UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Maali Tars

# Low-resource Finno-Ugric Neural Machine Translation through Cross-lingual Transfer Learning

Master's Thesis (30 ECTS)

Supervisor: Andre Tättar, MSc

Tartu 2023

## Low-resource Finno-Ugric Neural Machine Translation through Cross-lingual Transfer Learning

**Abstract:**

First high-quality machine translation models were mainly focusing on large languages, such as English and German. Thankfully, the trend has been growing toward helping languages with fewer resources. Most Finno-Ugric languages are low-resource and require the help of different techniques and larger languages for additional information during translation. Recently, multiple big companies have released multilingual pre-trained neural machine translation models that can be adapted to low-resource languages. However, some of the Finno-Ugric languages included in our work were not included in the training of these pre-trained models. Thus, we need to use cross-lingual transfer for fine-tuning the models to our selected languages. In addition, we do data augmentation by back-translation to alleviate the data scarcity issue of low-resource languages. We train multiple different models to determine the best setting for our selected languages and improve over previous results for all language pairs. As a result, we deploy the best model and create the first multilingual NMT system for multiple low-resource Finno-Ugric languages.

**Keywords:**

neural networks, automatic learning, machine translation, language technology, transfer learning

**CERCS:** P176 Artificial Intelligence

## Väheste ressurssidega soome-ugri keelte neuromasintõlge keeltevahelise siirdeõppe abil

**Lühikokkuvõte:**

Esimesed kvaliteetsed masintõlke mudelid treeniti enamasti suurte keelte peal nagu inglise või saksa keel. Hiljutine suundumus on aga arendada ka väikeste ressurssidega keeltele head masintõlget. Enamik soome-ugri keeltest on just nimelt väheste ressurssidega ning vajavad masintõlke treenimiseks erinevaid tehnikaid ning suurte keelte andmeid, millelt tõlkeprotsessis kasulikku infot saada. Selles töös kasutame mitmekeelseid eeltreenitud masintõlke mudeleid, mida saab peenhäälestada soovitud keeltele. Suur osa siin töös osalevatest soome-ugri keeltest ei kaasatud eeltreenitud mudelite treenimisprotsessi, nii et peame kasutama keeltevahelist siirdeõpet. Lisaks sünteesime andmeid juurde tagasitõlkimise abil, et leevendada andmevähesuse probleemi, mis väheste ressurssidega keeltega kaasneb. Me treenime mitmeid mudeleid erineva seadistusega, et saada teada, mis situatsioon valitud soome-ugri keeltele sobib. Näitame, et parandame tulemust kõikidel väikeste soome-ugri keelepaaridel ning mitme soome-ugri keele jaoks avaldame

nende esimese närvivõrkudel põhineva masintõlke mudeli.

**Võtmesõnad:**
tehisnärvivõrgud, tehisõpe, raaltõlge, keeletehnoloogia, siirdeõpe

**CERCS:** P176 Tehisintellekt

# Contents

# 1 Introduction

Neural machine translation has been making leaps in quality in recent years. Multiple publicly available models are exceeding the human level for a selection of large languages. However, for languages with fewer speakers and fewer resources, the quality is still lacking or they are not even included in the training of these large machine translation models (Fan et al., 2021; NLLB Team et al., 2022).

The smaller languages have a low amount of resources available for training a neural machine translation (NMT) network. NMT, however, needs a lot of sample sentence pairs of a language pair to be able to learn a good representation of it. This is why a lot of very small languages are not included in the training process.

High-resource languages are often leveraged to help low-resource languages achieve a reasonable translation quality because languages all share some parameters in sentence structure, orthography, and lexical attributes. It has been shown that using related high-resource languages can cause an even bigger positive effect on the quality of low-resource translation (Gu et al., 2018). The patterns that have been learned from huge amounts of high-resource language data can be transferred in part onto languages that have smaller amounts of data available for training. This phenomenon is one of the reasons why nowadays most models are multilingual, with languages sharing the parameters of the model.

In our work, we make use of large multilingual pre-trained machine translation models, that have been trained by large companies, specifically the M2M-100 models from Facebook AI (Meta AI). We continue training the models on our selected low-resource languages from the Finno-Ugric language family. Our work builds on the previous effort to train a translation system between Estonian (`et`), Finnish (`fi`), Võro (`vro`), North Sami (`sme`), and South Sami (`sma`) (Tars et al., 2021). In this work, we acquire data for four new low-resource Finno-Ugric languages: Inari Sami (`smn`), Lule Sami (`smj`), Skolt Sami (`sms`), and Livonian (`liv`). All in all, we are working with 7 low-resource Finno-Ugric languages and 5 high-resource languages: Estonian, Finnish, Latvian (`lv`), Norwegian (`no`), and English (en). Some of the high-resource languages are in the same language family while others are either geographically close to the areas where the low-resource languages are spoken or just have available parallel data paired with one of the low-resource languages.

All of the low-resource languages mentioned were not included in the original training of the M2M-100 models. We perform cross-lingual transfer learning in order to leverage information from large languages. Additionally, we produce synthetic data from monolingual data by back-translation and include that in our experiments to boost the performance. As M2M-100 is a resource-consuming system to fine-tune, we explore the capabilities of two different-sized versions of the model to find out whether the capacity of the smaller model is enough or the larger version is still needed to learn a better representation.

The main contributions of this work are the following:

1. Gathering data and creating a new benchmark for most of the low-resource Finno-Ugric languages.
2. Showing how to do cross-lingual transfer to unseen Finno-Ugric languages on the pre-trained M2M-100 models.
3. Analysis of the best data and model size scenarios for the selected Finno-Ugric languages.
4. Reporting state-of-the-art neural machine translation results for all of the low-resource Finno-Ugric languages on multiple benchmarks.
5. Publishing the first NMT system[1] for multiple low-resource Finno-Ugric language pairs included in our work.

The following work is based on two previously published articles. In Tars et al. (2022b), we explain how to do cross-lingual transfer on the M2M-100 model to low-resource Finno-Ugric languages, as well as perform model size comparison and find out which dataset settings are the most suitable for our selected low-resource languages. In Tars et al. (2022a), we perform back-translation iterations but in our experiments, we mainly focus on Livonian because the paper was a part of the WMT22 General Machine Translation task Kocmi et al. (2022) where English-Livonian language pair was one of the focus points.

In Section 2 we give some background and explain some terminology. In Section 3 we mention important publications that are related to our current work. Section 4 describes the data, the gathering, and pre-processing steps. Section 5 gives an overview of our used methods. Sections 6 and 7 describe the experiment setups and analyze the quality and results of our trained models.

[1] https://neurotolge.ee/

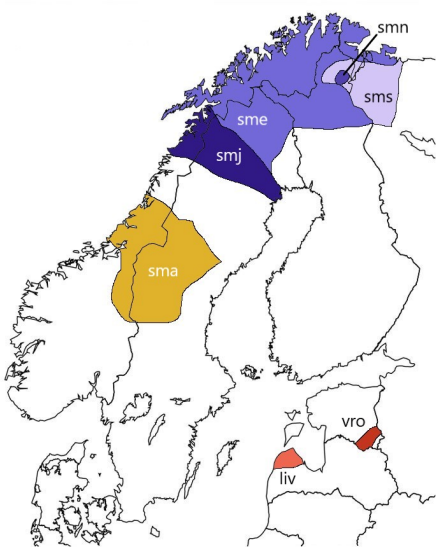



Figure 1. Map of low-resource Finno-Ugric languages included in our work: Võro (vro), Livonian (liv), North Sami (sme), South Sami (sma), Inari Sami (smn), Lule Sami (smj), Skolt Sami (sms).

# 2 Background

## 2.1 Finno-Ugric Languages and Low-resource Setting

The Finno-Ugric language family is small compared to other language families. About 23-25 million people speak Finno-Ugric languages in the world, with Hungarian, Finnish, and Estonian making up the majority of the speakers. The number of speakers of a language does not necessarily correspond to the amount of available data resources. For example, according to the OPUS corpora (Tiedemann, 2012) English-Estonian has twice as much public translation data as English-Hindi, while Hindi has ~425 million speakers[2] and Estonian has only ~900 000 speakers[3]. There are however dozens of more minor languages in the Finno-Ugric language family that do not get enough attention when it comes to advancing language technology.

Digital material which is needed to develop natural language processing (NLP) applications is fairly scarce for minor languages compared to the high-resource languages mentioned before. Some of the languages do not have a unified orthography among all the different dialects spoken inside the language which makes it difficult to gather a normalized corpus and train for all the uncommon symbols that might be present in the data. In this work, we attempt multiple techniques frequently used in low-resource

[2]https://www.britannica.com/topic/Hindi-language
[3]https://andmed.stat.ee/en/stat

settings, countering the lack of parallel data between language pairs.

The low-resource languages that are present in the experiments of this work are the following: Võro (vro), Livonian (liv), North Sami (sme), South Sami (sma), Inari Sami (smn), Lule Sami (smj), Skolt Sami (sms) (see Figure 1). In addition to Finno-Ugric languages, we include multiple other high-resource languages in the training process such as Latvian, Norwegian, and English. The reasons for adding an unrelated high-resource language differ for each of them but overall the main factor is the existence of parallel data between one of the mentioned high-resource languages and a low-resource Finno-Ugric language. Latvian has parallel data with Livonian because Livonian was mainly spoken in the areas of modern-day Latvia. Similarly, Norwegian has parallel data with a number of Sami languages, because Sami languages are spoken in Norwegian territories. The public parallel data usually consists of either news or legislation documents to help native low-resource language speakers stay informed about the country they are living in. Latvian and Norwegian have also influenced the low-resource Finno-Ugric languages grammatically as well as introducing new, more modern words and phrases into the languages.

## 2.2 Transformers

The dominant architecture across NLP tasks at the moment is the Transformer (Vaswani et al., 2017) (see Figure 2). This is also the case for the neural machine translation task. Transformer is a neural network model consisting of multiple encoder-decoder layers. It takes a sequence of tokens as an input and outputs a transformed sequence of tokens. The mechanism called self-attention is the property that makes Transformers so efficient and prevalent across different tasks.

Attention in the context of sequence-to-sequence tasks means learning which input tokens should focus on which output tokens during translation. Self-attention takes attention to another dimension. Here we learn to predict which tokens of the sequence should focus on which other tokens in the same sequence, hence the "self" part. Self-attention also reduces the problem of processing long sequences that in the past have been avoided because of the use of recurrent neural networks that required the information to flow linearly through the sequence.

The Transformer architecture allows the model to work in a much more parallelized fashion, compared to recurrent or convolutional networks, with multi-head attention compartments computing different attention representations at the same time. Since self-attention is a process with quadratic complexity, we benefit greatly if the layered computations of self-attention can be done in parallel, reducing the total time of training significantly compared to if we performed the computations in a linear fashion.

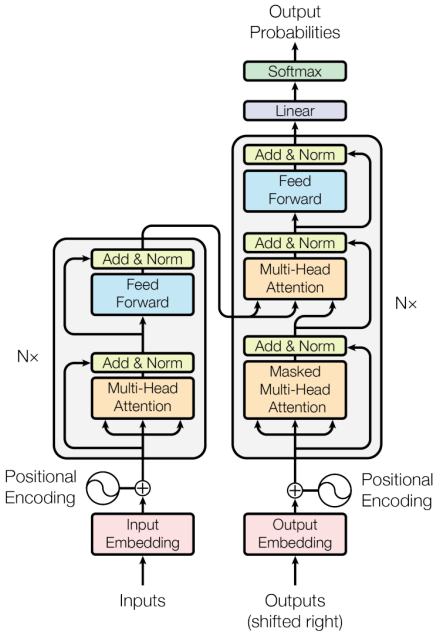



Figure 2. Transformer architecture (Vaswani et al., 2017).

