UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Taido Purason

Modular Septilingual Neural Machine Translation

Bachelor's Thesis (9 ECTS)

Supervisors: Andre Tättar, MSc
Elizaveta Korotkova, MSc

Tartu 2021

Modular Septilingual Neural Machine Translation

Abstract: Currently, the majority of state-of-the-art multilingual neural machine translation systems use a single universal model which fully shares parameters between all language pairs. The University of Tartu Neural Machine Translation system uses the universal architecture as well, and thus also suffers from the problems associated with it, such as limited capacity per language pair. Previous research has shown that a modularized approach with language-specific encoders and decoders successfully addresses many of the universal model's shortcomings. This thesis applies the modularized architecture and improves the University of Tartu translation system. Orders of magnitude larger dataset containing 7 languages is used to train the models compared to previous work. The modularized model achieves significantly higher BLEU scores than the University of Tartu model and the baseline universal model on all language pairs.

Keywords:
machine translation, multilingual machine translation, neural machine translation, neural networks, natural language processing

CERCS: P176, Artificial intelligence

Modulaarne seitsmekeelne neuromasintõlge

Lühikokkuvõte: Suur osa hiljutisi mitmekeelse neuromasintõlke süsteeme kasutab universaalset mudelit, mis jagab oma parameetreid kõikide keeltepaaride vahel. Tartu Ülikooli neuromasintõlke süsteem kasutab samuti universaalset mudelit ning on seega mõjutatud selle puudustest, näiteks madalast mudeli võimekusest keelepaari kohta. Eelnevad tööd on näidanud, et keelespetsiifiliste kodeerijate ja dekodeerijatega modulaarse mudeli kasutamine parandab edukalt mitmeid universaalse mudeli puuduseid. Selles töös rakendatakse Tartu Ülikooli neuromasintõlkemudeli parandusena modulaarset mudelit, kasutatades võrreldes eelnevate töödega mudeli treenimiseks suurusjärgu võrra suuremat seitset keelt hõlmavat andmestikku. Leiti, et modulaarne mudel saavutab oluliselt parema tõlkekvaliteedi kõigi keelepaaride vahel kui senine Tartu Ülikooli mudel ja võrdluseks treenitud universaalne mudel.

Võtmesõnad:
masintõlge, mitmekeelne masintõlge, neuromasintõlge, tehisnärvivõrgud, loomuliku keele töötlus

CERCS: P176, Tehisintellekt

Contents

1 Introduction

As the world is becoming more interconnected, having trustworthy multilingual machine translation is essential for countless companies, governments, and individuals. This has led multilingual machine translation to become an important topic to researchers and the industry alike. A paradigm shift took place in machine translation during the previous decade, with neural machine translation (NMT) replacing statistical machine translation as the most widely used approach. Since then multiple approaches to developing multilingual NMT models have emerged.

Currently, the most widely used approach is achieved using universal many-to-many models (Johnson et al., 2016). However, alongside its many benefits, it also has drawbacks. One of them is poor maintainability: to add a new language, it is necessary to retrain the existing model, which could affect the translation quality of the existing language pairs. Furthermore, to increase the model capacity, the whole model size needs to be increased, causing translation cost to increase. The option of having a separate one-to-one model for each language pair also has its downsides, such as the inability to fully utilize all the available data, which harms low-resource translation quality and makes zero-shot translation impossible. These issues have led researchers to propose a modularized approach with language-specific encoders and decoders (Escolano et al., 2020). This modularized approach has shown promising results in improving the drawbacks of the universal and single one-to-one models, making it a great candidate for use in the industry (Lyu et al., 2020).

This thesis aims to improve the translation quality of the University of Tartu NMT system by developing a new NMT model with the modular approach. In addition to the existing model, a baseline universal many-to-many model and three single one-to-one models are trained to compare the approaches in a more general sense in the 7-language setting used by the University of Tartu NMT model. The previous works (Lyu et al., 2020; Escolano et al., 2020; Escolano, Costa-Jussà, and Fonollosa, 2019) have looked at the model with a dataset size of up to a few million sentence pairs; however, this thesis analyzes the modular model's translation quality with orders of magnitudes more training data – a scenario that better reflects the performance of the approach in a practical, real-world scenario.

The main questions answered by this thesis are:

1. Can a model trained with the modular approach achieve better translation quality than the currently used universal University of Tartu NMT model?

2. How do modular models of different sizes compare to a baseline universal and single one-to-one models in terms of translation quality when trained with a large unbalanced septilingual dataset?

3. Is gradient accumulation a necessary part of training the modular model?

The contributions of this thesis are the following:

- a trained modular multilingual NMT model that can serve as the improvement to the current University of Tartu NMT model,
- a detailed analysis of the modular model in a scenario with a large multilingual unbalanced dataset,
- insights into the use of gradient accumulation in modular model training,
- a detailed description of an effective and flexible training process.

In addition to the Introduction, the thesis consists of 6 more sections with the following contents:

- The second section (Technical Background) introduces the most important theoretical concepts used in this thesis and provides the background needed to understand the related works and the rest of the thesis.
- The third section (Related Works) gives an overview of the common approaches in multilingual neural machine translation, including the recent work on the modularized architecture which this thesis is based on. This section helps the reader understand the background of the problem this thesis solves.
- The fourth section (Approach) gives an overview of the approach used and the reasoning behind it.
- The fifth section (Results and Analysis) analyzes the results the models produce. The quantitative analysis offers the most important results of this thesis. There are also selected examples in the qualitative analysis subsection, which demonstrate some of the differences in the models' translations.
- The sixth section (Conclusion) gives an overview of the key findings of this thesis.
- The seventh section (Future Research) discusses new questions that this thesis has raised and directions for future research on the topic of modular multilingual neural machine translation

2 Technical Background

2.1 Transformers

The Transformer architecture (see Figure 1) proposed by Vaswani et al. (2017) has left behind previously common recurrent neural network (RNN) and convolutional neural network (CNN) based architectures and assumed the place of the most prominent neural machine translation architecture. In their paper, Vaswani et al. (2017) demonstrated the Transformer's superior translation quality in machine translation. Since then, some version of it has been used in almost all recent neural machine translation systems. In addition to machine translation, it has also become widely used in language modeling, time series modeling, text-to-speech, and image processing tasks, to name a few.

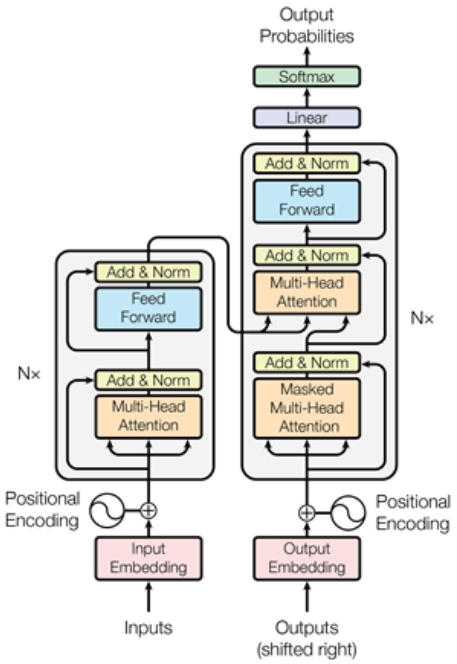


Figure 1. The Transformer architecture (Vaswani et al., 2017).

Neural machine translation models typically follow the encoder-decoder architecture, where the model consists of an encoder that calculates an embedding based on the input and a decoder that receives the encoder's output embeddings and generates the model's output. Furthermore, the models used in NMT are usually auto-regressive, meaning that in addition to the encoder output, the decoder takes its own output at the previous timestep as the input at the current step to generate the output. The Transformer is no different in that sense. However, unlike the previous approaches, the Transformer does not rely on convolutional or recurrent neural networks, thanks to multiple novel

approaches introduced by Vaswani et al. (2017). One of them is self-attention – each position in the input connects to all the other visible positions to calculate an embedding for itself. The specific attention implementation used was dot product attention, which was scaled to keep gradients from getting too small. The attention mechanism is present in various layers in the Transformer: as self-attention in the encoder and the decoder and as encoder-decoder attention in the decoder.

Self-attention alone is not enough to replace recurrence as, by itself, it does not have nor provide any information about the position of tokens in the input sentence. Vaswani et al. (2017) solved this issue by adding positional embeddings to each of the input embeddings. The authors described multiple benefits of their approach over RNNs and CNNs:

- In cases where the input length is smaller than the embedding size, it is equal or better in terms of computational complexity per layer.
- It has superior parallelizability compared to RNNs since it has no recurrent operations.
- It can learn long-range dependencies better than RNNs and CNNs by having a constant maximal path length between any two input tokens.

