Multilingual Neural Machine Translation With the Right Amount of Sharing

Taido Purason
University of Tartu
Tartu, Estonia
taido.purason@ut.ee

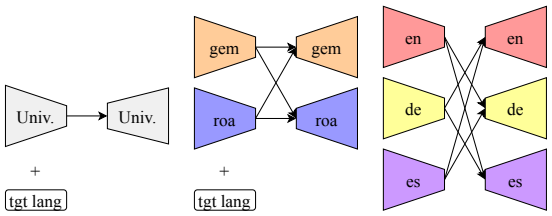
Andre Tättar
University of Tartu
Tartu, Estonia
andre.tattar@ut.ee

Abstract

Large multilingual Transformer-based machine translation models have had a pivotal role in making translation systems available for hundreds of languages with good zero-shot translation performance. One such example is the universal model with shared encoder-decoder architecture. Additionally, jointly trained language-specific encoder-decoder systems have been proposed for multilingual neural machine translation (NMT) models. This work investigates various knowledge-sharing approaches on the encoder side while keeping the decoder language- or language-group-specific. We propose a novel approach, where we use universal, language-group-specific and language-specific modules to solve the shortcomings of both the universal models and models with language-specific encoders-decoders. Experiments on a multilingual dataset set up to model real-world scenarios, including zero-shot and low-resource translation, show that our proposed models achieve higher translation quality compared to purely universal and language-specific approaches.

1 Introduction

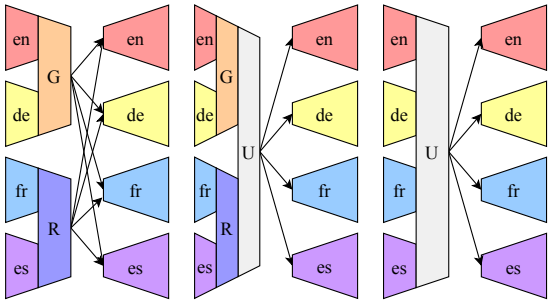
Multilingual neural machine translation has been a fundamental topic in recent years, especially for zero- and few-shot translation scenarios. Traditionally, universal NMT models (see Fig. 1a) have been used to produce zero-shot or low-resource translations (Johnson et al., 2016). However, previous research has established that universal NMT models with shared encoder-decoder architecture have some disadvantages: (1) high-resource language pairs tend to suffer loss in translation quality (Arivazhagan et al., 2019); (2) the vocabulary of the model increases greatly, especially for languages that do not share an alphabet such as English and Japanese; (3) the need to retrain from scratch when a new language does not share the model's vocabulary.



(a) Universal model **(b)** Model with language-group-specific modules **(c)** Model with language-specific modules

Figure 1: Different granularities of the modular architecture. *roa* – Romance; *gem* – Germanic; *tgt lang* – Target language token added to indicate the language of the output sentence.

Recently, there has been renewed interest in multilingual systems, which have jointly trained language-specific encoders-decoders (see Fig. 1c) which we call the modular architecture (Lyu et al., 2020). The goal of these models has been to achieve a better overall translation quality compared to universal or uni-directional NMT models. However, there is a disadvantage: lower zero-shot translation quality compared to universal models. To combat this problem, shared encoder/decoder layers (also called *interlingua* layers) have been proposed (Liao et al., 2021).



(a) Language group sharing **(b)** Language group and universal sharing (tiered) **(c)** Universal sharing (between all languages)

Figure 2: Different types of encoder layer sharing in the modular architecture. Note that the width of layers in the figure does not correspond to the actual width but rather reflects the sharing extent, i.e. all layers in the encoder have the same width dimension. U – universal, G – Germanic, R – Romance.

In this paper, we focus on improving the overall translation quality by using different knowledge- and layer-sharing methods. More specifically, we investigate the effect of sharing encoder layers to improve the generalizability and quality of NMT models. Secondly, we present novel language group based models that are inspired by the universal and modular systems. We propose (1) various degrees of granularity (or specificity) of modules (illustrated in Fig. 1); (2) layer sharing, including combining layers of various granularities into a tiered architecture (illustrated by Fig. 2). Our methods show better translation quality in all testing scenarions compared to the universal model without increasing training or inference time by having variable degrees of modularity or sharing in the encoder.