## 2.3 Multilingual Neural Machine Translation

Multilingual neural machine translation (MNMT) is a method of automatic translation where the translation system can translate between either 1) one source language and multiple target languages, 2) multiple source languages and one target language, or 3) multiple source languages and multiple target languages. With the rise of neural networks in the machine learning world, the most efficient way of deploying real-world models is to train one multilingual model that can cover multiple language pairs. Instead of training one model for each language pair or translation direction, which might have gotten the best result at some point with recurrent networks, with MNMT we train only one model. Overall it proves to be more cost-efficient and Transformer models also yield better translation quality utilizing the power of transfer learning between languages.

## 2.4 Byte-pair Encoding

Before the sentence pairs of training data are ready to be input into the model, we tokenize them using the byte-pair encoding (BPE) algorithm (Gage, 1994; Sennrich et al., 2016b). Tokenization is necessary, because languages have an infinite vocabulary, but NMT models need a finite vocabulary, and the smaller the vocabulary the more efficient the model is. We could add all the words from the training set into the vocabulary but then the vocabulary would be very large and some of the words that come up during test

time might not be present in the vocabulary. A better idea is to divide words into smaller pieces and compile the vocabulary out of those word pieces which can be glued together by the model at test time to produce the output.

The BPE algorithm divides words into characters and iteratively merges the most frequent character combinations together until the vocabulary reaches a pre-determined capacity of character combinations. The idea behind it is that very common words will be intact in the final vocabulary, but infrequent words can be combined from frequent subword pieces that occurred in the split training data. After the creation of the vocabulary, the training data can be tokenized according to the available subword pieces existing in the final vocabulary.

## 2.5 Cross-lingual Transfer Learning

In transfer learning (Olivas et al., 2009), the information learned during one task can be transferred to another task or domain. It is common, that the tasks are somewhat related to each other but it is not a strict requirement. In the process of transfer learning, the weights of the original model are tuned to the new task at hand while some of the weights that are deemed useful during training remain the same. If there was already a suitable model available, then this technique could reduce training time greatly for producing a model for the new task.

In this work, we utilize cross-lingual transfer learning. Cross-lingual transfer learning is a special case of transfer learning in NLP, where the original model has not seen the language(s) during training time, so it is seeing the new language data during the transfer learning for the first time. We take publicly available pre-trained multilingual translation models and introduce them to the multiple low-resource Finno-Ugric languages that were not present in the training process of the pre-trained models.

## 2.6 Back-translation

Low-resource language pairs by definition have significantly lower amounts of parallel training data than high-resource language pairs. However, low-resource languages often have available monolingual data, meaning just sentences in one language. These kinds of sources can be used for augmenting the parallel data with a method called back-translation (Sennrich et al., 2016a). There are multiple ways and scenarios for performing back-translation. In our case, we need a pre-trained machine translation model which has been tuned to small amounts of parallel data from the low-resource language pairs. Then we translate the monolingual data into languages that we wish would have more parallel data. After the translation process, we switch the source and target languages in the pair and add the produced translation samples to the original training data. The switch is what gives it the name "back-translation". It is more helpful if the clean and mostly correct original monolingual data is on the decoder side. This

enables the decoder to learn on clean data and thus learn to predict clean and correct sentences. The synthetic sentences produced during the back-translation process are oftentimes faulty to various degrees, depending on the amount of parallel data that the original model was trained on. This means that the encoder could learn to be more robust across input sentences if it sees some noise among the data but the decoder still tries to predict correct sentences.

## 2.7 Automatic Evaluation Metrics

There are multiple different automatic evaluation metrics to determine the quality of machine translation models. The most used in the literature for machine translation has been the BLEU (bilingual evaluation understudy) metric (Papineni et al., 2002) which compares the reference translation (human translation) to the translation produced by a model. The score is put together by comparing word n-grams (different length consecutive word sequences) of these two sentences and seeing how many of them match each other. During the years that BLEU has been dominant, researchers have noticed multiple shortcomings with the metric, especially when it comes to morphologically rich languages that Finno-Ugric languages definitely are. This is why just recently it has been recommended to take other metrics next to BLEU and not give too much importance to only one score. chrF (character n-gram F-score) is a metric designed with keeping morphologically rich languages in mind and it has been shown to have better correlation with human evaluations than BLEU (Popović, 2015, 2016, 2017). The main difference from the BLEU metric is that instead of comparing word n-grams between the reference and the model's translation, the metric compares character n-grams. In this work, we report results on BLEU and a variation of chrF (chrF++ – adds word n-grams to the calculation of the score).

## 2.8 Pre-trained Models and M2M-100

We utilize pre-trained multilingual neural machine translation models in all of our experiments. They are usually trained by big corporations (Google, Meta) that have access to a lot of computing resources and training data. As a result of the companies publishing these models and providing measures to repurpose the trained models for many different tasks, for a good machine translation model, one just needs to fine-tune the model to their desired dataset. This saves on computational costs and enables everyone to help advance machine translation and other language technology tools.

In this work, we use the M2M-100 model (Fan et al., 2021) pre-trained by Meta AI (see Figure 3). The architecture is based on the Transformer (Vaswani et al., 2017) with size modifications. M2M-100 main aim is to lose the English-centric bias in multilingual models. For creating a good non-English-centric multilingual model, a larger, more diverse dataset and a larger model are needed. They first create an efficient data mining

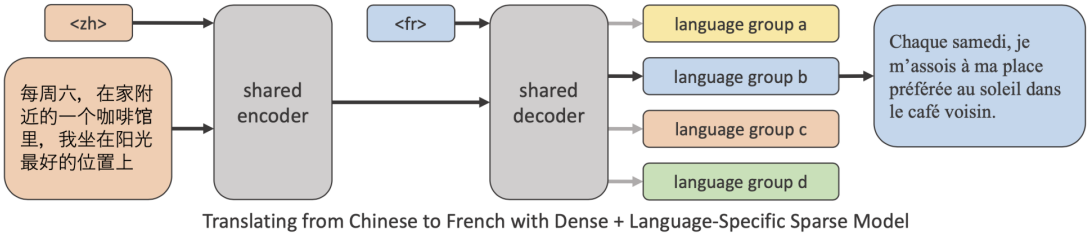



Figure 3. M2M-100 proposed architecture (Fan et al., 2021). The published pre-trained models that we use as starting points did not use language-specific layers.

method to conquer the dataset challenge of collecting sentence pairs between all of the 100 languages included in the model. Afterward, back-translation is also used as a standard technique to augment the data.

In order to learn the information provided by the new, 7.5 billion sentence big corpora mined, the model size also needs to increase. Here the creators of the model employ different techniques such as scaling the number of parameters and training the model by re-routing separated language-group-specific parameters. They publish models that are great baseline models for fine-tuning with a small amount of data because there is so much pre-existing knowledge already encoded into the parameters. Multiple language pairs can therefore achieve higher translation quality while spending less computational resources. The published models that we use in our experiments, however, do not use any language-specific layers.

The M2M-100 improves on multiple non-English translation directions, but the model itself is quite large compared to training a bilingual model or a multilingual model with a couple of languages. Fine-tuning it requires memory capacity to upload it to the training environment as well as updating the hundreds of millions of parameters. Another issue is the size of the vocabulary. To cover all 100 languages the vocabulary is relatively huge compared to smaller multilingual models containing about ~128 000 tokens (usually ~32 000 tokens for bilingual models), which also increases the training time. For our scenario, we are only interested in Finno-Ugric languages plus a couple of other high-resource languages. Thus, a large part of the vocabulary is unused in our case, for example, everything outside of the Latin alphabet is mostly unused (Arabic, Chinese).

# 3 Related Work

## 3.1 Low-resource Machine Translation

The authors of Gu et al. (2018) explore multilingual translation from multiple source languages into English. They propose new modules to the usual multilingual NMT approach of just sharing the encoder between multiple languages. One of the additional components is Universal Lexical Representation designed to help semantically similar tokens become closer to each other in the embedding space. Another added part is using Mixture of Language Experts (MoE) modules for high-resource languages to weigh each of their contributions to the translation at a certain time. In their experiments, they also experiment with different sets of high-resource languages added to the multilingual model and conclude that related high-resource languages are able to give more knowledge to the low-resource language translation process than unrelated ones. Overall, their methods offer a maximum 5 BLEU point increase from the usual multilingual NMT model in the low-resource English-Romanian setting.

In Sennrich and Zhang (2019), the authors try to find the best conditions for training an NMT model for a low-resource language pair in a bilingual setting. In the process, they also prove that with the right approach, the NMT model outperforms a phrase-based statistical machine translation model. One interesting finding during their experiments was that for low-resource settings, smaller training data batches are better, although the tendency in NMT training is to use as big of a batch size as can be used. It is also worth noting that they use Korean-English as a low-resource language although it has 90 000 parallel sentences whereas our datasets have multiple language pairs with less than 1000 training samples.

The experiments performed by Kocmi and Bojar (2018) are one of the most similar to the transfer learning approach that we use in this work. They train a bilingual model on a high-resource language pair and then train that model further to a low-resource language pair, with the vocabulary shared across both language pairs. We, however, are working with multilingual models, handling multiple language pairs at once, but also sharing the vocabulary between all of the language pairs. Interestingly they find that even non-related high-resource language pair helps to get better quality in this approach, which is somewhat in conflict with the claims made by Gu et al. (2018). They also include Estonian in their experiments as a low-resource language with 800 000 sentence pairs in the English-Estonian pair.

In Goyal et al. (2020), they experiment with Indian low-resource languages, utilizing multilingual transfer learning, as well as Unified Transliteration and subword segmentation (BPE algorithm that is also used in our work). Transfer learning here can happen from a related high- or low-resource language. Unified Transliteration involves unifying the orthographies of related languages. The combination of the methods described, gives them an average of 5 BLEU point gain over all the Indian-English language pairs, which

they consider a notable achievement.

In addition to multiple papers working on improving low-resource language machine translation, the topic has become more and more popular with multiple surveys emerging about the task (Haddow et al., 2022; Wang et al., 2021). The surveys mention using premade language models by leveraging data from high-resource languages, back-translation through utilizing larger amounts of monolingual data, as well as multiple forms of transfer learning. All of these methods are also in effect in our work.

## 3.2 Machine Translation for Finno-Ugric Languages

Earlier efforts of machine translation models were mainly bilingual. Some of the low-resource languages in our work have also been trained in this setting, most of them by ourselves in our previous work (Tars et al., 2021). There have also been models for single language pairs such as a translation system developed by Giellatekno at The Arctic University of Norway (UiT), translating between North, Inari, and Lule Sami[4]. For the Livonian language, Rikters et al. (2022) analyzed different settings to train Livonian in, comparing multilingual model where high-resource languages are Estonian, Latvian, and English to bilingual baseline models. The Livonian dataset that was created in the previously mentioned paper was also a part of the 2022 edition of the WMT competition (Kocmi et al., 2022). During the competition, multiple teams of machine translation developers submitted their best system for English-Livonian translation. Our system placed first in the Livonian-English direction and second in the opposite direction according to automatic metrics (COMET and chrF) on the WMT22 official test set (Tars et al., 2022a). Human evaluation results for Livonian-English placed us first, sharing the top place with four other systems. In the English-Livonian direction, we shared third place with another system.

In Kocmi and Bojar (2018), one of the low-resource languages is Estonian and they show multiple times that during transfer learning, the high-resource model does not need to be trained on a related language pair to the low-resource language pair, with English-Czech having more impact on English-Estonian than when English-Finnish is the parent model.

Rikters et al. (2018) include Estonian in their experiments as a low-resource language and also consider the aspects of NMT training with a morphologically rich language like Estonian and all of the Finno-Ugric languages. In their analysis, they note that comparing different NMT architectures, low-resource language pairs have more to gain when being trained on a Transformer-style model and in a multilingual setting.