2.2 Byte Pair Encoding

Machine translation is a problem with no finite vocabulary of possible input and output words, but a vocabulary with a predefined size is used in practice. One of the simplest ways to create a vocabulary is to choose a certain number of most frequent words to form the vocabulary. However, this has some drawbacks. Let us consider the Estonian word *kuulilennuteetunneliluuk* (translates roughly to the bullet's flight trajectory tunnel's hatch) as an example. Since it is a word that is rarely used in everyday language, it likely will not be found in the vocabulary composed in the aforementioned way. If the word would be split into *kuuli*|*lennu*|*tee*|*tunneli*|*luuk*, then the input would contain subwords that by themselves have meaning and are used relatively frequently. When split into subwords this way, the model would have tokens that have meaning and which the model can use to output a usable translation, whereas only using the original word, it would probably not be present in the vocabulary and get replaced by an unknown word token; thus the model would get no information from that word. This is one of the main reasons that motivated Sennrich, Haddow, and Birch (2015) to adapt the Byte Pair Encoding (BPE) algorithm for word segmentation.

In its original form, BPE is a compression algorithm, which finds the two most frequent bytes, replaces them with a byte that's not present in the data, and repeats the process until it is no longer possible to further compress the data (Gage, 1994). Instead of bytes, the segmentation-algorithm proposed by Sennrich, Haddow, and Birch (2015)

uses characters and after each word an end-of-word symbol is inserted so that the original text can be recreated from the segmented text.

2.3 BLEU Evaluation Metric

Evaluating translation quality is an irreplaceable step in the development of machine translation models. Simply using human evaluators for assessing the quality of machine translation output is possible at a small scale; however, it is much more expensive and time-consuming than automatic evaluation. For this reason, it is necessary to have an automatic evaluation metric that measures how well a machine translation output matches the human translated reference. While there are many options for such evaluation metrics, BLEU (Bilingual Evaluation Understudy), proposed by Papineni et al. (2001), is the most widely used.

BLEU compares machine translation output to one or more reference translations by using a modified n-gram precision (Papineni et al., 2001). As the modification to the n-gram count, the authors proposed using candidate n-gram count, which is clipped by a maximum count of the n-gram in any of the references and then divided by the candidate's total n-gram count. This modified precision measure penalizes n-grams that are repeated too many times and too long candidates in general. In addition to too long candidates, too short candidates need to be penalized as well, which is why a brevity penalty, calculated using the candidate and reference lengths, was introduced to the score (Papineni et al., 2001).

Papineni et al. (2001) demonstrated that BLEU highly correlates with human evaluation when calculated for a whole corpus. Furthermore, it does not require any language-specific resources, as some other metrics such as METEOR (Lavie and Denkowski, 2009) and BERTScore (Zhang et al., 2019) do, making it simple to apply, especially for multilingual and low-resource models.

3 Related Works

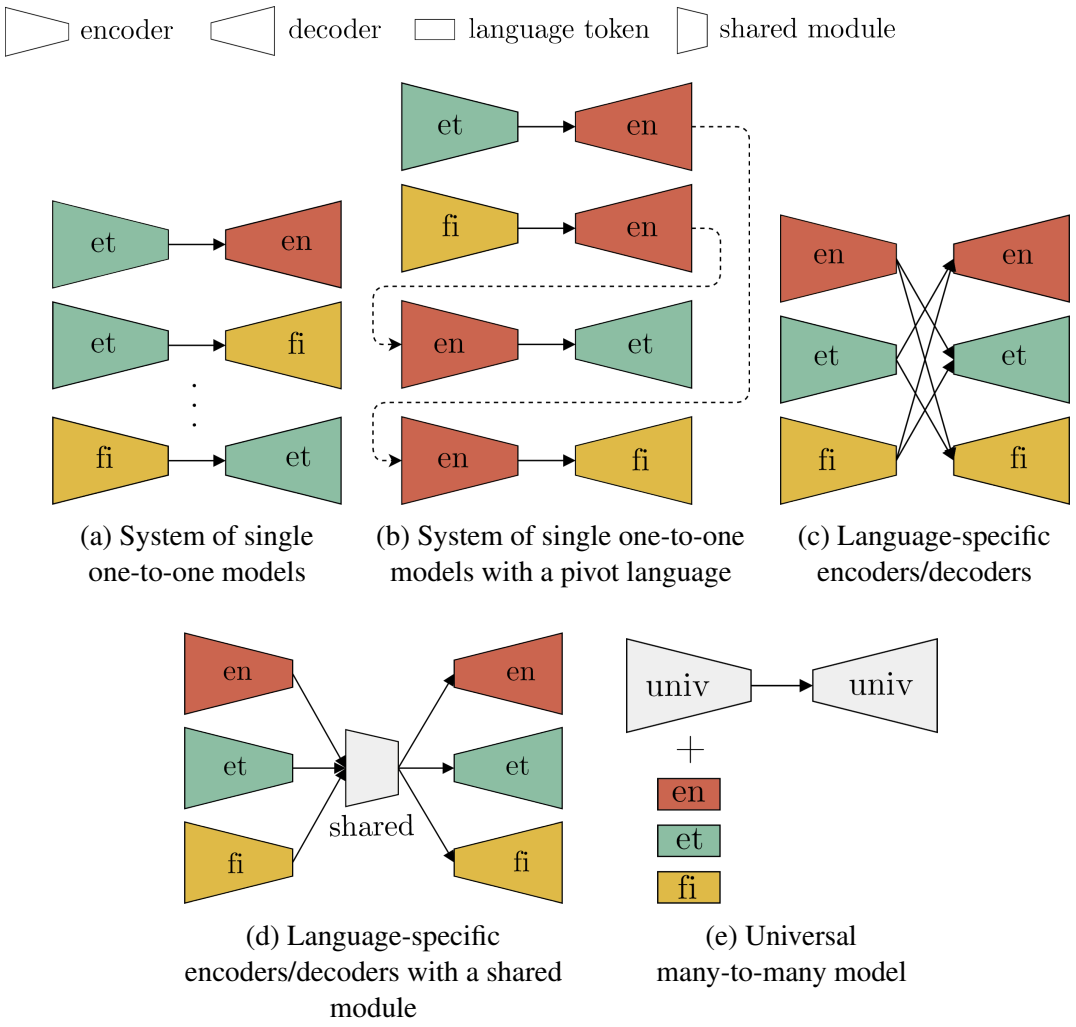
There exist many ways in which a many-to-many multilingual neural machine translation system can be trained. The system can have various degrees of parameter sharing across language pairs, from fully shared to no parameters shared. This thesis mainly compares the fully shared approach (Johnson et al., 2016) and partially shared approach proposed by Escolano et al. (2020) and further investigated by Lyu et al. (2020). This section gives an overview of the benefits and drawbacks of some of the common approaches (shown in Figure 2) in multilingual neural machine translation.

3.1 Universal Many-to-many Model

The universal many-to-many model with fully shared parameters is the most common approach to multilingual neural machine translation. In this approach, an encoder and a decoder are shared for all the language pairs, and during training time, a token is added to the input to indicate which language the source sentence is translated into (Johnson et al., 2016). It is the most popular approach for good reasons: it is relatively simple to train, and it enables zero-shot translation as demonstrated by Johnson et al. (2016). Some shortcomings of this approach can be poor modularizability and maintainability: when such a model is trained, adding new languages or modifying the existing languages requires fine-tuning or retraining the whole model, and it will affect the languages that are already trained. One of the issues identified for the shared model is that as the number of languages grows, there will be an increasing problem with the model's capacity: the translation quality of some of the language pairs starts to deteriorate with the increase in the number of languages (Zhang et al., 2020; Arivazhagan et al., 2019).

3.2 System of One-to-one Models

The system of one-to-one models can be considered the simplest approach in multilingual machine translation. In this system, no parameters are shared across languages or language pairs. Compared to the universal many-to-many models, they are very modular: training or modifying the models can be done separately since they are not dependent on one another. It is worth noting that there is a potential drawback to the modularity: when dealing with a many-to-many system, adding a new language to the existing N languages requires training $2N$ new models (translating to and from each existing language). For a system with many languages, this can grow out of control and become unmaintainable. Quite often, it is necessary to translate between low-resource language pairs, and for this type of system, their translation quality will be low since parameters are not being shared, and thus there will be no transfer learning across languages or language pairs. Furthermore, if there are language pairs with no parallel data available, it is impossible to translate between them without modifying the system.