Our research looks beyond zero-shot and high-resource NMT performance – we set up our experiments to investigate model performance for many data scenarios like zero-shot and low- to high-resource settings. We use a combination of Europarl (Koehn, 2005), EMEA (Tiedemann, 2012), and JRC-Acquis (Steinberger et al., 2006) datasets for training and evaluation and six languages grouped into two language groups: Germanic (German, English, Danish) and Romance (French, Spanish, Portuguese). The results show that our approaches can provide an improvement to universal models in all data scenarios. Furthermore, our approaches improve the zero-shot and low-resource translation quality of the modular architecture without harming the high-resource language translation quality.

The main contributions of our paper are:

- We introduce a novel language-group-specific modular encoder and decoder architecture (Fig. 1b).
- Showing that different architectures of shared encoder layers (Fig. 2) improve the low-resource MT quality of the modular model while also improving the high-resource MT quality that suffers in the universal NMT setting.
- We empirically show what effect sharing encoder layers has and present a detailed analysis that supports layer sharing.

2 Related Works

Multilingual neural machine translation models follow the encoder-decoder architecture and approaches following this architecture can vary in the amount of parameter sharing (Dabre et al., 2020).

The most straightforward approach with no parameter sharing would be having a system of uni-directional models. While it is feasible with a small amount of high-resource languages, it becomes problematic in scenarios with low-resource languages or a large number of languages. Firstly, the number of uni-directional models in the system grows quadratically with the number of languages, harming maintainability. Secondly, there is no transfer learning between language pairs due to separate models, which means that low-resource languages generally have low translation quality. These issues are addressed by pivoting with some success, however, it does not come without trade-offs (Habash and Hu, 2009). The main problem with pivoting is that it is not possible to fully utilize all the training data since we only use training data that contains the pivot language. Furthermore, due to multiple models being potentially used for a translation, the translation is slower, and there is a chance of error propagation and loss of information.

The most widely used approach in multilingual NMT uses a fully shared (universal) model, which has a single encoder and decoder shared between all the languages and uses a token added to the input sentence to indicate the target language (Johnson et al., 2016). Arivazhagan et al. (2019) identified that the universal model suffers from the capacity bottleneck: with many languages in the model, the translation quality begins to deteriorate.

This especially harms the translation quality of high-resource language pairs. Zhang et al. (2020) further confirmed this and suggested deeper and language-aware models as an improvement. Still, the problem of low maintainability remains, since adding the languages to the model is not possible without retraining the whole model. Furthermore, adding languages with different scripts likely results in lower translation quality since the vocabulary can not be altered.

Escolano et al. (2019) suggested a proof-of-concept model with language-specific encoders and decoders that started bilingual and was incrementally trained to include other languages. Escolano et al. (2020) further improved on it and proposed a joint training procedure that produced a model that outperformed the universal model in translation quality. Furthermore, their proposed model is expandable by incrementally adding new languages without affecting the existing languages' translation quality. Lyu et al. (2020) investigated the performance of the modular model from the industry perspective. They found that the modular model often outperforms single direction models thanks to transfer learning while being a competitor to the universal model as well due to the additional capacity of language-specific modules.

Modular models can contain shared modules as well. Liao et al. (2021) set out to improve the zero-shot performance of modular models, which is often worse than the zero-shot performance of universal models. They achieve this by sharing upper layers of language-specific encoders between all languages. The current paper is an extension of that work. While Liao et al. (2021) used English-centric training data and denoising autoencoder task to achieve universal interlingua, in this paper we are not using an autoencoder task, since our data is not one language centric.

Introducing language-specific modules into a universal model can be a good way to increase the capacity of the model without significantly increasing training or inference time. An example of a system that utilizes this is described in Fan et al. (2020). They use language-specific and language group layers in the decoder of the model following the universal architecture model to provide more capacity. They also note that language-specific layers are more effective when applied to the decoder. Liao et al. (2021) also found that sharing in decoder is not beneficial when there are shared layers in the encoder. These are also the main motivations for focusing on sharing encoder layers in this paper.