[4]https://gtweb.uit.no/mt

## 3.3 Cross-lingual Transfer Learning

Transfer learning has been shown to be beneficial in low-resource settings, with Zoph et al. (2016) being one of the first to employ the technique of transferring knowledge from high-resource to low-resource languages in the pre-Transformer era. Their method is to train a bilingual model on a high-resource language pair, then freeze some of the parameters of the trained model and continue to tune others on a low-resource language pair. The authors reason that transfer learning gets its advantage from having a strong bilingual base model prior to fine-tuning some of the parameters on a smaller dataset. Building on the work of Zoph et al. (2016), the authors in Nguyen and Chiang (2017) replace the high-resource language pair with a related low-resource language pair and share the vocabulary between the language pairs under question. They mark that since the language pairs are related, they share similar language attributes and similar word pieces in their vocabulary which can be exploited during the transfer learning.

The closest to our experiment is a work by Kim et al. (2019), where they employ a pre-trained NMT model to perform transfer learning to a low-resource language that did not share the vocabulary with the pre-trained model. They solve the vocabulary mismatch by learning a word-embedding mapping between the pre-trained model embeddings and the embeddings learned from the monolingual data of the new language. Then they replace the embeddings of the pre-trained model with the newly learned ones. However, they utilize techniques such as inserting noise into data and producing synthetic parallel data without back-translation which are out of the scope of our work.

## 3.4 Pre-trained Machine Translation Models

Multilingual pre-trained models that have a hundred or more languages are trained by big corporations like Google or Meta. One of these earlier examples is a work done by Google where they trained an English-centric multilingual model with 102 languages. In their experiments, they notice that a setting with 58 and more languages is especially beneficial to low-resource language pairs (Aharoni et al., 2019). Following this work, Zhang et al. (2020) notice that for some language pairs bilingual models still outperform the multilingual model and go on to suggest solutions for this issue. They mainly see the cause of poor performance in the capacity of the multilingual model itself and suggest a deeper Transformer architecture as a fix which in turn benefits the low-resource language pairs in the model.

Since the year 2020, there has been a steady trend of training more and more massive multilingual pre-trained machine translation models which are then made available to the public for free usage. One of the models that started this trend is the M2M-100 model developed by Facebook AI (Meta AI). M2M-100 aims to move away from English-centric models, including 100 languages and all of the translation directions between them (9900 translation directions). One of the main contributions of this work is creating

a large dataset between non-English-centric language pairs having 7.5 billion sentences in the end. As the number of data samples increases, the capacity of the model also has to grow. They propose scaling techniques and language-specific modules that interact with each other during training. In the end, they release multiple different versions of their models that get more than 10 BLEU points better scores than an English-centric baseline for non-English-centric language pairs (Fan et al., 2021).

After the M2M-100, Google published their work modifying the T5 model into a multilingual text-to-text transfer Transformer (mT5) for 101 languages. This model, however, is not trained specifically for the machine translation task, but rather for all the tasks where one is required to generate text conditioned on some input text (Xue et al., 2021).

As a follow-up to the M2M-100 project, Meta AI released their NLLB (No Language Left Behind) model (NLLB Team et al., 2022) in 2022, scaling up to 200 languages. Though this model has twice as many languages, it still lacks any of the low-resource Finno-Ugric languages that we work with. They enhance their data-mining techniques to successfully gather data for all the translation directions between 200 languages as well as develop a more intricate Mixture of Language Experts system to make up for the increase of training data when trying to maintain the same model capacity. In our work, we use M2M-100 because NLLB is quite recent and the released pre-trained models have somewhat more parameters which requires longer training time and more resources.

## 3.5 Back-translation for Low-resource Setting

The authors in Sennrich et al. (2016a) try utilizing monolingual data in the target side of the parallel data during training for NMT models. To fill the source side they have two strategies: using a dummy sentence or using a synthetic sentence produced by some base translation model. In their experiments, they realize that the synthetic translation is more of a benefit when the encoder receives that instead of a more primitive dummy sentence.

Following this work, Hoang et al. (2018) demonstrate that the quality of the base model which is used to produce the synthetic source sentences matters to the final translation quality of the model trained on parallel data that includes back-translation data. In addition, they introduce the idea of iterative back-translation. The reasoning behind this is that if the synthetic data paired with the monolingual target data makes the models better, then the developed model will be better at producing the synthetic side another time around. They show that they see gains with only 1 or 2 additional iterations, especially in the low-resource settings. It is worth noting that both Sennrich et al. (2016a) and Hoang et al. (2018) work with bilingual models, not multilingual models, but their techniques can be easily adapted into the multilingual setting also.

In Gu et al. (2018) one of the components they added to their own ideas was back-translation using a multilingual NMT system to produce synthetic translations. This

showed to give further improvements upon their own efforts to develop low-resource NMT quality.

There are multiple different aspects of producing synthetic parallel data from monolingual sentences. In Edunov et al. (2018), the authors mainly explore the different algorithms or strategies for creating the synthetic source sentences and conclude that instead of the usual beam and greedy search, sampling or adding some noise to the beam search might be even more beneficial. This keeps the synthetic outputs more random instead of always choosing the best possible translation produced. Burchell et al. (2022) investigate the diversity of synthetic data in further detail, also noting that the usual methods of choosing the most likely translation is not a good strategy and move to suggest diversifying the synthetic bitext lexically as well as syntactically.

# 4 Data

## 4.1 Gathering and Description

Low-resource language data collection took place in multiple parts. Part of the data is from our previous work (Tars et al., 2021), namely the data for Võro, North Sami, and South Sami languages. The data for other Sami languages was gathered by us from publicly available sources[5] of The Arctic University of Norway (UiT) as is described in Tars et al. (2022b). UiT itself compiled the parallel data and published it for free downloading. Monolingual data, however, needed to be gathered semi-manually from a list of documents[6] that were in various formats (TXT, XML, HTML). Livonian data was provided by the WMT22 workshop competition on English-Livonian machine translation (Kocmi et al., 2022). Parallel and monolingual data for Livonian used in the competition is also published in the OPUS corpora (Tiedemann, 2012) in a collection under the name of "liv4ever"[7] (Rikters et al., 2022).

Table 1. Monolingual data amounts per low-resource language. `vro` - Võro, `sma` - South Sami, `sme` - North Sami, `sms` - Skolt Sami, `smn` - Inari Sami, `smj` - Lule Sami, `liv` - Livonian. The table is adapted from Tars et al. (2022b).

| language | vro | sma | sme | sms | smn | smj | liv |
|---|---|---|---|---|---|---|---|
| nr of segments | 162 807 | 55 088 | 33 964 | 76 685 | 122 916 | 128 180 | 40 329 |

Parallel data for language pairs between high- and medium-resource languages (Estonian, Finnish, Norwegian, Latvian, English) was sampled from the OPUS corpus. 20 000 sentence pairs for each language pair were sampled and added to the dataset to keep the amount of high-resource data in balance with the amount of low-resource data. Monolingual data for the high-resourced languages originates from WMT news crawl corpora[8] (Kocmi et al., 2022) with 500 000 sentences randomly sampled for each language. The number of high-resource monolingual sentences was chosen according to the estimation of our time and resource capabilities for translating each high-resource language into each low-resource language with which they had original parallel data. The amounts of monolingual data for the low-resource languages can be seen in detail in Table 1. Here we notice, that when comparing to Table 2, for multiple languages there is more monolingual data than parallel data in their respective language pairs. This is in coherence with the usual tendency of low-resource languages to be richer in monolingual data.

[5] https://giellalt.uit.no/tm/TranslationMemory.html
[6] https://gtsvn.uit.no/freecorpus/orig/
[7] https://opus.nlpl.eu/liv4ever.php
[8] https://data.statmt.org/news-crawl/

## 4.2 Pre-processing

The pre-processing of the whole dataset included detokenization, punctuation normalization, and filtering. The detokenization and punctuation normalization was performed ahead of filtering with the help of Moses scripts[9]. Here, detokenization is defined as a process, where unnecessary whitespace between a word and punctuation mark is removed. Punctuation was normalized following a set of pre-determined regex transformations. We modified the original Moses normalization script to be more suitable to the Finno-Ugric languages by removing and adding some regex rules[10].

Monolingual data was not filtered because in large part it was either collected semi-manually (low-resource) or sampled from the WMT news crawl corpus (Kocmi et al., 2022) which has already undergone quality control. Parallel data, however, went through a series of filtering heuristics calibrated to the data at hand. We use a pre-processing tool OpusFilter (Aulamo et al., 2020). It is easy to use with predetermined filter options and adjustable thresholds for each filter. Before filtering, the whitespace across the parallel data was normalized. The filtering steps that we used were the following:

- maximum segment length: 1000 characters or 400 words
- maximum word length: 50 characters
- source and target segment length difference: max 3 times
- ratio of numeric characters in the segment: 0.5 or less
- ratio of alphabetic characters in the Latin alphabet: 1
- ratio of alphabetic characters in the segment: 0.75 or more
- ratio of similar numerals between parallel segments, with zeros removed: 0.5 or more

After filtering, the number of parallel data per low-resource language pair varied strongly. For example, Norwegian-North Sami direction is at one extremum with 200 000 sentence pairs, whereas English-Livonian only has about 300 sentence pairs of training data. All of the amounts before and after the filtering steps can be found in Table 2.

## 4.3 Evaluation and Validation Data

We evaluate our models on a couple of different test sets, containing data from different domains and varying in quality of the sentences in the test set. One test set that we

[9]https://github.com/moses-smt/mosesdecoder

[10]https://github.com/Project-MTee/model_training/blob/main/normalization.py

Table 2. Parallel data amounts before and after filtering (in sentence pairs) per language pair. The table is adapted from Tars et al. (2022b).

| lang-pair | raw | filtered |
|---|---|---|
| et-vro | 31 551 | 29 775 |
| fi-sme | 77 710 | 62 837 |
| fi-sma | 2913 | 2766 |
| fi-smn | 10 639 | 9459 |
| fi-sms | 5769 | 2708 |
| no-sma | 17 388 | 15 702 |
| no-sme | 241 598 | 195 970 |
| no-smj | 12 400 | 11 627 |
| sme-sma | 21 993 | 19 963 |
| sme-smj | 16 440 | 14 985 |
| sme-smn | 934 | 894 |
| en-liv | 617 | 280 |
| et-liv | 14 261 | 12 887 |
| lv-liv | 11 732 | 10 763 |

Table 3. Evaluation and validation datasets (in sentence pairs) per language pair. The table is adapted from Tars et al. (2022b)

| lang-pair | test | valid |
|---|---|---|
| et-vro | 500 | 200 |
| fi-sme | 500 | 200 |
| fi-sma | 500 | 200 |
| fi-smn | 500 | 200 |
| fi-sms | 500 | 200 |
| no-sma | 500 | 200 |
| no-sme | 500 | 200 |
| no-smj | 500 | 200 |
| sme-sma | 500 | 200 |
| sme-smj | 500 | 200 |
| sme-smn | 500 | 200 |
| en-liv | 856 | 586 |
| et-liv | 856 | 586 |
| lv-liv | 856 | 586 |

use to compare to previous results is from our previous work on Finno-Ugric NMT systems (Tars et al., 2021). This test data however did not contain any of the newly added languages (Inari Sami, Lule Sami, Skolt Sami, Livonian), had some quality issues, as well as contained some overlapping sentences with the training data. As a result of these circumstances, we decided to compile a new held-out test set from the filtered parallel training data[11] in this work. For some of the language pairs, this test set is the first machine translation benchmark to the best of our knowledge.