(a) System of single one-to-one models

(b) System of single one-to-one models with a pivot language

(c) Language-specific encoders/decoders

(d) Language-specific encoders/decoders with a shared module

(e) Universal many-to-many model

Figure 2. The main types of multilingual NMT models demonstrated with a many-to-many system containing Estonian (en), Finnish (fi), and English (en). An extended version of Figure 1 from Lyu et al. (2020).

Using the system of one-to-one models with a pivot language (Habash and Hu, 2009) solves many of the shortcomings of the plain one-to-one models. In this case, a pivot language is chosen, which is used to translate between other languages. For example, in a system with English as the pivot language, when translating from Estonian to Finnish, Estonian would be first translated to English, and then the English translation would be translated to Finnish. This system is also very modular and more maintainable. To add a new language to this system, only two new models need to be trained: from the added language to the pivot language and vice versa. Furthermore, the models of this system can be trained independently, and it is possible to translate between low-resource language pairs, given that the pivot language is chosen so that there is plenty of data for training the models between pivot language and all the other languages.

A drawback of this system is that it is not possible to learn from the data available for the language pairs that do not contain the pivot language. Furthermore, translating through two models means that it will be computationally more expensive, errors might propagate, and there could potentially be a loss of information. It should be noted that there are attempts to overcome some of those limitations (Cheng et al., 2016), however, those are out of the scope of this thesis.

3.3 Language-specific Encoders and Decoders

Language-specific encoders and decoders are one approach to overcome the shortcomings of one-to-one and universal models. With this approach, it is not necessary to have a model for each language pair: instead, there is an encoder and a decoder for each of the languages. For example, when translating from Estonian to English, the encoder for Estonian and the decoder for English are being used. This approach can further be divided into two implementations based on degrees of parameter sharing: the model could have partially shared parameters or no parameter sharing between languages.

3.3.1 Partially Shared Parameters Between Languages

Previous research has attempted to train a system with separate encoders and decoders for each language and a shared attention-based mechanism, which alleviates the one-to-one model's capacity issue (Firat, Cho, and Bengio, 2016; Vázquez et al., 2018). There is also a recent approach where the decoders and lower layers of the encoders are language-specific, and the top layers of the encoder are shared (Liao et al., 2021). The authors trained the translation task jointly with the denoising auto-encoder task. Their work has shown that the proposed methods improve zero-shot translation while also not harming the overall translation quality compared to the universal model approach. The proposed approach also supports adding languages without retraining the whole model.

3.3.2 No Parameter Sharing Between Languages

The recent works have proposed having no parameter sharing between languages by having separate encoders and decoders for each language (Lyu et al., 2020; Escolano et al., 2020). Although the model has separate encoders/decoders for each language, they are trained together, which means that there is a shared intermediate representation for the languages. These shared representations allow for transfer learning between language pairs even though no parameters are explicitly shared between them.

The architecture addresses the problems of modularity and maintainability: adding new languages to the model does not affect the existing languages and only requires training a new encoder and decoder with the rest of the model frozen. Having separate encoders and decoders also means that the model is affected less by the capacity bottleneck. This is confirmed by the results by Lyu et al. (2020), which show that models trained with this approach obtain competitive results compared to the universal and the one-to-one approach.

While this architecture has many parameters during training due to a separate encoder and decoder for each language, during translation between two languages, the number of parameters is relatively low since only one encoder-decoder pair is used. This means that the model capacity can be increased without harming the inference time, which is important in industry, where translations need to be provided to customers without significant delay.

Because of these benefits and the promising results demonstrated by Lyu et al. (2020), this modular approach will be explored in this thesis as the potential improvement and the successor to the University of Tartu neural machine translation system, which currently uses a universal model.

4 Approach

Three model architectures will be compared to determine the effect of the modular approach on translation quality. This thesis mostly focuses on the comparison of the universal and modular models. Additionally, three single one-to-one models are trained for a high-resource (Estonian–English), a medium-resource (Lithuanian–English), and a low-resource (Latvian–Russian) language pair. In addition, the University of Tartu NLP Group's NMT model[1] (from now on referred to as UT Model or UT NMT Model) will be used for comparison.

4.1 Dataset

The previous approaches have used datasets of a few million pairs for training the models (Escolano, Costa-Jussà, and Fonollosa, 2019; Escolano et al., 2020; Lyu et al., 2020), however, the dataset used in this thesis is significantly larger and consists of multiple domains. A dataset compiled from publicly available corpora and cleaned by the University of Tartu NLP Group is used for training and evaluating the results. It was constructed by combining publicly available multilingual parallel corpora for 7 languages from various domains. The corpora used were JRC-Acquis (Steinberger et al., 2006), ParaCrawl (Esplà-Gomis et al., 2019), OpenSubtitles (Lison and Tiedemann, 2016), Europarl (Koehn, 2005), DGT (Steinberger et al., 2012), News-Commentary (Tiedemann, 2012), and MultiUN (Tiedemann, 2012).

The dataset contains over 292 million sentence pairs in total for the 42 language pairs. The languages used were Estonian, Finnish, Latvian, Lithuanian, English, German, and Russian, and the model was trained to translate in a many-to-many scenario, where it is possible to translate between all 42 language pairs. The dataset was split into train, development, and test set (referred to as the test set or the internal test set from now on). The amount of training data for each language pair can be seen in Table 1 and a more detailed overview for train, development, and test sets can be found in the appendix in Tables 12, 13, and 14. Since Estonian translations are essential for the UT NLP Group and also from the perspective of this thesis, Estonian is the largest language in the dataset by the number of sentence pairs.

[1] https://koodivaramu.eesti.ee/tartunlp/translate/, deployed on https://neurotolge.ee/

Table 1. The total number of sentences for each language pair in the training set.

	Target							
Source	en	de	fi	lt	et	lv	ru	Total
en	-	8.2M	8.0M	5.9M	12.1M	4.9M	8.2M	47.4M
de	8.2M	-	6.5M	7.0M	12.0M	6.6M	6.3M	46.6M
fi	8.0M	6.5M	-	6.1M	12.6M	5.8M	6.5M	45.6M
lt	5.9M	7.0M	6.1M	-	7.4M	6.9M	944.9K	34.3M
et	12.1M	12.0M	12.6M	7.4M	-	7.3M	6.6M	58.0M
lv	4.9M	6.6M	5.8M	6.9M	7.3M	-	428.9K	31.9M
ru	8.2M	6.3M	6.5M	944.9K	6.6M	428.9K	-	29.0M
Total	47.4M	46.6M	45.6M	34.3M	58.0M	31.9M	29.0M	292.8M

As an additional, out-of-domain test set, the news translation task test data from WMT17 (Bojar et al., 2017) is used for Latvian↔English, test data from WMT18 (Bojar et al., 2018) is used for Estonian↔English, and test data from WMT19 (Barrault et al., 2019) is used for Finnish↔English, Lithuanian↔English, German↔English and Russian↔English.

4.2 Preprocessing

The BPE segmentation algorithm (Sennrich, Haddow, and Birch, 2015) from SentencePiece (Kudo and Richardson, 2018) was used for splitting the sentences into subwords.

SentencePiece is a subword tokenizer and detokenizer that provides a lossless and language-independent segmentation of the input (Kudo and Richardson, 2018). The BPE algorithm implemented in it by Kudo and Richardson (2018) is very efficient, having the computational complexity of $O(N \cdot log(N))$ (where N is the number of symbols), while the naive algorithm would have the complexity of $O(N^2)$. SentencePiece's high efficiency, losslessness, and language-independency are the main reasons it was chosen as the subword tokenizer in this thesis.

For the universal many-to-many model, a single SentencePiece BPE model was trained. The SentencePiece model was trained on 10 million sentences uniformly sampled from each language's training set so that each language has equal representation. The vocabulary size was chosen to be 32,000.

For the modular model, a SentencePiece BPE model with a vocabulary size of 16,000 was trained for each language, similar to Lyu et al. (2020). For training each of the models, a total of 10 million sentences were randomly sampled from each language's training set and then used to train the BPE model.

The single one-to-one models reuse the segmentation models from the modular approach.

4.3 Model Architecture

4.3.1 The Universal Many-to-many Model

For the universal many-to-many model, there is a single Transformer model (Vaswani et al., 2017) for all the language pairs. The architecture has an embedding size of 512, feed-forward neural network size of 2048, and both the encoder and decoder have 6 layers and 8 attention heads. These sizes are chosen because the University of Tartu machine translation model uses the same model size and is trained on a similar dataset.