3 Experiment setup

3.1 Data

Our aim was to create a dataset that resembles a real-world scenario where language pairs with varying amounts of data are encountered. The data is collected from Europarl (Koehn, 2005), EMEA (Tiedemann, 2012), and JRC-Acquis (Steinberger et al., 2006). The training dataset is created by sampling from the aforementioned datasets so that the training dataset is composed of 70% Europarl, 15% EMEA, and 15% JRC-Acquis. The test set is composed of completely multi-parallel sentences.

Language combination	Direction (lang. group)	
	intra	inter
high–high	1,000,000	1,000,000
high–mid	500,000	500,000
mid–mid	500,000	100,000
low–high	100,000	10,000
low–mid	100,000	0
low–low	0	0

Table 1: Dataset size rules per language type pair and language group. intra – translation within language group, inter – translating between language groups

The dataset is composed of English, German, Danish, French, Spanish, and Portuguese. For creating the dataset and defining models, these are divided into Germanic (English, German, Danish) and Romance (French, Spanish, Portugese) language groups. We define high-resource (English, German, French), medium-resource (Spanish), and low-resource (Danish, Portuguese) languages that produce high-resource (1,000,000 lines), higher medium resource (500,000 lines), lower medium resource (100,000 lines), low-resource (10,000 lines), and zero-shot (0 lines) language pairs when combined according to the rules in Table 1. With these rules, we also give low and medium resource language directions less training sentences if they consist of languages from different language groups compared to the pairs consisting of the same language group languages. The resulting dataset composition from these rules is visible in Table 2. The test set consists of 2000 multi-parallel sentences for each language pair from the same distribution as the training data. Since the training dataset is cre-

src	tgt						
	en	de	da	fr	es	pt	all
en	–	1,000,000	100,000	1,000,000	500,000	10,000	2,610,000
de	1,000,000	–	100,000	1,000,000	500,000	10,000	2,610,000
da	100,000	100,000	–	10,000	0	0	210,000
fr	1,000,000	1,000,000	10,000	–	500,000	100,000	2,610,000
es	500,000	500,000	0	500,000	–	100,000	1,600,000
pt	10,000	10,000	0	100,000	100,000	–	220,000
all	2,610,000	2,610,000	210,000	2,610,000	1,600,000	220,000	9,860,000

Table 2: Dataset sizes (number of sentence pairs) per language pair.

ated by randomly sampling data for each language pair, it is not completely multi-parallel, however, it probably contains many multi-parallel lines. The validation dataset is created for all non-zero-shot pairs with size per language pair defined by $n_{\text{test}}(\text{langpair}) = \max(n_{\text{train}}(\text{langpair}) \cdot 0.0006, 100)$.

The dataset size is quite small compared to data used for training state-of-the-art models mainly due to limited computational resources. However, we believe that it still allows us to draw conclusions that can be applied at larger scales.

3.2 Model architecture

Previous research has investigated sharing layers of the modular architecture (Liao et al., 2021). In this work, we mainly focus on layer sharing in the encoders. The layers are shared in 2 ways: (1) inside language groups (Fig. 2a), and (2) between all languages (universally, Fig. 2c). These two methods are also combined into a tiered architecture (Fig. 2b). We also experiment with different levels of granularity of modules and introduce language-group-specific modules referred to as *group modular* model (Fig. 1b).

As baselines, we use a modular architecture without layer sharing (Fig. 1c) and a universal architecture with one encoder and decoder shared between all languages (Fig. 1a).

All of the models in our experiments follow the transformer base architecture (Vaswani et al., 2017) (6 encoder layers, 6 decoder layers). In addition to dropout of 0.1, attention and activation dropout of 0.1 are used. The embeddings are shared within a language module (encoder-decoder) for language-specific modular models and within a language group module for group modular models. For the universal model, all embeddings are shared.