With the exception of Livonian, the new held-out test set contains 500 and the validation set contains 200 sentences per language pair. The "liv4ever" dataset had complementary test and validation sets already available. The exact amounts of test and validation data for each language pair can be found in Table 3.

Recently, we have managed to translate parts of the FLORES-200 benchmark (Goyal et al., 2022; NLLB Team et al., 2022) into Livonian (Yankovskaya et al., 2023). FLORES is a collection of the same sentences translated into 200 languages from English, consisting of 3000 sentences from the Wikimedia[12] corpora collection. Having the same sentences as a test set across all languages allows for a fair comparison between machine translation models for all 200 languages. We have 250 sentences for Livonian translated

[11] https://huggingface.co/datasets/tartuNLP/finno-ugric-benchmark

[12] https://www.wikimedia.org/

from the FLORES-200 set[13]. This should give a more unbiased evaluation of the quality of our models than our own held-out test set.

In order to report the most trustworthy results, we remove the test sentences from the training set as best as possible. We use some usual heuristics for detecting overlap between the training and test sets, by comparing pairs of sentences with punctuation and whitespace removed.

[13] https://huggingface.co/datasets/tartuNLP/smugri-flores-testset

# 5 Methods

## 5.1 Cross-lingual Transfer Learning

One of the methods that we use to compensate for the lack of training data for low-resource languages is cross-lingual transfer learning. We take a pre-trained multilingual machine translation model as a starting point and continue training it on our Finno-Ugric language pairs. All of the low-resource languages in this work were not included in the initial training of the pre-trained model. This means that there are no parameters in the model trained for the new language pairs and no knowledge of these languages. However, as the pre-trained model is trained on 100 languages, a number of them are related to the low-resource Finno-Ugric languages or share some linguistic patterns with the new language pairs that are going to be introduced during the transfer learning stage. The knowledge encoded into the model parameters during pre-training is therefore partly transferable across languages and can be adapted to the new language pairs.

One of the problems with this method, however, is catastrophic forgetting, which means that we overwrite some parameters trained for a particular language (e.g. a high-resource language pair) with information from a new language that we are tuning the model for. Since our aim is to focus on Finno-Ugric language pairs only, we do not apply any complicated measures to hold off the forgetting process. However, to mitigate it for translation directions that interest us (Estonian-English, Finnish-English, etc.) we include some data for the language pairs between two high-resource languages and mix it with the other parallel training data. This allows the model to see those language pairs during training and will not completely forget them.

## 5.2 Vocabulary and Embedding Matrix Enhancement

Since all of the low-resource languages were unknown to the pre-trained model before the transfer learning process, the model has no knowledge of the language codes and will not recognize the new characters or symbols that were not present in the original languages. To get around this obstacle, we create scripts[14] to expand the vocabulary file of the model as well as increase the embedding matrix size of the encoder to include the codes and symbols of the new languages. The indexes of the new symbols are randomly initialized.

We use the Huggingface implementation of the pre-trained M2M-100 model and its fine-tuning framework[15]. The implementation requires each source sentence and each target sentence to precede with the language ID token (language code). We adapt our script to input JSON files, in which the sample sentence pairs are in the format of the

[14]https://github.com/TartuNLP/m2m-100-finetune

[15]https://huggingface.co/docs/transformers/model_doc/m2m_100

following example: {"translation": {"et": "Aga ometi olime nii lähedal.", "vro": "Aga ummõhtõ ollimi nii lähkün."}}.

## 5.3 Back-translation

Another technique for reducing the data scarcity of low-resource languages is producing synthetic sentence pairs that act as proxy parallel data. Synthetic data is usually produced by back-translation (Sennrich et al., 2016a). In this work, for back-translation, we take a machine translation model that has been previously trained on some amounts of parallel data for a certain language pair and use it to translate monolingual data in a translation direction known to the base model.

As a result of this process, one side of the parallel data is the original monolingual text and should be grammatically correct. The other side is synthetically produced by the base model. The artificial parallel data is then turned around so that the synthetic sentence becomes the source sentence and the original sentence becomes the target sentence. This way the decoder learns on data that we are certain is clean and probably with very few mistakes (monolingual data) and thus the model makes fewer mistakes during the decoding of the input sentence into the output sentence. The new synthetic parallel data is then added to the original parallel data and the models are trained again.

## 5.4 SentencePiece

The vocabulary of the M2M-100 in the Huggingface implementation is created by the popular framework SentencePiece (Kudo and Richardson, 2018) which implements the byte-pair encoding mechanism. The pre-determined vocabulary has ~128 000 most frequent word pieces obtained from the training data of the original M2M-100 model. The vocabulary was created in a balanced manner between all of the languages. Huggingface's M2M100Tokenizer which uses the SentencePiece framework was used to encode the sentences into tokens. M2M100Tokenizer was adapted to recognize new symbols from previously unseen languages.

## 5.5 Supporting Applications

Basic free Grammarly[16] was used to check grammatical correctness throughout the written part of this work.

[16]https://app.grammarly.com/

# 6 Experiments

## 6.1 Model Size Comparison

M2M-100 has multiple different versions of their model by parameter amount. We compare the 418 million parameter (418M) and 1.2 billion parameter (1.2B) size models. One of our research questions was to find out whether the smaller and the bigger model level out at some point in translation quality during the training time. There is a big difference in resource consumption between the two model sizes when training them and also in the test mode (inference process). For example, in our setting, the 1.2B model consumes 5710 MB of GPU memory at the start of training whereas the 418M model consumes 2816 MB of GPU memory. If at some point the smaller, 418 million parameter model catches up to the larger, 1.2 billion parameter one, it would be optimal to do further experiments with the smaller model and at the end of training to deploy the smaller model, because it will be faster to load and faster while translating the user's input.

One of the reasons behind the model size comparison experiment was that the dataset we tune the models on is relatively small compared to the amount of data that the pre-trained model was trained with. Thus, we hypothesize that maybe our relatively tiny dataset does not need so many parameters to learn our language pairs.

## 6.2 Effect of English

As a separate experiment branch, we test whether removing language pairs where one side is English affects the quality of translation. The reasoning for this experiment is that English is not a related language to any of the Finno-Ugric low-resource languages and it is not a national language in any country where the low-resource Finno-Ugric languages are spoken. Specifically, we remove the high-resource language pairs involving English (`en-fi`, `en-no`, `en-et`, `en-lv`) and one low-resource language pair data (`en-liv`) which only had about 300 training sentences as a part of the original parallel data.

## 6.3 Related Language Groups

Another hypothesis that we raise is that training related language pairs, rather than unrelated language pairs, in the same model, helps to transfer more relevant and accurate translation pattern information between languages during training. Since some of the low-resource languages can differ from each other across the Finno-Ugric language family notably, they can be divided into even smaller language groups. In turn, we set up multiple datasets with different combinations of smaller language groups.

We separate Livonian, Võro, and Sami languages into three branches. For the separate Livonian experiment, we have language pairs where Livonian is paired with multiple

high-resource languages (English, Latvian, Estonian). For the Võro experiment, we only have Estonian-Võro translation directions to tune the pre-trained model. Sami languages are all paired with either Finnish, Norwegian, both or to each other (language pairs between `fi-sm*`, `no-sm*`, `sm*-sm*`).

## 6.4 Back-translation Iterations

There were in total two back-translation iterations in the experiments. In the first iteration, the monolingual data was translated with the model that performed best overall in the initial tuning with the original parallel data. After completing the first back-translation iteration, the different language group experiments are repeated, but this time the turned-around synthetic parallel data was added to the original parallel data. This means that there was a lot more training data and thus it takes longer for the model to reach one epoch (go through all of the training samples once).

From the experiments trained with the first synthetic dataset, we select the best-performing model. That model is then used for the second iteration of back-translation for the same monolingual data used in the first back-translation iteration. This time, however, the base model that is being used for synthetic data creation should be better and produce higher-quality translations. After the second back-translation iteration, the pre-trained model is tuned again from the start now with the training data consisting of original parallel data and the data from only the second back-translation iteration. We do not evaluate the models produced after the second back-translation iteration on all of the translation directions, but only language pairs connected to Livonian in order to optimize the models to reach the best result for English-Livonian as was the objective in the WMT22 competition (Kocmi et al., 2022).

## 6.5 Additional Transfer Learning to Livonian

We took part in the WMT22 competition (Kocmi et al., 2022) and published our results (Tars et al., 2022a) which are partly described in this work also. To achieve the best results for the English-Livonian language pair, we proceed to fine-tune the best model after training with the second back-translation iteration on the original parallel dataset of Livonian-related language pairs (`en-liv`, `et-liv`, `lv-liv`). The model will somewhat forget the other Finno-Ugric language pairs as their translation quality drops, but since we were only interested in getting the best English to Livonian and vice versa results, we accept the drawback of further fine-tuning to a specific language.

## 6.6 Experimental Setup

Throughout the work, the BLEU[17] and chrF++[18] results were calculated using the SacreBLEU code (Post, 2018). The models all were trained using the Huggingface implementation of the M2M-100 framework. M2M-100 418 million and 1.2 billion parameter versions were used in the experiments. All models were trained using one Tesla A100 GPU with 40GB of VRAM, using the University of Tartu High-Performance Cluster (HPC) (University of Tartu, 2018). The models were initialized with the default hyperparameters set by the Huggingface implementation with the learning rate at 0.00005. The batch size was set to 12 segments with gradient accumulation step value of 8, which means that the backward pass (updating of model's parameters/weights) was performed after 8 update steps. This means that the batch size was 96 sentences. All of the experiments were trained for different numbers of epochs.

Model size comparison experiments were stopped at 25 epochs. For other experiments, the epochs were chosen according to validation data. The setup was to train 40 epochs with no early stopping or patience. Most of the epochs took very long to complete (multiple hours to multiple days depending on dataset size) so many of them did not reach 40 epochs due to time constraints.

More specifically, a premature end of the experiment was only needed for models trained with all of the original parallel data and models trained with original parallel data plus synthetic data from the first back-translation iteration. For other experiments, the epochs were chosen by monitoring the loss value and BLEU metric on the validation data. When the values started to worsen or plateau, we stopped the experiments and chose the best epochs. Here the models did not need more than 12 epochs to reach the highest quality. In cases where we compare the 1.2B model and the 418M model, the epoch we compare has seen the same number of updates.

[17] sacreBLEU signature: `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0`
[18] sacreBLEU signature: `nrefs:1|case:mixed|eff:yes|nc:6|nw:2|space:no|version:2.0.0`

# 7 Results

## 7.1 Model Size Comparison

### 7.1.1 Validation Data Results

**Averaged over all language pairs** We used two sizes of the M2M-100 model that we tuned on all of the original parallel data. In Figure 4, we compare the learning curve of the two models on validation data and average the result over all low-resource translation directions. Both of the models were trained on 25 epochs. From the graph, we can see that during the 25 epochs, the models learn gradually in parallel, with the smaller model being always a couple of BLEU points behind the bigger model. The difference is already apparent after the first epoch with the larger model beating the smaller one by about 2 BLEU points.

According to this, there is a clear benefit in choosing the larger model over the smaller one, although it comes with the downside of increased computational costs. However, the experiments might need to train longer, for more epochs, before showing any signs of the gap getting smaller. We trained the models for 25 epochs because of time and computational resource constraints.

**Detailed analysis** For deeper analysis, we take a look at a couple of language pairs separately. From Figure 5a, where we see the results of the Norwegian-South Sami language pair, at the ninth epoch, the smaller model actually surpasses the larger one and stays ahead of the larger model for the rest of the training period. Both of the curves, however, are trending down after the ninth epoch, which means that the best epoch for this particular language pair was the ninth one. However, since we are tuning a multilingual model with a shared encoder and decoder between all the languages, we cannot make decisions based only on one language pair.