4.3.2 The Modular Model

The modular model's architecture is based on Lyu et al. (2020) and Escolano et al. (2020). It uses a separate encoder and decoder from the Transformer architecture (Vaswani et al., 2017) for each of the languages. Each language shares all the input and output embedding weight matrices. All encoders and decoders have 6 layers and 8 attention heads.

Two model sizes were experimented with: the base architecture from Lyu et al. (2020) with embedding size of 256 and feed-forward neural network size of 1024, and a larger architecture (from now on referred to as the big modular model or modular big) used by both Lyu et al. (2020) and Escolano et al. (2020) with embedding size of 512 and feed-forward neural network size of 2048.

As shown in Table 2, the base modular model is comparable to the universal model in the total number of parameters, and the big modular model is comparable in parameters used during translating between two languages (inference-time).

Table 2. Number of parameters for the models used in this thesis.

Model	Total Parameters	Inference-time Parameters
universal	94.9M	94.9M
single one-to-one	60.5M	60.5M
modular base	106.1M	19.3M
modular big	366.3M	60.5M

4.3.3 Single One-to-one Models

The single one-to-one models have the architecture equivalent to using a single encoder and decoder from the modular big architecture.

4.4 Training

Fairseq sequence modeling toolkit (Ott et al., 2019) was used to train and implement the models. Fairseq has wide support among the research community, with implementations of the latest papers frequently added. Furthermore, Fairseq is implemented in PyTorch, making it easier to add custom components to it since PyTorch is one of the most widely used deep learning frameworks.

Adam optimizer (Kingma and Ba, 2014) with inverse square root scheduler from Fairseq was used with 4000 warm-up steps and a learning rate of 0.0008. Label smoothed cross-entropy with label smoothing of 0.1 was used as the loss function. Dropout with the probability of 0.1 and weight decay of 0.0001 was used. The models are trained with a batch size of 15,000 tokens per GPU with 4 GPUs. Mixed precision training was used for all the models.

All of the models were trained for 20 epochs. This is because training until full convergence with the large amount of data and the limited availability of computation resources would take too long: the training times for a model with the current setup is 2–3 weeks (except for the single one-to-one models) and training until convergence would increase the time to train all models by a few months. Still, the results for 20 epochs provide enough to conclude if the modular approach is feasible for the setting explored in this thesis.

4.4.1 The Universal Model

Training the universal model differs from the other models by adding a target language token to the tokenized source language sentence before passing the sequence to the model. This is necessary to indicate which language the model should translate into. An example of the preprocessing can be seen in Table 3, where the third step is the input to the universal model, while the second step is used as the input for the other models.

Table 3. An example of preprocessing steps for the universal model when translating from English into Estonian.

1. raw sentence	When can I restart sports activities?
2. byte-pair encoded sentence	_When _can _I _rest art _s ports _activities ?
3. language token added	<2et> _When _can _I _rest art _s ports _activities ?

4.4.2 The Modular Model

With a different encoder and decoder for each language, the batches need to be constructed in a way where each batch contains data from only a single language pair so that

the correct encoder-decoder pair can be selected for the batch. Several ways of achieving this were tried:

- Oversampling the language pairs with fewer sentences to match the size of the language pair with the most sentence pairs, iterating over the language pairs in a round-robin way, and creating a batch for each language pair. At the same time, gradients are being accumulated for each pass over the language pairs. This will be referred to as the round-robin oversampled approach (abbreviated as OS).

- Constructing the batches from each language pair so that they contain samples from only that language pair. The batches of the language pairs are concatenated and shuffled to obtain the final training batches. Multiple experiments are conducted, training with and without gradient accumulation. This will be referred to as the proportional sampling approach (abbreviated as PS).

The round-robin oversampled approach is already implemented in Fairseq as the *multilingial_translation* task, but results by Lyu et al. (2020) have shown that this might not perform as well as training with proportional sampling. The proportionally sampled modular model also offers a more fair comparison to the universal model since it does not oversample some of the language pairs, and the models process the same amount of data per epoch. For this reason, a new dataset extension and a task to support it were implemented for Fairseq for the proportional sampling approach.

The dataset extends Fairseq's *SampledMultiDataset* and batches the data indices as described in Algorithm 1. In essence, much of the existing implementation from Fairseq is used, which determines the inputs and outputs of the algorithm. The main change compared to the existing implementations is the batch creation. With the described algorithm, any batch will contain samples only from a single language pair since batches are constructed separately for language pairs and then concatenated. The created batches are then shuffled before iterating over them during training. The training task extends Fairseq's *multilingual_translation* task. When iterating over batches during training, the encoder and the decoder are chosen based on the source and the target language of the batch, respectively. It should be noted that this implementation allows for various sampling methods already implemented in Fairseq (for example, temperature sampling) to be used, making it a much more flexible solution than the default *multilingual_translation* task.

Lyu et al. (2020) also used gradient accumulation but reduced the batch size per GPU to keep the overall batch size the same. However, this is less effective performance-wise. Ott et al. (2018) used gradient accumulation to train NMT models with relatively large batch size, and they found no decrease in translation quality and noticed a reduction in training times. For this reason, the batch size per GPU is not being decreased in this thesis when the gradient accumulation is used. For the modular models with gradient accumulation, gradients of 10 batches are accumulated before updating.

Algorithm 1: Batch Construction for Proportional Training

input :

indices – indices for the main dataset, which is concatenated from the individual language pair datasets

datasets – list of all the language pair datasets used to construct the main dataset

getDatasetIndex – given an index of the sentence, returns the index of the dataset in the datasets list

constructBatches – constructs batches from the given indices

output :

batches – list of batches where each batch is a list of sentence indices in the current dataset

```
// create a list of the n empty lists where n is the number of
   datasets
indicesByDataset := list()
foreach dataset ← datasets do indicesByDataset.append(list())
// separate indices by dataset
for i ← indices do
    datasetIndex := getDatasetIndex(i)
    indicesByDataset[datasetIndex].append(i)
// create batches for each dataset separately and concatenate
   all the lists of batches
batches := list()
for datasetIndices ← IndicesByDataset do
    datasetBatches := constructBatches(datasetIndices)
    batches.extend(datasetBatches)
```

4.4.3 Single One-to-one Models

Since the single one-to-one models use datasets with different sizes for training, they are trained each in a different way. The Estonian–English model (high resource) is trained with the same batch size as the universal model. The Lithuanian-English model has a total batch size 2 times lower than the Estonian–English model. The Latvian–Russian model has 4 times lower total batch size since it is trained on only 1 GPU. Additionally,

for the Latvian–Russian model, the warmup steps are reduced to 1500.

4.5 Hardware

The models were trained at the University of Tartu HPC Center (University of Tartu, 2018) using 2 or 4 NVIDIA Tesla V100 GPUs depending on availability, except for the low-resource single one-to-one model where 1 GPU is used. When 2 GPUs were used for training the universal or modular models instead of 4, the gradients were accumulated to make the batch size the same as with 4 GPUs.

4.6 Evaluation

BLEU (Papineni et al., 2001) is used to evaluate translation quality since it is the most widely accepted translation quality metric. SacreBLEU (Post, 2018) is used as the implementation for calculating the scores. For evaluation, the following procedure is used:

1. the test set is tokenized with the appropriate SentencePiece BPE model,
2. the tokenized text is translated with the trained model using beam size of 5,
3. the translation is detokenized with SentencePiece,
4. the detokenized text is compared with the reference text using SacreBLEU to obtain the BLEU score.

5 Results and Analysis

5.1 Quantitative Analysis

5.1.1 Modular Training Approaches

Training the model without gradient accumulation was determined to be unfeasible, and the training was canceled before reaching 20 epochs since it could be easily observed that the loss and perplexity decrease plateaued early at high values compared to the experiment with gradient accumulation (see Figure 3). For this reason, further modular models are all trained with gradient accumulation.

