, but the copula verb needed to be translated and the case is also different, adessive in Finnish but locative in North Sámi. As another example, the verb paradigms of North Sámi include dual forms while Finnish only has singulars and plurals, on the other hand North Sámi does not usually pro-drop despite having full verb paradigms whereas Finnish generally does pro-drop.

For testing and development of the resulting system, we used the corpus data available as part of the shared task. The corpus-based development in context of Apertium means running the whole corpora through the translator and reading through the automatically translated texts to find which things are translated wrong, and then fixing them. For this experiment, I only reviewed words that didn't get translated due to them being missing from the dictionaries, I mainly focussed on words missing in bilingual dictionary, and went through both high frequency ones and ones that appeared in the several texts I read more thoroughly. For structural transfer errors, I fixed the most obvious ones that appeared in reading the texts.

The overall workflow that I propose can be summarised in a kind of a cascade of two-step processes, where one first generates candidates of linguistic data based on either corpora, curated test cases or programmatic manipulation of existing data, then a linguist goes through each and every output (e.g. translation, grammar rule, ...), either validating them or modifying them.

1. programmatically generate candidate translations by crossing dictionaries and
2. a linguist accepts, discards and ranks them, then
3. programmatically generate candidate rules by copying and replacing and
4. a linguist translates and adapts them, eventually with larger edits if there are divergences

between grammars of languages, then

5. programmatically generate OOV lists from training corpora and
6. a linguist translates them (starting in frequency order when there is no time to finish it all) then
7. * e.g. generate lexical selection rules from corpora (Tyers et al., 2012) and
8. a linguist goes through and accepts or discards them (* we did not reach this phase in the experiment of this paper but it would be one of the likely next steps)
9. ...

The effort needed to semi-automatically validate the bilingual dictionary was roughly a week's work, and the testing and development maybe one more work-day, where the work was carried among other non-related tasks. To compare, usual development time of new language pair is approximately 3 months (e.g. student project, Google Summer of Code) before it starts to be usable for any purpose.

5 Discussion

Both within the Apertium project and within our project for building resources for Sámi languages, we generally only work on systems and applications that are useful to the language user community. The reason why there is no high demand for English to North Sámi machine translations is that practically all speakers are bilingual, with either Finnish or Norwegian as another language. Most will also be fluent in English. Therefore, many of the common use cases of automatic machine translation from majority to minority languages can actually pose threat to the language and the language community:

- instead of hiring professional translators to do translations, companies and governments send out machine translated documents
- amount of machine translated documents in our corpora overtake the amount of human written documents

To exacerbate the problems, even the existing corpora, that have been used to train these machine translators, do not contain similar quantities of grammatically correct texts with current orthography as one is used to from a established long-term majority language point-of-view; this means

that data-driven models are likely to generate more spelling and grammar errors. As such we generally only build machine translators for the direction of from minority language to majority language, this has several use cases that empower the minority language community to use their language while allowing the majority language community to understand, for example in social media. However, for experimentation, such as this article’s subject matter, a language pair that is not highly sought after, is perfectly suitable.

6 Conclusion

With this experiment we found out that crossing dictionaries and translating rules can be a good starting point for a new language pair, where there is available existing resources in related pairs. The work needed to start a new pair in this way is relatively low. The results from the initial work are not yet good enough for actual use in any normal use cases. However, the pair selected for the competition doesn’t have an expected use case so using it for experimentation was a good fit.

Limitations

The experimental translation model that was built in this article does not represent a mature translation system, while care was taken to manually verify the quality of dictionaries and rules. The system provides insight into possibilities and limitations of semi-automatically created rule-based machine translation pairs and will require further work to be used in end-user systems or even further research.

Ethical Considerations

Linguistic work was performed by a fully paid computational linguist, no crowd work was used. Compiling and running a rule-based machine translation uses negligible amount of electricity on a local, low-end laptop and wastes no drinkable water.

Automatic machine translation from English to North Sámi has mainly ethically problematic use cases and may contribute to deterioration in quality of North Sámi corpora. The experiment was performed and system will be published to gain insight in semi-automated system building in the super high resource to low resource setting, it is important that the published open source system won’t be used for corpus creation or automatic translation without professional translator.

Acknowledgments

Some of the computations were performed on resources provided by Sigma2 - the National Infrastructure for High-Performance Computing and Data Storage in Norway.

References

- Lene Antonsen, Trond Trosterud, and Francis M Tyers. 2016. A North Saami to South Saami machine translation prototype. *Northern European Journal of Language Technology*, 4:11–27.
- 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.
- Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).
- Ryan Johnson, Flammie A Pirinen, Tiina Puolakainen, Francis Tyers, Trond Trosterud, and Kevin Unhammer. 2017. North-Sámi to Finnish rule-based machine translation system. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 115–122.
- Tanmai Khanna, Jonathan N Washington, Francis M Tyers, Sevilay Bayatlı, Daniel G Swanson, Tommi A Pirinen, Irene Tang, and Hector Alos i Font. 2021. Recent advances in apertium, a free/open-source rule-based machine translation platform for low-resource languages. *Machine Translation*, 35(4):475–502.
- Krister Lindén, Erik Axelsson, Sam Hardwick, Tommi A Pirinen, and Miikka Silfverberg. 2011. Hfst—framework for compiling and applying morphologies. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 67–85. Springer.
- Flammie A Pirinen. 2019. Apertium-fin-eng-rule-based shallow machine translation for wmt 2019 shared task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 335–341.
- Flammie A Pirinen, Sjur Moshagen, and Katri Hiovain-Asikainen. 2023. Giellalt—a stable infrastructure for Nordic minority languages and beyond. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 643–649.
- SIKOR. 2018. SIKOR uit norgga árktaš universitehta ja norgga sámédikki sámi teakstačokkáldat, veršuvdna 06.11.2018. <http://gtweb.uit.no/korp>. Accessed: 2023-06-12.

- Steinþór Steingrímsson, Iben {Nyholm Debess}, Kimmo Granqvist, Per Langgård, and Trond Trosterud. 2024. *Language Technology for Less-Resourced Languages in the Nordics: Current Developments and Collaborative Opportunities*. Stjórnarráð Íslands, Iceland.
- Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of LREC*, pages 2214–2218.
- Trond Trosterud and Kevin Brubeck Unhammer. 2012. Evaluating north sámi to Norwegian assimilation rbmt. In *Proceedings of the Third International Workshop on Free/Open-Source Rule-Based Machine Translation*, pages 13–26.
- Francis Tyers, Felipe Sánchez-Martínez, and Mikel L Forcada. 2012. Flexible finite-state lexical selection for rule-based machine translation. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 213–220.