The second example is about Estonian-Livonian which can be seen in Figure 5b. Here the larger model is ahead of the smaller one for the whole duration of the training, but the lines merge at the 25th epoch. For the larger model, the curve stays level for most of the training time, whereas the smaller model gets steadily better. The most likely reason for this is the amount of data. Since the amount of original parallel data is quite small for us, the larger model reaches the highest quality quite quickly, whereas the smaller one takes more time to adjust the parameters to the new data. The larger model starts to at one point overfit to the training data and lose in quality for some of the language pairs that we are training for when evaluated on the test data.

### 7.1.2 Evaluation Data Results

The results of the models compared on the held-out test set show similar tendencies, with the larger model performing consistently better than the smaller one. This can be seen in Table 4, where the gain over the smaller model is on average over all of

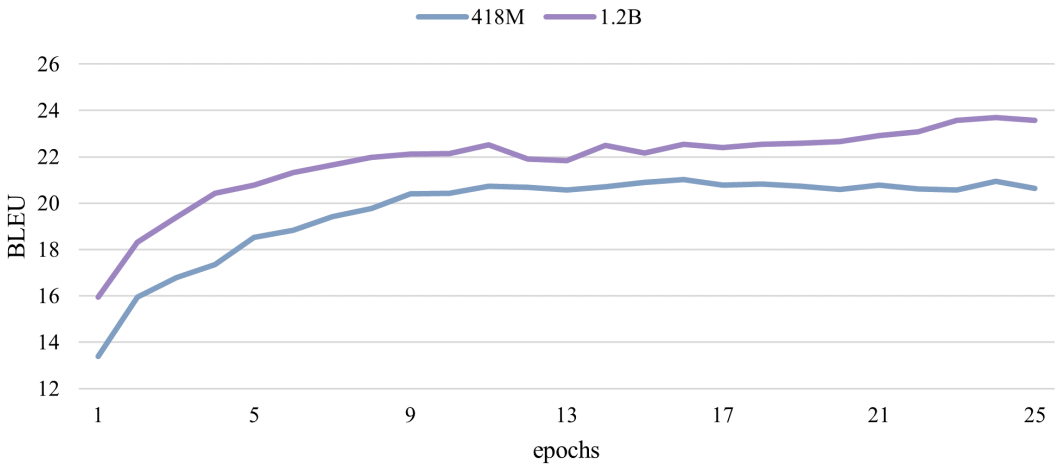
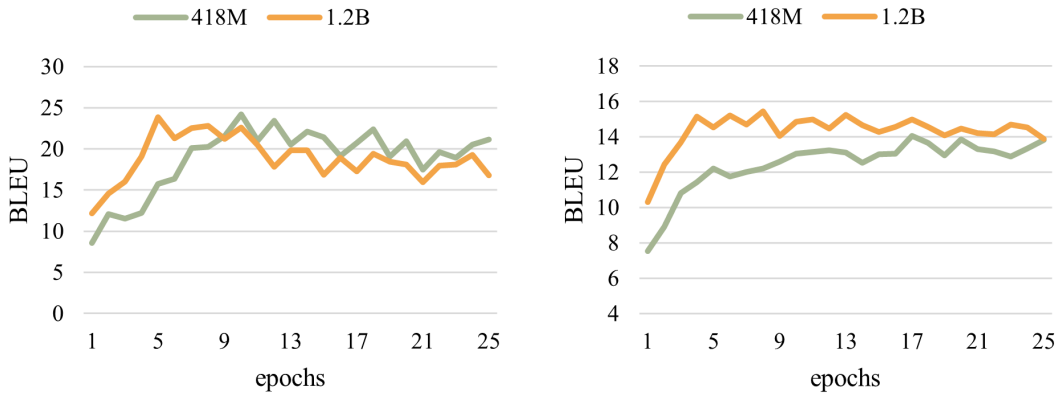



Figure 4. 418M vs 1.2B model on validation data for 25 epochs (averaged over all low-resource language pairs). The figure is from Tars et al. (2022b).




(a) 418M vs 1.2B model for `no-sma`.

(b) 418M vs 1.2B model for `et-liv`.

Figure 5. Comparison of 418M vs 1.2B models on specific language pairs. The figure is from Tars et al. (2022b).

the translation directions 3.3 BLEU points. The most significant improvements are related to the Sami languages with South Sami-Finnish translation direction scoring the highest quality improvement with the larger, 1.2 billion parameter, model. Another aspect related to this is the considerable gaps in improvement between different language pairs. The pattern here does not seem to be connected to language groupings inside the Finno-Ugric language family. The size of the dataset per language pair is also not an explanation, because the Norwegian-North Sami language pair had the most parallel data but improved quite modestly, by 0.4 BLEU compared to South Sami-Finnish translation direction (+10.6 BLEU points).

## 7.2 Effect of English

We trained the smaller 418 million parameter model with the dataset where we remove all language pairs with English before training. We then evaluate both of the models (**418M** and **not-en**) on the held-out test set created in Tars et al. (2022b) and "liv4ever" test set for Livonian language pairs. In Table 4, we notice that the quality difference of this model to the standard 418M model trained on all of the original parallel data is actually quite small, apart from Livonian-English, because now the model was shown zero examples of this language pair.

For other language pairs, the average difference was 0.2 BLEU points, which could be counted as a circumstantial difference and not a significant change in the automatic metric score. Other than for North Sami-Lule Sami and Inari Sami-North Sami, the change was around 1 BLEU point or smaller. The lack of impact could be explained by the amounts of English language pairs in the pre-training stage. The small amounts we added in the further training with only the original parallel data might not be significant enough to cause any change. Another explanation would be that the Finno-Ugric languages that we tune for do not actually get any useful information from English which also supports the theory that related high-resource languages should help low-resource languages more than non-related high-resource languages.

## 7.3 Related Language Groups

We performed these experiments also only with the smaller model 418M, because of resource constraints, and since we know that the larger model is better anyway, we can make quicker and more efficient comparisons between models trained with different sets of language pairs.

The aim of these experiments was to see whether certain language pairs need to be trained in even smaller language groups rather than with all of the Finno-Ugric language pairs together. We divided the low-resource languages into three smaller clusters: 1) Võro language pairs; 2) Livonian language pairs, and 3) Sami language pairs. The results of the experiments are reported in Table 4. Here we notice that since Estonian-Võro was

Table 4. BLEU scores on new held-out test data. **1.2B** was trained on the same data as **418M**. **not-en** had English removed from the training data. **only-group** signifies separate models trained with smaller datasets containing data for only that smaller language group. Δ shows the difference from **418M** model. **Green** indicates the best improvement per Δ column and **red** indicates the worst improvement. **bold line** separates the language pairs into groups.

| lang-pair | **418M** | **1.2B** | Δ | **not-en** | Δ | **only-group** | Δ |
|---|---|---|---|---|---|---|---|
| et-vro | 29.3 | 30.1 | 0.8 | 30.4 | 1.1 | 34.1 | 4.8 |
| vro-et | 36.3 | 37.2 | 0.9 | 35.4 | -0.9 | 40.0 | 3.8 |
| en-liv | 10.6 | 11.5 | 0.9 | 2.5 | -8.1 | 9.0 | **-1.7** |
| liv-en | 14.1 | 15.9 | 1.8 | 4.3 | **-9.7** | 14.3 | 0.2 |
| et-liv | 14.0 | 14.5 | 0.5 | 13.7 | -0.3 | 13.9 | -0.1 |
| liv-et | 19.0 | 19.6 | 0.6 | 18.3 | -0.7 | 18.7 | -0.3 |
| lv-liv | 13.6 | 15.1 | 1.5 | 13.4 | -0.2 | 13.9 | 0.3 |
| liv-lv | 18.4 | 20.5 | 2.1 | 18.3 | -0.1 | 20.3 | 2.0 |
| fi-sme | 40.2 | 42.9 | 2.7 | 40.2 | 0.0 | 41.3 | 1.1 |
| sme-fi | 48.5 | 50.1 | 1.6 | 47.9 | -0.6 | 47.7 | -0.8 |
| fi-sma | 17.4 | 26.6 | 9.2 | 18.2 | 0.7 | 21.7 | 4.3 |
| sma-fi | 20.8 | 31.5 | **10.6** | 22.2 | 1.4 | 27.9 | **7.1** |
| fi-smn | 52.1 | 53.3 | 1.2 | 52.1 | 0.0 | 53.2 | 1.0 |
| smn-fi | 70.6 | 75.4 | 4.9 | 71.0 | 0.4 | 74.3 | 3.7 |
| fi-sms | 32.9 | 33.1 | **0.3** | 32.6 | -0.3 | 33.7 | 0.8 |
| sms-fi | 57.0 | 61.4 | 4.4 | 58.9 | 1.9 | 61.5 | 4.5 |
| no-sma | 43.9 | 46.8 | 2.9 | 43.3 | -0.7 | 45.9 | 2.0 |
| sma-no | 50.4 | 53.5 | 3.1 | 49.9 | -0.5 | 51.3 | 0.9 |
| no-sme | 34.6 | 34.9 | 0.4 | 35.4 | 0.8 | 35.2 | 0.6 |
| sme-no | 44.8 | 45.8 | 1.1 | 44.8 | 0.0 | 45.0 | 0.2 |
| no-smj | 33.4 | 40.0 | 6.6 | 34.4 | 1.0 | 37.0 | 3.6 |
| smj-no | 48.2 | 52.4 | 4.3 | 48.2 | 0.0 | 50.0 | 1.8 |
| sme-sma | 33.1 | 36.3 | 3.2 | 33.7 | 0.6 | 35.6 | 2.5 |
| sma-sme | 37.8 | 44.4 | 6.7 | 37.5 | -0.2 | 40.3 | 2.5 |
| sme-smj | 28.7 | 34.1 | 5.4 | 31.1 | **2.4** | 31.1 | 2.4 |
| smj-sme | 38.9 | 46.4 | 7.5 | 39.9 | 1.0 | 42.5 | 3.6 |
| sme-smn | 27.4 | 33.5 | 6.2 | 28.5 | 1.1 | 30.0 | 2.6 |
| smn-sme | 32.6 | 34.2 | 1.6 | 30.5 | -2.1 | 31.7 | -0.9 |
| average | | | **3.3** | | **-0.4** | | **1.9** |

the only language pair in its group, the impact of being the sole focus gets an approximate 4 BLEU point gain. For Livonian, however, the effect seems to be the opposite. Here the

three language pairs involving Livonian were also in their own separate group. For the Sami language groups, where all of them were placed in the same language group, the results vary, but overall the impact of separating them from Võro and Livonian language pairs during training seems to produce a positive result with only North Sami-Finnish getting worse.

Overall, it seems that inside the Finno-Ugric language family, the low-resource languages do not help each other as much as we hoped. Perhaps one of the reasons is the scarcity of the data combined with a likely domain mismatch between language pairs' training data which has a damaging impact. As we have seen from some previous works, related high-resource languages help the low-resource languages, which might be because of the relatedness but also because high-resource language data is often cleaner and from different domains. The high-resource language data is representing the language, on the whole, better than some of the low-resource language pairs have the ability to do as a consequence of the low amounts of data.

## 7.4 Back-translation

We perform one back-translation iteration for all of the language pairs and present results for experiments on both sizes (418M, 1.2B) of the pre-trained M2M-100 model. This time we do not examine the results of removing some language pairs but rather train the models with all of the data. Otherwise training all the variants again would become too resource consuming and we already analyzed the effect of different language group settings in the previous parts.

The results of training the models with original parallel data and synthetic parallel data produced in the back-translation part can be seen in Table 5. We trained the **1.2B + bt1** and **418M + bt1** both for 12 epochs.