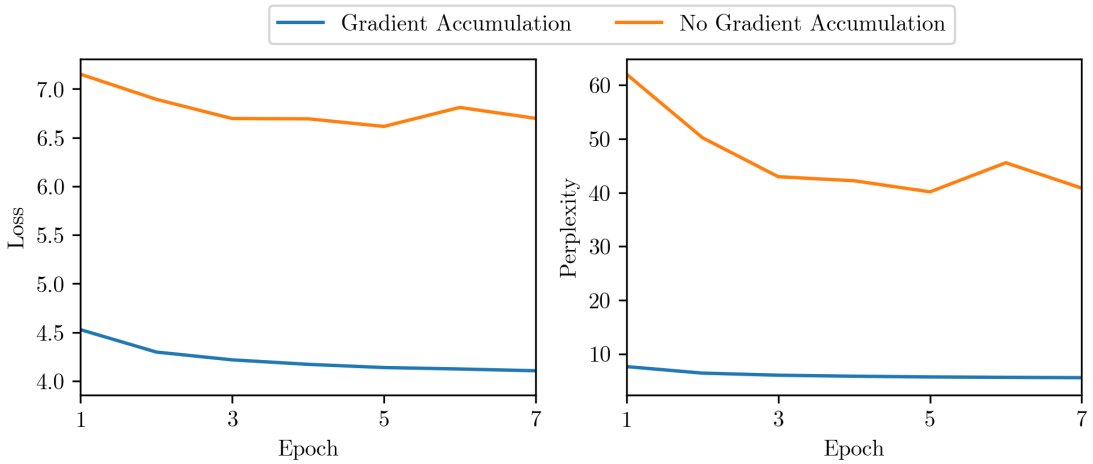


Figure 3. Comparison of the development set metrics (lower is better) after each training epoch for the proportional sampling approach with gradient accumulation and without gradient accumulation.

When comparing the two approaches of training the modular models, it can be noted that, on average, the round-robin oversampling approach performs better than the proportional sampling approach with a difference of 0.27 points in the average BLEU score on the internal test set (see Table 4, for full results see Tables 16 and 17 in the appendix). Due to oversampling, the biggest difference can be seen in low resource pairs such as Russian to Latvian, where the difference is 3.5 BLEU points in favor of the oversampled model. However, the oversampled model performs worse for many high resource pairs, especially Estonian ones, which are important from the perspective of this thesis. It should also be noted that the comparison is biased to favor the oversampled approach because, with the oversampling approach, the model sees more data per epoch for lower resource language pairs. Furthermore, the oversampled apporach is also less flexible than the proportional approach, allowing no control over the sampling approach

and minimal control over gradient accumulation, making it less suitable for further research. Thus the big modular model was trained using the proportional sampling approach.

Table 4. Modular models (base) internal test set BLEU score difference: proportional model's gain over the round-robin model. The bottom rightmost cell contains the average gain over all the language pairs.

	Target							
Source	en	de	fi	lt	et	lv	ru	Average
en	-	-0.2	0.42	-0.16	0.55	-1.62	0.63	-0.06
de	0.56	-	0.05	0.79	1.29	-0.36	0.71	0.51
fi	0.3	-0.36	-	0.39	0.57	-0.15	-0.29	0.08
lt	-0.58	-0.13	-0.38	-	0.24	-0.61	-0.69	-0.36
et	0.19	0.44	-0.31	-0.18	-	-0.28	0.28	0.02
lv	-0.87	-0.41	-0.7	0.47	-0.55	-	-4.63	-1.11
ru	1.07	0.27	-1.44	-3	0.9	-3.5	-	-0.95
Average	0.11	-0.06	-0.39	-0.28	0.50	-1.09	-0.66	-0.27

5.1.2 Comparison of the Modular Model and the Universal Model

The results show that the base models trained with round-robin oversampling and proportional sampling are underperforming compared to the universal model (for results see Table 15 in the appendix) by 1.36 and 1.63 BLEU points on average, respectively. This is most likely because of the modular models' low embedding and feed forward neural network size causing a capacity bottleneck. To address this, a bigger modular model was trained.

The big modular model has achieved significantly better scores (see Table 5) than the base modular models and the baseline universal model. The big modular model's translation quality is an improvement over the universal model on all the language pairs, and on average, the translation quality is 2.32 BLEU points higher (see Table 19 in the appendix).

Figure 4 shows the differences between the training progress of the modular model and the universal model, where it can be observed that the modular model converges slower than the universal model, but it achieves better results by the 20th epoch. The slower convergence is not unexpected, and it is likely not caused by the architecture but by the much larger batch size that results from gradient accumulation in the modular

Table 5. Internal test set BLEU scores of the big modular model.

	Target						
Source	en	de	fi	lt	et	lv	ru
en	-	43.43	35.74	43.93	41.36	44.55	33.28
de	51.17	-	28.54	32.74	37.16	37.1	38.88
fi	43.49	30	-	27.29	31.33	32.99	15.34
lt	52.23	34.69	26.92	-	34.94	41.65	22.1
et	51.39	38.53	32.66	35.24	-	39.74	20.31
lv	50.73	38.58	32.45	39.87	36.43	-	27.21
ru	41.73	39.88	12.89	22.53	23.73	21.7	-

model. This was also observed by Ott et al. (2018) where they used gradient accumulation to achieve more efficient training by having large batch sizes.

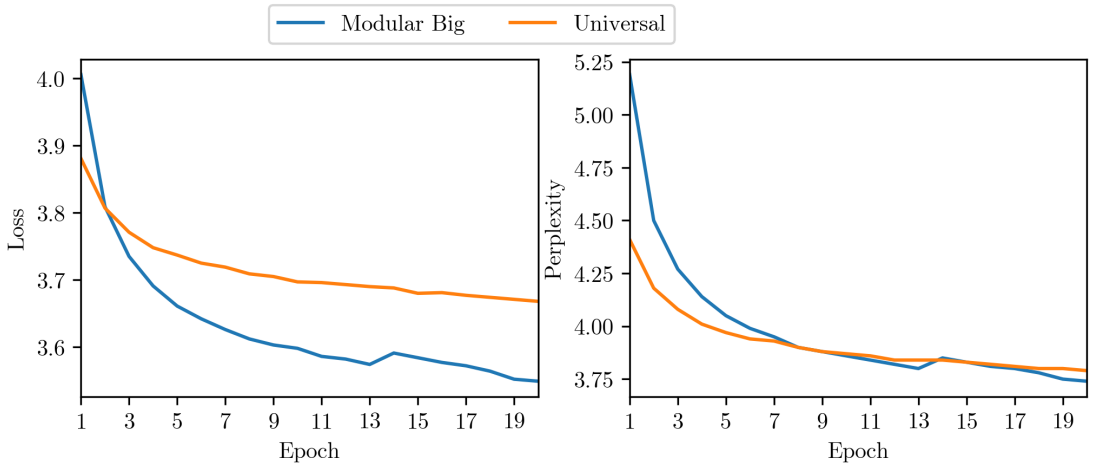


Figure 4. The big modular model's training metrics (lower is better) compared to the universal model's training metrics on the development set after each epoch.

5.1.3 Comparison of the Modular Model with the University of Tartu Model

The UT NMT model features translation into various domains, the most general of which would be news, which is chosen for the translations in this thesis. Compared to the UT NMT model (see the results in Table 18 in the appendix), the big modular model performs 7.26 BLEU points better on the internal test set on average. Since the UT model

is fine-tuned to the news domain, the comparison is not entirely fair. For this reason, the models were compared on the WMT news test set. The BLEU scores on the WMT test set show that the big modular model's translation quality is higher than the UT model by 2.85 BLEU points on average (Table 6). It can be seen that the big modular model achieves better translation quality on all the language pairs compared to the UT model and the universal baseline model.

Table 6. Comparisson of BLEU scores on the WMT test set.

		X						
Model	Direction	de	fi	lt	et	lv	ru	Average
UT Model	en → X	31.96	17.71	11.87	18.06	16.7	23.01	23.07
	X → en	28.63	25.46	29.63	25.89	20.95	26.96	
Universal	en → X	30.94	20.07	12.94	19.89	17.25	21.61	23.44
	X → en	30.04	25.25	28.83	26.22	20.74	27.55	
Modular Big	en → X	**35.65**	**22.08**	**14.18**	**22.34**	**19.2**	**26.55**	**25.92**
	X → en	**31.81**	**28.2**	**30.95**	**28.34**	**22.09**	**29.66**	

5.1.4 Comparison of the Modular Model with Single One-to-one Models

Three single one-to-one models were trained to compare the translation of languages with training sets of various sizes, since the amount of available training data has a significant effect on the translation quality of the trained model. The single one-to-one models can be viewed as a baseline for model capacity: worse translation quality than that of single one-to-one models indicates a capacity bottleneck in the multilingual model, while higher translation quality indicates a gain from transfer learning between language pairs. Since it was shown that the base modular model suffers from a capacity bottleneck, this section is focused on comparing the big modular model and the single one-to-one models.

On the internal test set, the Estonian-English (high-resource, 12.1M sentence pairs) model achieved a BLEU score of 52.27 (see the results in Table 7), which is a gain of 0.88 BLEU points over the big modular model. Conducting a paired bootstrap resampling significance test (Koehn, 2004) reveals that the p-value of the single model results compared to the big modular model is 0.0420. It is lower than the commonly used p-level of 0.05, indicating that the results might be significant enough to suggest that the single one-to-one model achieves better translation quality than the big modular model. However, when looking at the WMT test set results, the results of the single one-to-one and modular big model are almost the same with the gain of the one-to-one

model being 0.02 BLEU points over the big modular model. From all this, it is not possible to conclude with certainty, that the big modular model is experiencing a capacity bottleneck compared to the single model on the high-resource language pair. Further experiments and analysis is necessary to determine that.

On the internal test set, similar results are also observed with the Lithuanian-English (mid-resource, 5.9M sentence pairs) model with the single one-to-one model achieving 0.76 points higher BLEU score than the big modular model on the test set. In contrast, on the WMT test set the big modular model achieves 0.28 BLEU points higher result than the one-to-one model. These results are not enough to confidently say that one of the models performs better than the other.

When looking at Latvian-Russian, the lowest resource language pair in the dataset (428.9K sentence pairs), it is clear that that the models featuring parameter sharing to some degree perform much better, with the big modular model achieving the best results with a gain of 16.54 BLEU points over the single one-to-one model. This demonstrates clearly that the big modular model performs significantly better on low-resource language pairs compared to the single one-to-one model. This can mostly be attributed to transfer learning in the modular model.

Table 7. Internal test set BLEU scores for the single one-to-one models, universal many-to-many model, big modular model and the University of Tartu Model.

	Model			
Lang. pair	Single 1-to-1	Modular Big	Universal	UT Model
et-en	**52.27**	51.39	49.02	41.9
lt-en	**52.99**	52.23	48.09	42.16
lv-ru	10.67	**27.21**	22.98	23.22
et-en (WMT)	**28.36**	28.34	26.22	25.89
lt-en (WMT)	30.67	**30.95**	28.83	29.63

5.2 Qualitative Analysis

Since the model contains 7 languages (42 translation directions), it is not feasible to analyze all of them for this thesis. Estonian–English translation direction is chosen for the qualitative analysis, since the English translations will be understandable for all readers of this thesis, and this language pair has all of the discussed model types trained for it. The analysis is done by comparing translations of four sentences picked with the goal to highlight the differences between the models. It should be noted that this analysis

is too limited to draw any general conclusions and only serves to showcase some of the errors the models make.

In the first example shown in Table 8, it can be observed that only the big modular model and the single one-to-one model can translate the sentence correctly. The other models made errors translating *kelkudest* (meaning *of sleds*) and/or *suuskadest* (meaning *of skis*). For all the models, the start of the sentence is slightly different from the reference, however, it is still a perfectly valid translation of the source sentence.

Table 8. The first sentence (from the WMT test set) translated by the models discussed in this thesis.

Source	Rääkimata purunenud kelkudest ja suuskadest.
Reference	To say nothing of broken sleds and skis.
Modular Big	Not to mention broken sleds and skis.
UT Model	Not to mention the broken clocks and mouthpieces.
Single 1-to-1	Not to mention broken sleds and skis.
Universal	Not to mention broken skiing and skiing.
Modular Base (Oversampled)	Not to mention broken circles and skis.
Modular Base (Proportional)	Not to mention the broken coats and the skirts.

The second sentence highlights how some of the models have interpreted a part of the sentence in an incorrect way. In Table 9 it can be seen that *Eelmisel suvel oli lava teistpidi* (*Last summer, the stage faced the other way*) is interpreted incorrectly by the UT model, the universal model, and the modular base models. The UT model and the modular models translated it as there being a different stage and the universal model translated it as the stage being on the other side. Almost all of the models translated the second part of the sentence identically, with the exception of the modular base oversampled model where an *and* is added.

Table 9. The second sentence (from the WMT test set) translated by the models discussed in this thesis.

Source	Eelmisel suvel oli lava teistpidi, siis kuulsime rohkem.
Reference	Last summer, the stage faced the other way. Then we heard more.
Modular Big	Last summer, the stage was the other way around, then we heard more.
UT Model	Last summer there was a different stage, then we heard more.
Single 1-to-1	Last summer, the stage was the other way around, then we heard more.
Universal	Last summer, the stage was on the other side, then we heard more.
Modular Base (oversampled)	Last summer, the stage was different, and then we heard more.
Modular Base (proportional)	Last summer, the stage was different, then we heard more.

When looking at the translations of the third sentence (Table 10), a number of differences can be observed between the translations of the models. The most common mistake that all models have, is either not translating the *teleobjektiiv* (meaning *telephoto lens*) correctly or not translating it at all. The Single one-to-one model and modular base oversampled model are the only two that have a translation of it in the sentence (although an incorrect one). The second common mistake is translating *lainurk* (meaning *wide-angle*). The Modular base models have both translated it as *lawn* or *lawnmower*, which is incorrect and totally unrelated to the the correct word. The universal model has translated it incorrectly as corner which can be explained by there being the same word for both corner and angle in Estonian. UT model has translated it as *wave angle*, which is still incorrect, but closer than the previous models. The only models that translated it correctly are the big modular model and the single one-to-one model. Another common mistake is incorrect translation of *field of view*, which is not present in the reference, but can be considered to be a correct translation. Only modular big and the universal model translate it correctly, while other models make mistakes such as translating it as *viewing field* or *field of vision*. Taking all that into consideration, the big modular model has produced the best translation out of the models analysed in this thesis.

Table 10. The third sentence (from the WMT test set) translated by the models discussed in this thesis.

Source	Telefonil peaks olema kaks kaamerat, millest üks on lainurk ja teine kitsama vaateväljaga teleobjektiiv.
Reference	The phone should have two cameras, of which one is wide-angle and the other one is a telephoto lens with narrower angle.
Modular Big	The phone should have two cameras, one with a wide angle and the other with a narrower field of view.
UT Model	The phone should have two cameras, one with a wave angle and the other with a narrower viewing field.
Single 1-to-1	The phone should have two cameras, one of which is a wide angle and the other is a TV lens with a narrower field of vision.
Universal	There should be two cameras on the phone, one of which is the corner and the other a lens with the narrower field of view.
Modular Base (OS)	There should be two cameras on the phone, one of which is a lawnmower and the other a narrower viewable television object.
Modular Base (PS)	There should be two cameras on the phone, one of which is a lawn and the other with a narrow field of vision.

In the modular model, the source and target languages need to be explicitly stated, whereas in the universal model only the target language needs to be specified. In some cases, the user can even give input to the universal model in multiple languages within the same sentence, and the model translates it successfully, while it is not possible for the modularized model due to language-specific encoders. While the automatic source language detection might be convenient, the single encoder brings some problems with it. In many languages, the same words have different meanings, and thus the universal model might automatically assign an incorrect meaning to them. For example, the Finnish word *ase* means *weapon* in English, but the Estonian word *ase* means *bed* in English. An example of this is shown in Table 11. We can see that the models with the universal architecture translated *ase* as *a gun*. The base modular models had trouble translating *nurgas on ase*, likely because of their low capacity. The Single one-to-one model and big modular model translated *ase* as *a place* which is incorrect, however it a much better option than translating it as *a gun*. The reason why the universal architecture models suggest a Finnish word might be that the Estonian word is rarer than the Finnish word. This might not happen in all such cases, since most of the time the model is able to imply the source language from the context. However, when a single word is translated it might

happen more frequently. This example demonstrates that in some cases using a modular architecture over the universal architecture results in a more trustworthy translation.

Table 11. Translations of a sentence chosen to demonstrate translation from incorrect language.

Source	Nurgas on ase, kuhu saab pikali heita.
Reference	There is a bed in the corner where you can lie down.
Modular Big	There's a place in the corner where you can lie down.
UT Model	There is a gun in the corner where you can lay down.
Single 1-to-1	There's a place in the corner where you can lie down.
Universal	There's a gun in the corner where you can lie down.
Modular Base (OS)	The nerve is a place to lie down.
Modular Base (PS)	There's a place where you can lie down.

6 Conclusion

This thesis has found that the big modular model offers superior translation quality compared to the baseline universal model and the UT NMT model both on the internal test set and WMT test set. In addition, the qualitative analysis offered insight into possible benefits of the modular model, such as not translating from an incorrectly inferred source language. It can be concluded that the thesis has accomplished its main goal of finding an improvement to the UT model. Considering the other benefits besides translation quality offered by the modular approach, such as the ability to add new languages without retraining and the inference time staying the same even though languages are added, and capacity is increased, it is a promising candidate for future research and application in production.