Low-resource languages usually benefit from additional synthetic parallel data, according to multiple experiments in related research. Here, however, some language pairs, especially some **low-high** pairs have gotten worse as a result. For example, in the low- to high-resource directions, Sami languages seem to struggle with the new information from the synthetic sentence pairs with South Sami(sma)-Finnish worsening by 12 BLEU points in comparison to the baseline 1.2B model. In the **high-low** direction, however, the overall result is positive with some considerable improvements happening in Finnish-South Sami, Norwegian-South Sami, and Finnish-Inari Sami(smn) directions. For the **low-low** directions, the results are more irregular with scores jumping between Lule Sami(smj)-North Sami(sme)'s -12.6 and North Sami-South Sami's 18.9 point changes in the score. The same experiments on the smaller 418M model seem to follow the patterns seen with the larger 1.2B model.

Overall, the 1.2B model trained with back-translation data is still on average 2.1 BLEU points better than the one trained on only original parallel data. While we see that for the language pairs involving Sami languages, the results vary strongly, even inside

Table 5. BLEU scores on new held-out test data comparing models trained only on original parallel data to models trained with original parallel data + data from the first back-translation iteration (**bt1**). Δ signifies the difference between **bt1** model and the respective baseline model. **Green** indicates the best improvement per Δ column and **red** indicates the worst improvement

| **high-low** | **1.2B** | **1.2B + bt1** | Δ | **418M** | **418M + bt1** | Δ |
|---|---|---|---|---|---|---|
| en-liv | 11.5 | 14.3 | 2.8 | 10.6 | 12.2 | 1.5 |
| et-liv | 14.5 | 22.9 | 8.4 | 14.0 | 19.7 | 5.8 |
| et-vro | 30.1 | 30.4 | 0.4 | 29.3 | 29.9 | 0.6 |
| fi-sma | 26.6 | 43.4 | **16.8** | 17.4 | 20.9 | 3.5 |
| fi-sme | 42.9 | 39.3 | **-3.6** | 40.2 | 36.6 | **-3.6** |
| fi-smn | 53.3 | 64.9 | 11.6 | 52.1 | 58.7 | **6.6** |
| fi-sms | 33.1 | 35.2 | 2.1 | 32.9 | 31.7 | -1.2 |
| lv-liv | 15.1 | 25.0 | 9.8 | 13.6 | 19.8 | 6.1 |
| no-sma | 46.8 | 58.7 | 11.9 | 43.9 | 47.7 | 3.8 |
| no-sme | 34.9 | 33.6 | -1.4 | 34.6 | 33.1 | -1.4 |
| no-smj | 40.0 | 45.2 | 5.1 | 33.4 | 32.5 | -0.9 |
| average | | | **5.8** | | | **1.9** |
| **low-high** | | | | | | |
| liv-en | 15.9 | 17.7 | 1.9 | 14.1 | 16.4 | 2.3 |
| liv-et | 19.6 | 24.8 | 5.2 | 19.0 | 23.3 | 4.3 |
| liv-lv | 20.5 | 27.7 | **7.2** | 18.4 | 26.9 | **8.5** |
| sma-fi | 31.5 | 19.1 | **-12.3** | 20.8 | 13.0 | **-7.9** |
| sma-no | 53.5 | 49.3 | -4.2 | 50.4 | 45.8 | -4.6 |
| sme-fi | 50.1 | 49.1 | -1.1 | 48.5 | 47.1 | -1.4 |
| sme-no | 45.8 | 44.8 | -1.1 | 44.8 | 42.5 | -2.3 |
| smj-no | 52.4 | 47.9 | -4.5 | 48.2 | 44.3 | -3.8 |
| smn-fi | 75.4 | 69.9 | -5.6 | 70.6 | 63.1 | -7.5 |
| sms-fi | 61.4 | 61.6 | 0.2 | 57.0 | 55.2 | -1.8 |
| vro-et | 37.2 | 41.8 | 4.7 | 36.3 | 42.4 | 6.1 |
| average | | | **-0.9** | | | **-0.7** |
| **low-low** | | | | | | |
| sma-sme | 44.4 | 35.8 | -8.7 | 37.8 | 29.1 | -8.7 |
| sme-sma | 36.3 | 55.2 | **18.9** | 33.1 | 39.0 | 5.9 |
| sme-smj | 34.1 | 37.5 | 3.5 | 28.7 | 28.6 | -0.2 |
| sme-smn | 33.5 | 39.0 | 5.5 | 27.4 | 35.6 | **8.2** |
| smj-sme | 46.4 | 33.8 | **-12.6** | 38.9 | 27.8 | **-11.2** |
| smn-sme | 34.2 | 31.9 | -2.3 | 32.6 | 32.4 | -0.2 |
| average | | | **0.7** | | | **-1.0** |

one language pair (e.g. fi-sma), for Võro and Livonian back-translation seems to give a positive effect. The average result is carried by some big increases in the evaluation

of this test set, with sme-sma, fi-sma, no-sma and fi-smn jumping for more than 10 BLEU points. Meaning that sma and smn monolingual data was probably very similar to the test data.

The reasons for this variability might be the low quality of the base model, low-quality original parallel data, the overwhelming amount of out-of-domain monolingual data, or the lack of evaluation data that represents the language pair well enough. In order to find out the exact reason, more large-scale and in-depth analysis is needed which is out of the scope of this work.

## 7.5 chrF++ Results

We also report chrF++ results for models described in Sections 7.1 to 7.4, showing complementary results to Tables 4 and 5. The analysis of chrF++ scores, which can be seen in the Appendix Table 11 and Table 12, shows an overall correlation with the results calculated with the BLEU metric.

In Table 11 we see that the averages follow a similar pattern to BLEU scores, however, the gaps between the best improvement and the worst improvement per model sets are generally smaller than those seen in Table 4. The only exception is the **not-en** experiment, where Livonian-English shows to be more sensitive to losing the English data according to chrF++.

For the back-translation experiment analysis, we notice similar tendencies. The best and the worst improvements in Table 12 overlap with those in Table 5, except for the high-resource to low-resource translation directions on the 418 million parameter model.

## 7.6 Additional Experiments for Livonian

The second back-translation iteration was performed for all of the language pairs. The first batch of the models described in Table 6 were trained with all original parallel data and the synthetic parallel data. For the second batch seen in Table 7 the **bt1** and **bt2** included only the Livonian language pairs.

From Table 6 we notice that **bt2** improves over the baseline models (**1.2B** and **418M**) but overall underperforms compared to **bt1**. This is an odd behavior because **bt2** data was produced by the **1.2B + bt1** model and thus **bt2** should improve over the quality of **bt1**. When we look at the results in Table 7, where back-translation data involved only Livonian language pairs, we see that **bt2** actually does improve over **bt1**, contrasting the results seen in Table 6. Comparing the results of these two tables we could infer that perhaps for the first batch of experiments, the other language pairs disturbed the Livonian pairs either during the back-translation production or during training. Another reason for this could be that **bt2** was trained with the model that was deemed best over all of the language pairs. Thus, even though Livonian scores got better, for one they are still quite low-quality (under 30 BLEU), and as we noticed in Table 5, some of the language pairs

Table 6. BLEU metric results for Livonian-specific experiments. **bt1** and **bt2** included all of the Finno-Ugric language pairs. **tune** refers to additional fine-tuning with the original parallel data for the Livonian language pairs. $\Delta$ is the difference between the **bt2 + tune** model and the baseline model (**1.2B** or **418M**).

| lang-pair | **1.2B** | **1.2B + bt1** | **1.2B + bt2** | **1.2B + bt2 + tune** | $\Delta$ |
|---|---|---|---|---|---|
| en-liv | 11.5 | 14.3 | 11.6 | 12.9 | 1.4 |
| liv-en | 15.9 | 17.7 | 17.9 | 20.4 | 4.5 |
| et-liv | 14.5 | 22.9 | 17.9 | 20.5 | 5.9 |
| liv-et | 19.6 | 24.8 | 23.7 | 26.5 | 6.9 |
| lv-liv | 15.1 | 25.0 | 18.7 | 21.6 | 6.5 |
| liv-lv | 20.5 | 27.7 | 26.1 | 29.1 | 8.6 |
| average | | | | | **5.6** |
| | **418M** | **418M + bt1** | **418M + bt2** | **418M + bt2 + tune** | |
| en-liv | 10.6 | 12.2 | 9.1 | 9.7 | -0.9 |
| liv-en | 14.1 | 16.4 | 15.4 | 18.8 | 4.7 |
| et-liv | 14.0 | 19.7 | 15.6 | 15.9 | 1.9 |
| liv-et | 19.0 | 23.3 | 22.5 | 24.5 | 5.4 |
| lv-liv | 13.6 | 19.8 | 15.6 | 16.5 | 2.8 |
| liv-lv | 18.4 | 26.9 | 23.9 | 25.8 | 7.5 |
| average | | | | | **3.6** |

had a significant decrease in the BLEU score, which in turn could disturb the Livonian language pairs since the languages are all sharing the model parameters.

For both of the situations (training with all back-translation data vs with only Livonian data), the end result improves almost always if the final fine-tuning is performed only on Livonian original parallel data.

### 7.6.1 FLORES-200

For Livonian-specific experiments, we had a chance to use the 250 sentences translated from the FLORES-200 benchmark into Livonian. Since the FLORES-200 was already available in English, Estonian and Latvian, we can compare the results of the "liv4ever" test set to the results computed on part of the FLORES-200 test set. This way we gain a better overview of the quality of our models and have the chance to see whether the test sets agree with each other.

In Table 8 we have displayed BLEU scores for experiments with the larger, 1.2 billion parameter model. In the first part of the table, the experiment with back-translation data included all of the language pairs (Table 8a). In the second part of the table, the back-translation experiments had data only for Livonian-specific language pairs (Table 8b).

Right away we can see a pattern where translation directions into Livonian (*-liv) are

Table 7. BLEU metric results for Livonian-specific experiments. **bt1** and **bt2** include only Livonian back-translation data. **tune** refers to additional fine-tuning with the Livonian original parallel data. $\Delta$ is derived by getting the difference between the **bt2 + tune** model and the baseline model (**1.2B** or **418M**).

| lang-pair | **1.2B** | **1.2B + bt1** | **1.2B + bt2** | **1.2B + bt2 + tune** | $\Delta$ |
|---|---|---|---|---|---|
| en-liv | 11.5 | 14.8 | 15.4 | 18.03 | 6.5 |
| liv-en | 15.9 | 17.81 | 18.24 | 20.76 | 4.9 |
| et-liv | 14.5 | 23.06 | 24.19 | 25.5 | 11.0 |
| liv-et | 19.6 | 24.04 | 25.11 | 26.41 | 6.8 |
| lv-liv | 15.1 | 25.05 | 26.81 | 27.13 | 12.0 |
| liv-lv | 20.5 | 27.48 | 28.68 | 30.37 | 9.9 |
| average | | | | | **8.5** |
| | **418M** | **418M + bt1** | **418M + bt2** | **418M + bt2 + tune** | |
| en-liv | 10.6 | 12.39 | 12.06 | 14.08 | 3.4 |
| liv-en | 14.1 | 17.72 | 17.37 | 18.75 | 4.7 |
| et-liv | 14.0 | 19.51 | 20.52 | 22.16 | 8.2 |
| liv-et | 19.0 | 23.23 | 23.42 | 25.9 | 6.9 |
| lv-liv | 13.6 | 20.49 | 21.93 | 24.42 | 10.8 |
| liv-lv | 18.4 | 26.64 | 27.2 | 28.94 | 10.6 |
| average | | | | | **7.4** |

performing much worse than the translation directions from Livonian (liv-*) directions. This tendency was not very clear from the "liv4ever" test set results (Tables 6 and 7), except for maybe in Table 6 with the experiments on the **418M** model, but the gaps between *-liv and liv-* directions here are much smaller.