Based on the results of this thesis, the questions raised in the introduction can now be answered:

1. The big modular model trained in this thesis achieves significantly better translation quality than the University of Tartu model based on BLEU scores. This is confirmed by the qualitative analysis.

2. When comparing the universal model and the modular base model, which have roughly the same number of total parameters, the modular model offers worse translation quality of the two. However, when comparing the universal model to the big modular model, which has a larger total number of parameters but the same encoder and decoder sizes, the big modular model shows significantly better translation quality than the universal model. Compared to single one-to-one models, the base modular models achieve better translation quality only for the Latvian–Russian translation out of the three language pairs chosen for analysis. For other language pairs, the translation quality of the modular base model is significantly worse than of the single one-to-one models because of the modular base model's low capacity. However, the big modular model achieves results that are more competitive with the single one-to-one models: it achieves significantly better results for the Latvian–Russian translation direction and similar results for Estonian–English and Lithuanian–English directions. For the latter two directions, it can not be definitively concluded that one of the approaches performs better than the other.

3. Training with and without gradient accumulation was tried, and training without gradient accumulation was deemed unfeasible due to the subpar results it produced. In addition to being essential in the training process, gradient accumulation also makes training the models more computationally efficient.

7 Future Research

Although the thesis fulfilled its goal, its results have raised new questions. Comparing the modular model with the single one-to-one models revealed that the big modular model might suffer from a slight capacity bottleneck with high-resource languages, which suggests that by increasing the model size, it could be possible to achieve even better translation quality. To confirm that, further experiments are needed. The universal model very likely suffers from an even bigger capacity bottleneck given that it has significantly fewer total parameters and achieves lower translation quality than the single one-to-one model and the big modular model. It would be beneficial to see how a universal model with a larger capacity compares to the big modular model.

This thesis compared the models based on the inference-time parameters, but they could also be compared by the total number of parameters. The base modular model and the universal model had a relatively similar total number of parameters, but the universal model had the better average translation quality of the two. However, this does not prove that this would be the case in a scenario where both models have a larger model size. This is again something that would need to be verified by further experiments. Ultimately by what criteria the models should be compared is determined by the constraints under which the models are applied (for example, translation time).

The use of large batch sizes by not reducing the batch size per GPU has proven to be an approach that provides excellent results with the modular model. However, it has not been compared to training the model with keeping the overall batch size the same by reducing batch size per GPU when gradient accumulation is used. There is also a question of how changing the gradient accumulation frequency affects the training and the results of the modular model. Since this thesis showed that the modular model could not be trained without gradient accumulation, the gradient accumulation frequency likely has a larger effect on modular models than on universal models.

The UT model can translate to multiple domains, but the modular model was trained to translate in a general domain in this thesis. For further research, it would be beneficial to see how it performs when translating to multiple specific domains compared to the universal models. A massively multilingual scenario is also something the modular model has not been applied to, and it poses its own unique challenges compared to the 7-language scenario explored in this thesis. This would also be a topic that could be beneficial in the industry since many companies such as Facebook and Google are developing their own massively multilingual models based on universal models, and a solution based on the modular approach could potentially offer an improvement.

References

[1] Naveen Arivazhagan et al. “Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges”. In: (July 2019). URL: https://arxiv.org/abs/1907.05019.

[2] Loïc Barrault et al. “Findings of the 2019 Conference on Machine Translation (WMT19)”. In: *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 1–61. URL: http://www.aclweb.org/anthology/W19-5301.

[3] Ondřej Bojar et al. “Findings of the 2017 Conference on Machine Translation (WMT17)”. In: *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 169–214. URL: http://www.aclweb.org/anthology/W17-4717.

[4] Ondřej Bojar et al. “Findings of the 2018 Conference on Machine Translation (WMT18)”. In: *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*. Belgium, Brussels: Association for Computational Linguistics, Oct. 2018, pp. 272–307. URL: http://www.aclweb.org/anthology/W18-6401.

[5] Yong Cheng et al. “Neural Machine Translation with Pivot Languages”. In: (Nov. 2016). URL: http://arxiv.org/abs/1611.04928.

[6] Carlos Escolano, Marta R. Costa-Jussà, and José A. R. Fonollosa. “From Bilingual to Multilingual Neural Machine Translation by Incremental Training”. In: (June 2019). URL: http://arxiv.org/abs/1907.00735.

[7] Carlos Escolano et al. “Multilingual Machine Translation: Closing the Gap between Shared and Language-specific Encoder-Decoders”. In: (Apr. 2020). URL: http://arxiv.org/abs/2004.06575.

[8] Miquel Esplà-Gomis et al. “ParaCrawl: Web-scale parallel corpora for the languages of the EU”. In: *Proceedings of Machine Translation Summit XVII* 2 (2019).

[9] Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. “Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism”. In: (Jan. 2016). URL: http://arxiv.org/abs/1601.01073.

[10] P Gage. “A new algorithm for data compression”. In: *The C Users Journal archive* 12 (1994), pp. 23–38.

[11] Nizar Habash and Jun Hu. “Improving Arabic-Chinese statistical machine translation using English as pivot language”. In: (Mar. 2009), pp. 173–181. DOI: 10.3115/1626431.1626467.

[12] Melvin Johnson et al. “Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation”. In: (Nov. 2016). URL: http://arxiv.org/abs/1611.04558.

[13] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: (Dec. 2014). URL: https://arxiv.org/abs/1412.6980.

[14] Philipp Koehn. “Europarl : A Parallel Corpus for Statistical Machine Translation”. In: *MT Summit* 11 (2005).

[15] Philipp Koehn. “Statistical significance tests for machine translation evaluation”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing* 4 (2004).

[16] Taku Kudo and John Richardson. “SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing”. In: (Aug. 2018). URL: https://arxiv.org/abs/1808.06226.

[17] Alon Lavie and Michael J. Denkowski. “The METEOR metric for automatic evaluation of Machine Translation”. In: *Machine Translation* 23.2-3 (2009). ISSN: 09226567. DOI: 10.1007/s10590-009-9059-4.

[18] Junwei Liao et al. “Improving Zero-shot Neural Machine Translation on Language-specific Encoders-Decoders”. In: (Feb. 2021). URL: http://arxiv.org/abs/2102.06578.

[19] Pierre Lison and Jörg Tiedemann. “OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles”. In: *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*. 2016.

[20] Sungwon Lyu et al. “Revisiting Modularized Multilingual NMT to Meet Industrial Demands”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 5905–5918. DOI: 10.18653/v1/2020.emnlp-main.476. URL: https://www.aclweb.org/anthology/2020.emnlp-main.476.

[21] Myle Ott et al. “fairseq: A Fast, Extensible Toolkit for Sequence Modeling”. In: (Apr. 2019). URL: https://arxiv.org/abs/1904.01038.

[22] Myle Ott et al. *Scaling neural machine translation*. 2018. DOI: 10.18653/v1/w18-6301.

[23] Kishore Papineni et al. “BLEU: a method for automatic evaluation of machine translation”. In: *ACL* (2001). DOI: 10.3115/1073083.1073135.

[24] Matt Post. *A Call for Clarity in Reporting BLEU Scores*. 2018. DOI: 10.18653/v1/w18-6319.

[25] Rico Sennrich, Barry Haddow, and Alexandra Birch. “Neural Machine Translation of Rare Words with Subword Units”. In: (Aug. 2015). URL: http://arxiv.org/abs/1508.07909.

[26] Ralf Steinberger et al. “DGT-TM: A freely available translation memory in 22 languages”. In: *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012*. 2012.

[27] Ralf Steinberger et al. “The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages”. In: *Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006*. 2006.

[28] Jörg Tiedemann. “Parallel data, tools and interfaces in OPUS”. In: *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012*. 2012.

[29] University of Tartu. *UT Rocket*. 2018. DOI: 10.23673/PH6N-0144. URL: https://share.neic.no/#/marketplace-public-offering/c8107e145e0d41f7a016b72825072287/.

[30] Ashish Vaswani et al. “Attention Is All You Need”. In: (June 2017). URL: http://arxiv.org/abs/1706.03762.

[31] Raúl Vázquez et al. “Multilingual NMT with a language-independent attention bridge”. In: (Nov. 2018). DOI: 10.18653/v1/W19-4305. URL: http://arxiv.org/abs/1811.00498http://dx.doi.org/10.18653/v1/W19-4305.