## 7.7 Comparison to Previous State-Of-The-Art

In terms of achieving a new state-of-the-art quality for all of the low-resource Finno-Ugric language pairs in our work, we reach that goal as can be seen in Table 9 and 10. The back-translation results are not described in Table 9, because we beat the previous best without using it. The 1.2B baseline yields the most improvements in a number of language pairs. Võro gets a huge gain from the **only-group** experiment where it was the sole language pair in the dataset. For the Livonian pairs, we compare to results achieved after four back-translation iterations in Rikters et al. (2022). Our best model was the 1.2B model trained on original parallel data and **bt2** data that only contained Livonian language pairs plus final fine-tuning on original Livonian parallel data. All in all, our approach overtakes the previous best results by using fewer back-translation iterations.

Table 8. BLEU scores for Livonian-specific experiments evaluated on a part of the FLORES-200 test set.

(a) **bt** experiments performed with all of the language pairs

| lang-pair | **1.2B** | **1.2B + bt1** | **1.2B + bt2** | **1.2B + bt2 + tune** | Δ |
|---|---|---|---|---|---|
| en-liv | 7.3 | 7.4 | 8.3 | 10.4 | 3.1 |
| liv-en | 15.0 | 24.2 | 23.6 | 25.0 | 10.1 |
| et-liv | 10.7 | 11.2 | 12.7 | 14.4 | 3.6 |
| liv-et | 19.8 | 25.7 | 27.3 | 28.9 | 9.0 |
| lv-liv | 5.3 | 5.0 | 6.1 | 6.6 | 1.3 |
| liv-lv | 9.9 | 14.3 | 12.7 | 14.0 | 4.1 |
| average | | | | | **5.2** |

(b) **bt** experiments performed with only Livonian language pairs.

| lang-pair | **1.2B** | **1.2B + bt1** | **1.2B + bt2** | **1.2B + bt2 + tune** | Δ |
|---|---|---|---|---|---|
| en-liv | 7.3 | 6.2 | 7.9 | 8.1 | 0.9 |
| liv-en | 15.0 | 23.5 | 23.5 | 23.8 | 8.8 |
| et-liv | 10.7 | 9.1 | 11.1 | 12.1 | 1.3 |
| liv-et | 19.8 | 27.1 | 27.9 | 28.7 | 8.9 |
| lv-liv | 5.3 | 4.9 | 6.2 | 6.4 | 1.1 |
| liv-lv | 9.9 | 15.5 | 13.8 | 14.6 | 4.8 |
| average | | | | | **4.3** |

Table 9. BLEU scores with test data from Tars et al. (2021). **418M** and **1.2B** refer to models trained with all data. **not-en** refers to model trained without any English data. **only-group** refers to a model trained only on that specific language group data. *prev_best* refers to best results by Tars et al. (2021). **bold** - best BLEU score for a language pair. $\Delta$ indicates the difference between our best and the previous best per language pair. Table is adapted from Tars et al. (2022b).

| lang-pair | **418M** | **not-en** | **only-group** | **1.2B** | *prev_best* | $\Delta$ |
|---|---|---|---|---|---|---|
| et-vro | 25.7 | 25.7 | **30.3** | 26.0 | *26.2* | 4.1 |
| vro-et | 30.3 | 29.7 | **34.0** | 31.7 | *31.7* | 2.3 |
| fi-sme | **38.5** | 38.3 | 38.0 | 37.8 | *32.3* | 6.2 |
| sme-fi | 42.8 | 45.1 | 45.2 | **45.8** | *37.5* | 8.3 |
| fi-sma | 17.8 | 20.0 | 22.6 | **25.3** | *12.4* | 12.9 |
| sma-fi | 21.7 | 21.9 | 27.9 | **29.4** | *10.9* | 18.5 |
| sme-sma | 33.5 | 33.7 | 34.9 | **38.1** | *21.6* | 16.5 |
| sma-sme | 35.4 | 36.5 | 38.4 | **43.0** | *21.0* | 22.0 |
| average | | | | | | **11.4** |

Table 10. BLEU scores comparing our **1.2B + bt2 + tune** model trained with only Livonian specific **bt2** to results reported in Rikters et al. (2022). Differences are portrayed in the $\Delta$ column.

| lang-pair | Rikters et al. (2022) | our best | $\Delta$ |
|---|---|---|---|
| en-liv | *11.0* | 18.0 | 7.0 |
| liv-en | *19.0* | 20.8 | 1.8 |
| et-liv | *16.5* | 25.5 | 9.0 |
| liv-et | *23.1* | 26.4 | 3.4 |
| lv-liv | *17.7* | 27.1 | 9.5 |
| liv-lv | *25.2* | 30.4 | 5.1 |
| average | | | **6.0** |

# 8 Discussion

## 8.1 Back-translation

Throughout the analysis of our results, we noticed out-of-ordinary patterns where back-translation did not perform as expected but rather varied very strongly between different language pairs and dataset scenarios.

There could be multiple reasons for the erratic behavior of the back-translation experiments. One explanation could be that we produced synthetic sentences from high amounts of monolingual data that was outside of the domain of the test data. Since the test data was originally held-out from original parallel training data, it was from the same domain as all of the data that the 1.2B baseline model was trained on. Now, taking, for example, the Finnish monolingual news dataset to produce back-translation data for `sma-fi`, it throws off the balance of in-domain and out-of-domain data. There were only 2700 sentences of parallel data for that language pair. After back-translation, there were 502 700 sentences. This means that there were 500 000 possibly out-of-domain sentences added to the training set, affecting the test results severely. For certain language pairs, we can see this problem as either overfitting to the back-translated data or that the test data does not actually represent the domain that we are training for.

This also highlights the importance of creating a good benchmark and testing on multiple test sets from different domains for a more accurate estimation of model quality. In conjunction with that, the other side of the problem is acquiring well-rounded training sets for a language pair, or well-rounded monolingual data, which for low-resource languages is as problematic as finding multiple test sets for these languages. In addition, this demonstrates the analysis done by back-translation research in the past about the diversity of back-translation data and how the base model's quality is one of the key aspects of allowing good translation models to be trained.

One of the obvious culprits behind these variations in back-translation results could be the low quality of the parallel data itself. The original parallel data underwent several filtering heuristics, but due to not understanding the languages ourselves, we are unable to clearly state if the quality of the parallel data is good enough for training. For the same reason, we are not able to give a grade of quality to the test data that was ultimately held out from the filtered training data and therefore shares its features.

## 8.2 Future Work

Part of the future work that involves adding more low-resource Finno-Ugric languages has been implemented and published but was out of the scope of this work (Yankovskaya et al., 2023).

For all of the experiments analyzed, we could try training them for more epochs, to get a better overview of quality and be more sure that the data was exhausted during the

training to the best of its possibilities. Another way to gain a more trustworthy view of the quality is by performing human evaluation. However, finding multiple people who speak any of the low-resource Finno-Ugric languages could prove to be problematic and it is much more resource-heavy than using automatic metrics.

In terms of back-translation problems, we could perform a deeper analysis of the reasons why it did not work as expected, to determine whether the problem was in the training data, test data, or training methods. Otherwise, we could try filtering the back-translation data, because we left that step out in this work. Another way would be to retrain the models but add back-translation data to only those language pairs that benefited from it according to our evaluation results.

# 9 Conclusion

The aim of this work was to achieve higher machine translation quality for Finno-Ugric low-resource languages as well as create the first neural machine translation system for some of the low-resource Finno-Ugric languages involved in this work.

We utilized pre-trained multilingual machine translation models and performed cross-lingual transfer learning to the selected low-resource language pairs. We compared two different-sized pre-trained models throughout the analysis and concluded that even with our small-sized dataset, the larger (more than twice the size of the smaller one) model performs better in most experiments than the smaller one. The experiments with different dataset settings showed that training the multiple low-resource language pairs all together in one model could be disadvantageous to some smaller groups in the family and additional experiments showed that indeed they benefited from being tuned separately. This, however, unfortunately means that there would have to be multiple models trained and deployed at the same time, which is more complicated and resource-consuming than one multilingual neural machine translation model.

We also employed the technique of back-translation to create synthetic data in order to mitigate the data scarcity issue with low-resource languages. In our detailed analysis of the results, we found that although back-translation has been a trustworthy method in low-resource machine translation, here it produced very variable results, with some language pairs gaining ~18 BLEU points and others losing ~12 BLEU points as a result of adding the synthetic data. We speculated on the multiple reasons why this might be but leave the in-depth analysis for future work.

Overall, we achieved state-of-the-art results for all of our low-resource Finno-Ugric language pairs.

# References


R. Aharoni, M. Johnson, and O. Firat. Massively multilingual neural machine translation. 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 3874–3884, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1388. URL https://aclanthology.org/N19-1388.

M. Aulamo, S. Virpioja, and J. Tiedemann. OpusFilter: A configurable parallel corpus filtering toolbox. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 150–156, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-demos.20.

L. Burchell, A. Birch, and K. Heafield. Exploring diversity in back translation for low-resource machine translation. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 67–79, Hybrid, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.deeplo-1.8. URL https://aclanthology.org/2022.deeplo-1.8.

S. Edunov, M. Ott, M. Auli, and D. Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1045. URL https://aclanthology.org/D18-1045.

A. Fan, S. Bhosale, H. Schwenk, Z. Ma, A. El-Kishky, S. Goyal, M. Baines, O. Celebi, G. Wenzek, V. Chaudhary, N. Goyal, T. Birch, V. Liptchinsky, S. Edunov, M. Auli, and A. Joulin. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48, 2021. URL http://jmlr.org/papers/v22/20-1307.html.

P. Gage. A new algorithm for data compression. *C Users J.*, 12(2):23–38, feb 1994. ISSN 0898-9788.

N. Goyal, C. Gao, V. Chaudhary, P.-J. Chen, G. Wenzek, D. Ju, S. Krishnan, M. Ranzato, F. Guzmán, and A. Fan. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538, 2022. doi: 10.1162/tacl_a_00474. URL https://aclanthology.org/2022.tacl-1.30.

V. Goyal, S. Kumar, and D. M. Sharma. Efficient neural machine translation for low-resource languages via exploiting related languages. In *Proceedings of the 58th Annual*

*Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 162–168, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-srw.22. URL https://aclanthology.org/2020.acl-srw.22.

J. Gu, H. Hassan, J. Devlin, and V. O. Li. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1032. URL https://aclanthology.org/N18-1032.

B. Haddow, R. Bawden, A. V. M. Barone, J. Helcl, and A. Birch. Survey of Low-Resource Machine Translation. *Computational Linguistics*, 48(3):673–732, 09 2022. ISSN 0891-2017. doi: 10.1162/coli_a_00446. URL https://doi.org/10.1162/coli_a_00446.

V. C. D. Hoang, P. Koehn, G. Haffari, and T. Cohn. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-2703. URL https://aclanthology.org/W18-2703.

Y. Kim, Y. Gao, and H. Ney. Effective cross-lingual transfer of neural machine translation models without shared vocabularies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1246–1257, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1120. URL https://aclanthology.org/P19-1120.

T. Kocmi and O. Bojar. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium, Oct. 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6325. URL https://aclanthology.org/W18-6325.

T. Kocmi, R. Bawden, O. Bojar, A. Dvorkovich, C. Federmann, M. Fishel, T. Gowda, Y. Graham, R. Grundkiewicz, B. Haddow, R. Knowles, P. Koehn, C. Monz, M. Morishita, M. Nagata, T. Nakazawa, M. Novák, M. Popel, and M. Popović. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid), Dec. 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.wmt-1.1.

T. Kudo and J. Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*,

pages 66–71, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012.

T. Q. Nguyen and D. Chiang. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan, Nov. 2017. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I17-2050.

NLLB Team, M. R. Costa-jussà, J. Cross, O. Çelebi, M. Elbayad, K. Heafield, K. Heffernan, E. Kalbassi, J. Lam, D. Licht, J. Maillard, A. Sun, S. Wang, G. Wenzek, A. Youngblood, B. Akula, L. Barrault, G. Mejia-Gonzalez, P. Hansanti, J. Hoffman, S. Jarrett, K. R. Sadagopan, D. Rowe, S. Spruit, C. Tran, P. Andrews, N. F. Ayan, S. Bhosale, S. Edunov, A. Fan, C. Gao, V. Goswami, F. Guzmán, P. Koehn, A. Mourachko, C. Ropers, S. Saleem, H. Schwenk, and J. Wang. No language left behind: Scaling human-centered machine translation. 2022.

E. S. Olivas, J. D. M. Guerrero, M. M. Sober, J. R. M. Benedito, and A. J. S. Lopez. *Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques - 2 Volumes*. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA, 2009. ISBN 1605667668.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

M. Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3049. URL https://aclanthology.org/W15-3049.

M. Popović. chrF deconstructed: beta parameters and n-gram weights. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 499–504, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2341. URL https://aclanthology.org/W16-2341.

M. Popović. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4770. URL https://aclanthology.org/W17-4770.

M. Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, Oct. 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W18-6319.

M. Rikters, M. Pinnis, and R. Krišlauks. Training and adapting multilingual NMT for less-resourced and morphologically rich languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://aclanthology.org/L18-1595.

M. Rikters, M. Tomingas, T. Tuisk, V. Ernštreits, and M. Fishel. Machine translation for Livonian: Catering to 20 speakers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 508–514, Dublin, Ireland, May 2022. Association for Computational Linguistics.

R. Sennrich and B. Zhang. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1021. URL https://aclanthology.org/P19-1021.

R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, Aug. 2016a. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009. URL https://aclanthology.org/P16-1009.

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, Aug. 2016b. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

M. Tars, A. Tättar, and M. Fišel. Extremely low-resource machine translation for closely related languages. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 41–52, Reykjavik, Iceland (Online), 2021. Linköping University Electronic Press, Sweden.

M. Tars, T. Purason, and A. Tättar. Teaching unseen low-resource languages to large translation models. In *Proceedings of the Seventh Conference on Machine Translation*, pages 375–380, Abu Dhabi, December 2022a. Association for Computational Linguistics. URL https://aclanthology.org/2022.wmt-1.33.

M. Tars, A. Tättar, and M. Fišel. Cross-lingual transfer from large multilingual translation models to unseen under-resourced languages. *Baltic Journal of Modern Computing*, 10.3:435–446, 2022b. doi: https://doi.org/10.22364/bjmc.2022.10.3.16.

J. Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

University of Tartu. UT Rocket, 2018. URL https://share.neic.no/#/marketplace-public-offering/c8107e145e0d41f7a016b72825072287/.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

R. Wang, X. Tan, R. Luo, T. Qin, and T.-Y. Liu. A survey on low-resource neural machine translation. In Z.-H. Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4636–4643. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/629. URL https://doi.org/10.24963/ijcai.2021/629. Survey Track.

L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.41. URL https://aclanthology.org/2021.naacl-main.41.

L. Yankovskaya, M. Tars, A. Tättar, and M. Fishel. Machine translation for low-resource finno-ugric languages. In *The 24rd Nordic Conference on Computational Linguistics*, 2023. URL https://openreview.net/forum?id=DX-XHq9_Pa.

B. Zhang, P. Williams, I. Titov, and R. Sennrich. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.148. URL https://aclanthology.org/2020.acl-main.148.

B. Zoph, D. Yuret, J. May, and K. Knight. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas, Nov. 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1163. URL https://aclanthology.org/D16-1163.

# Appendix

# I. chrF++ Results

Table 11. chrF++ scores on new held-out test data. **1.2B** was trained on the same data as **418M**. **not-en** had English removed from the training data. **only-group** signifies separate models trained with smaller datasets containing data for only that smaller language group. Δ shows the difference from **418M** model. **Green** indicates the best improvement per Δ column and **red** indicates the worst improvement. **bold line** separates the language pairs into groups.

| lang-pair | **418M** | **1.2B** | Δ | **not-en** | Δ | **only-group** | Δ |
|---|---|---|---|---|---|---|---|
| et-vro | 54.8 | 54.8 | -0.1 | 55.3 | 0.5 | 57.7 | 2.9 |
| vro-et | 58.0 | 58.7 | 0.7 | 57.4 | -0.6 | 60.7 | 2.7 |
| en-liv | 29.9 | 31.1 | 1.2 | 17.4 | -12.5 | 30.6 | 0.7 |
| liv-en | 37.2 | 38.5 | 1.3 | 22.2 | **-15.0** | 39.0 | 1.8 |
| et-liv | 38.3 | 38.4 | 0.1 | 37.6 | -0.7 | 38.8 | 0.4 |
| liv-et | 46.5 | 46.9 | 0.4 | 46.2 | -0.2 | 47.1 | 0.6 |
| lv-liv | 36.8 | 37.9 | 1.0 | 36.7 | -0.1 | 38.1 | 1.3 |
| liv-lv | 44.3 | 44.9 | 0.7 | 43.7 | -0.6 | 46.4 | 2.1 |
| fi-sme | 65.5 | 66.8 | 1.2 | 65.2 | -0.3 | 66.1 | 0.6 |
| sme-fi | 68.6 | 69.4 | 0.8 | 68.1 | -0.5 | 68.2 | -0.4 |
| fi-sma | 47.1 | 52.4 | 5.3 | 47.2 | 0.1 | 49.6 | 2.5 |
| sma-fi | 44.6 | 51.4 | **6.8** | 45.4 | 0.8 | 48.9 | **4.3** |
| fi-smn | 75.8 | 76.7 | 0.9 | 75.8 | 0.0 | 76.5 | 0.8 |
| smn-fi | 85.0 | 87.3 | 2.3 | 85.1 | 0.1 | 87.0 | 2.0 |
| fi-sms | 61.6 | 61.3 | **-0.3** | 61.3 | -0.3 | 61.2 | -0.4 |
| sms-fi | 75.8 | 77.7 | 2.0 | 76.8 | **1.1** | 78.3 | 2.6 |
| no-sma | 69.4 | 70.8 | 1.4 | 69.3 | 0.0 | 70.5 | 1.1 |
| sma-no | 69.3 | 71.3 | 2.0 | 69.1 | -0.2 | 69.8 | 0.5 |
| no-sme | 62.5 | 62.6 | 0.1 | 63.1 | 0.6 | 62.8 | 0.3 |
| sme-no | 67.0 | 67.5 | 0.5 | 66.8 | -0.2 | 66.8 | -0.3 |
| no-smj | 62.6 | 66.1 | 3.5 | 62.9 | 0.4 | 64.7 | 2.2 |
| smj-no | 68.5 | 71.2 | 2.7 | 68.4 | 0.0 | 69.8 | 1.3 |
| sme-sma | 60.8 | 62.4 | 1.6 | 60.7 | -0.1 | 62.1 | 1.3 |
| sma-sme | 61.7 | 65.1 | 3.4 | 61.2 | -0.5 | 62.7 | 1.0 |
| sme-smj | 58.1 | 60.8 | 2.6 | 58.7 | 0.6 | 59.2 | 1.0 |
| smj-sme | 63.2 | 67.7 | 4.5 | 63.7 | 0.5 | 65.3 | 2.1 |
| sme-smn | 56.7 | 61.2 | 4.4 | 57.4 | 0.7 | 58.1 | 1.3 |
| smn-sme | 61.5 | 62.6 | 1.1 | 60.1 | -1.4 | 60.7 | **-0.8** |
| average | | | **1.9** | | **-1.0** | | **1.3** |

Table 12. chrF++ scores on new held-out test data comparing models trained only on original parallel data to models trained with original parallel data + data from the first back-translation iteration (**bt1**). Δ signifies the difference between **bt1** model and the respective baseline model. **Green** indicates the best improvement per Δ column and **red** indicates the worst improvement

| **high-low** | **1.2B** | **1.2B + bt1** | Δ | **418M** | **418M + bt1** | Δ |
|---|---|---|---|---|---|---|
| en-liv | 31.1 | 33.5 | 2.4 | 29.9 | 31.5 | 1.6 |
| et-liv | 38.4 | 43.2 | 4.8 | 38.3 | 40.9 | 2.6 |
| et-vro | 54.8 | 56.1 | 1.4 | 54.8 | 55.9 | 1.1 |
| fi-sma | 52.4 | 63.4 | **11.0** | 47.1 | 49.5 | 2.4 |
| fi-sme | 66.8 | 64.5 | **-2.2** | 65.5 | 62.6 | -2.9 |
| fi-smn | 76.7 | 82.1 | 5.4 | 75.8 | 78.6 | 2.8 |
| fi-sms | 61.3 | 60.9 | -0.4 | 61.6 | 56.8 | **-4.8** |
| lv-liv | 37.9 | 43.5 | 5.6 | 36.8 | 40.0 | **3.2** |
| no-sma | 70.8 | 76.8 | 6.0 | 69.4 | 71.6 | 2.2 |
| no-sme | 62.6 | 62.5 | -0.2 | 62.5 | 61.9 | -0.6 |
| no-smj | 66.1 | 67.0 | 0.9 | 62.6 | 59.7 | -2.9 |
| average | | | **3.2** | | | **0.4** |
| **low-high** | | | | | | |
| liv-en | 38.5 | 41.5 | 3.0 | 37.2 | 39.9 | 2.8 |
| liv-et | 46.9 | 50.6 | 3.7 | 46.5 | 49.3 | 2.8 |
| liv-lv | 44.9 | 50.0 | **5.1** | 44.3 | 49.2 | **5.0** |
| sma-fi | 51.4 | 43.6 | **-7.9** | 44.6 | 39.3 | **-5.3** |
| sma-no | 71.3 | 69.0 | -2.3 | 69.3 | 67.3 | -2.0 |
| sme-fi | 69.4 | 68.9 | -0.5 | 68.6 | 68.3 | -0.3 |
| sme-no | 67.5 | 67.7 | 0.2 | 67.0 | 66.2 | -0.9 |
| smj-no | 71.2 | 68.9 | -2.3 | 68.5 | 66.5 | -2.0 |
| smn-fi | 87.3 | 85.2 | -2.1 | 85.0 | 82.3 | -2.7 |
| sms-fi | 77.7 | 78.8 | 1.1 | 75.8 | 75.0 | -0.7 |
| vro-et | 58.7 | 62.0 | 3.3 | 58.0 | 62.4 | 4.4 |
| average | | | **0.1** | | | **0.1** |
| **low-low** | | | | | | |
| sma-sme | 65.1 | 60.3 | -4.8 | 61.7 | 56.6 | -5.1 |
| sme-sma | 62.4 | 73.8 | **11.4** | 60.8 | 64.8 | 4.0 |
| sme-smj | 60.8 | 61.9 | 1.2 | 58.1 | 56.5 | -1.6 |
| sme-smn | 61.2 | 64.8 | 3.6 | 56.7 | 63.3 | **6.6** |
| smj-sme | 67.7 | 60.0 | **-7.7** | 63.2 | 56.7 | **-6.6** |
| smn-sme | 62.6 | 60.9 | -1.8 | 61.5 | 61.9 | 0.4 |
| average | | | **0.3** | | | **-0.4** |

# II. Licence

## Non-exclusive licence to reproduce thesis and make thesis public

I, **Maali Tars**,
*(*author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   **Low-resource Finno-Ugric Neural Machine Translation through Cross-lingual Transfer Learning**,
   *(*title of thesis)

   supervised by Andre Tättar.
   *(*supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Maali Tars
***09/05/2023***