[32] Biao Zhang et al. “Improving Massively Multilingual Neural Machine Translation and Zero-Shot Translation”. In: (Apr. 2020). URL: http://arxiv.org/abs/2004.11867.

[33] Tianyi Zhang et al. “BERTScore: Evaluating Text Generation with BERT”. In: (Apr. 2019). URL: http://arxiv.org/abs/1904.09675.

Appendix

I. Dataset Sizes

Table 12. Train set sizes (number of sentence pairs).

	Dataset							
Lang. pair	JRC-A	PC	OS	EP	DGT	N-C	M-UN	Total
et-lv	1500000	0	400000	633491	4792234	0	0	7325725
et-lt	900000	0	1100000	617918	4783718	0	0	7401636
et-fi	14174	0	8000000	617286	4000000	0	0	12631460
et-ru	0	0	6556926	0	0	0	0	6556926
et-en	497086	1000000	8000000	647060	2000000	0	0	12144146
et-de	1400000	0	6000000	577610	4000000	0	0	11977610
lv-lt	1400000	0	200000	617786	4700000	0	0	6917786
lv-fi	15297	0	461735	613144	4700000	0	0	5790176
lv-ru	0	0	428927	0	0	0	0	428927
lv-en	450000	1000000	500000	635182	2309736	0	0	4894918
lv-de	1000000	0	400000	572577	4600000	0	0	6572577
lt-fi	15087	0	1000000	609372	4500000	0	0	6124459
lt-ru	0	0	944897	0	0	0	0	944897
lt-en	500000	1000000	1400000	630043	2400000	0	0	5930043
lt-de	900000	0	900000	565167	4600000	0	0	6965167
fi-ru	0	0	6500000	0	0	0	0	6500000
fi-en	11662	2000000	3000000	1000000	2000000	0	0	8011662
fi-de	16148	0	3000000	1500000	2000000	0	0	6516148
ru-en	0	0	4000000	0	0	189549	4000000	8189549
ru-de	0	0	6000000	0	0	175631	162064	6337695
en-de	400000	0	4000000	1500000	2000000	207409	123721	8231130
Total	9019454	5000000	62792485	11336636	53385688	572589	4285785	146392637

Table 13. Development set sizes (number of sentence pairs).

	Dataset							
Lang. pair	JRC-A	PC	OS	EP	DGT	N-C	M-UN	Total
et-lv	50	0	50	50	50	0	0	200
et-lt	50	0	50	50	50	0	0	200
et-fi	50	0	50	50	50	0	0	200
et-ru	0	0	250	0	0	0	0	250
et-en	50	50	50	50	50	0	0	250
et-de	50	0	50	50	50	0	0	200
lv-lt	50	0	50	50	50	0	0	200
lv-fi	50	0	50	50	50	0	0	200
lv-ru	0	0	250	0	0	0	0	250
lv-en	50	50	50	50	50	0	0	250
lv-de	50	0	50	50	50	0	0	200
lt-fi	50	0	50	50	50	0	0	200
lt-ru	0	0	250	0	0	0	0	250
lt-en	50	50	50	50	50	0	0	250
lt-de	50	0	50	50	50	0	0	200
fi-ru	0	0	250	0	0	0	0	250
fi-en	50	50	50	50	50	0	0	250
fi-de	50	0	50	50	50	0	0	200
ru-en	0	0	100	0	0	100	50	250
ru-de	0	0	100	0	0	100	50	250
en-de	50	0	50	50	50	50	0	250
Total	750	200	1950	750	750	250	100	4750

Table 14. Test set sizes (number of sentence pairs).

	Dataset							
Lang. pair	JRC-A	PC	OS	EP	DGT	N-C	M-UN	Total
et-lv	100	0	100	100	100	0	0	400
et-lt	100	0	100	100	100	0	0	400
et-fi	100	0	100	100	100	0	0	400
et-ru	0	0	100	0	0	0	0	100
et-en	100	100	100	100	100	0	0	500
et-de	100	0	100	100	100	0	0	400
lv-lt	100	0	100	100	100	0	0	400
lv-fi	100	0	100	100	100	0	0	400
lv-ru	0	0	100	0	0	0	0	100
lv-en	100	100	100	100	100	0	0	500
lv-de	100	0	100	100	100	0	0	400
lt-fi	100	0	100	100	100	0	0	400
lt-ru	0	0	100	0	0	0	0	100
lt-en	100	100	100	100	100	0	0	500
lt-de	100	0	100	100	100	0	0	400
fi-ru	0	0	100	0	0	0	0	100
fi-en	100	100	100	100	100	0	0	500
fi-de	100	0	100	100	100	0	0	400
ru-en	0	0	100	0	0	100	100	300
ru-de	0	0	100	0	0	100	100	300
en-de	100	0	100	100	100	100	100	600
Total	1500	400	2100	1500	1500	300	300	7600

II. BLEU Scores

Table 15. Universal model – internal test set BLEU scores.

	Target						
Source	en	de	fi	lt	et	lv	ru
en	-	40.44	32.31	41.31	39.37	41.6	31.65
de	47.93	-	26.11	32.3	35.15	36.66	35.49
fi	40.24	28.03	-	26.22	29.66	32.65	12.47
lt	48.09	33.23	24.87	-	33.86	40.2	20.41
et	49.02	37.18	31.11	33.34	-	38.15	16.87
lv	48.67	35.38	30.35	38.91	35.02	-	22.98
ru	37.97	35.35	9.19	20.25	20.78	18.11	-

Table 16. Modular model (base) trained with proportional sampling – internal test set BLEU scores.

	Target						
Source	en	de	fi	lt	et	lv	ru
en	-	37.75	29.85	39.31	37.04	39.63	30.65
de	45.59	-	23.59	30.25	34.18	34.67	34.93
fi	37.97	26.33	-	24.28	28.18	29.95	13.67
lt	45.97	30.86	22.55	-	31.53	37.8	20.02
et	45.36	34.85	28.02	32.44	-	36.07	18.99
lv	44.74	32.78	27.56	36.54	32.68	-	23.33
ru	36.38	33.23	11.25	19.16	22.12	18.29	-

Table 17. Modular model (base) trained with round-robin oversampling – internal test set BLEU scores.

	Target						
Source	en	de	fi	lt	et	lv	ru
en	-	37.95	29.43	39.47	36.49	41.25	30.02
de	45.03	-	23.54	29.46	32.89	35.03	34.22
fi	37.67	26.69	-	23.89	27.61	30.1	13.96
lt	46.55	30.99	22.93	-	31.29	38.41	20.71
et	45.17	34.41	28.33	32.62	-	36.35	18.71
lv	45.61	33.19	28.26	36.07	33.23	-	27.96
ru	35.31	32.96	12.69	22.16	21.22	21.79	-

Table 18. UT model – internal test set BLEU scores.

	Target						
Source	en	de	fi	lt	et	lv	ru
en	-	35.45	26.35	33.91	32.71	36.44	29.63
de	40.27	-	21.36	25.96	26.37	28.87	33.67
fi	35.47	22.9	-	21.87	24.01	26.79	11.69
lt	42.16	26.47	20.18	-	25.87	33.05	18.52
et	41.9	29.96	24.79	26.39	-	32.38	16.78
lv	43.35	27.03	22.42	33.57	29.18	-	23.22
ru	36.28	34.15	5.92	16.78	19.55	18.23	-

Table 19. Big modular model's gain in internal test set BLEU scores over the universal model. The bottom rightmost cell shows the average gain in BLEU score over all the language pairs.

	Target							
Source	en	de	fi	lt	et	lv	ru	Average
en	-	2.99	3.43	2.62	1.99	2.95	1.63	2.60
de	3.24	-	2.43	0.44	2.01	0.44	3.39	1.99
fi	3.25	1.97	-	1.07	1.67	0.34	2.87	1.86
lt	4.14	1.46	2.05	-	1.08	1.45	1.69	1.98
et	2.37	1.35	1.55	1.90	-	1.59	3.44	2.03
lv	2.06	3.20	2.10	0.96	1.41	-	4.23	2.33
ru	3.76	4.53	3.70	2.28	2.95	3.59	-	3.47
Average	3.14	2.58	2.54	1.55	1.85	1.73	2.88	2.32

III. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Taido Purason**,

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,

   Modular Septilingual Neural Machine Translation,

   supervised by Andre Tättar and Elizaveta Korotkova.

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.

Taido Purason
07/05/2021