3.3 Segmentation model training

We use Byte Pair Encoding (BPE) (Sennrich et al., 2016) implemented in SentencePiece (Kudo and Richardson, 2018) as the segmentation algorithm. For the language-specific encoder-decoder approach, we train a BPE model with a vocabulary size of 16,000 for each of the languages. In the group-specific approach, we have a BPE model for each of the language groups with a vocabulary size of 32,000. For the universal model, we have a single unified BPE model with vocabulary size of 32,000. For training the BPE models, we use character coverage of 1.0 and training data consisting of the training set of the corresponding languages.

3.4 Model training

Fairseq (Ott et al., 2019) is used to implement training and models. We made the code for our custom implementations publicly available[1].

For the following experiments, we set the convergence criteria to be 5 epochs of no improvement in the validation set loss. To evaluate the experiments, we always use the best epoch according to the validation loss.

The learning rate is selected from {0.0002, 0.0004, 0.0008} by the highest BLEU score on the validation set after 20 training epochs. Gradient accumulation frequency is selected using BLEU score on the validation set after convergence from 8, 16, 32, 48. For all experiments in this paper, the total maximum batch size is 384,000 tokens (max tokens in a batch multiplied by the gradient accumulation frequency and the number of GPUs).

From the initial experiments, learning rate of 0.0004 and gradient accumulation frequency of 48 is selected. For all experiments, Adam optimizer (Kingma and Ba, 2015), inverse square root learning-rate scheduler with 4,000 warm-up steps, and label smoothing (Szegedy et al., 2016) of 0.1

[1] https://github.com/TartuNLP/fairseq/tree/modular-layer-sharing

Architecture	Language pair resource					
	zero-shot	low	medium-low	medium-high	high	all
Universal	33.62	38.12	39.64	43.64	42.32	39.87
Group modular (GM)						
EA3–6	**35.03**	**39.48**	40.89	44.66	43.31	**41.06**
EA5–6	34.52	39.23	40.78	44.59	43.19	40.88
No sharing	33.76	38.90	40.75	44.60	43.32	40.73
Language modular (LM)						
EA3–6	34.73	38.79	**40.91**	44.68	43.36	40.90
EG3–4 EA5–6	34.57	38.61	40.76	**44.91**	**43.59**	40.90
EG 3–6	34.37	38.56	40.56	44.90	43.42	40.78
EA5–6	33.81	38.28	40.32	44.75	43.38	40.54
EG5 EA6	33.51	38.07	40.33	44.72	43.41	40.46
EG5–6	33.59	37.85	40.32	44.69	43.44	40.43
No sharing	32.14	37.19	39.92	44.74	43.50	40.02

Table 3: Average test set BLEU scores per language pair resource. EG - encoder layer shared within language group, EA - encoder layer shared between all languages. Best score(s) per resource (column) in bold.

are used.

The training approach is similar to the proportional approach in Lyu et al. (2020). The batches are created according to the granularity of the modules, so that the correct module can be chosen for each batch. For the modular models with language-specific encoders-decoders, each batch contains only samples from one language pair. For the group-specific models, the batch contains data from one group pair. We determined by preliminary experiments that gradient accumulation is necessary for the modular models to learn, which we speculate is due to language-specific modules and the aforementioned batch creation strategy. Since the universal model does not have that constraint, a lower gradient accumulation frequency of 8 is used. For group-specific and universal models, target language tokens are added to the input sentence.

We used one NVIDIA A100 GPU for training the models. All models were trained with mixed precision.

3.5 Evaluation

BLEU (Papineni et al., 2001) score is used as the primary metric for translation quality. It is calculated using SacreBLEU[2] (Post, 2018). Beam search with beam size of 5 is used for decoding. Since there are 30 language pairs in total, we group the languages depending on the size of the language pair dataset and mostly look at average test set BLEU scores for analysis.

4 Results

4.1 Main results

As a baseline, we trained a universal and a modular model. We then trained modular models with 2 uppermost or 4 uppermost layers of the encoder shared universally, language-group-specifically or tiered (bottom half of the shared layers shared group-specifically, the rest universally). We also explore language-group-specific modules (group modular model). The main results are visible in Table 3 (evaluation results of individual directions are in Appendix B). Note that the ordering of rows in the table corresponds to the increasing order of total number of parameters which can be found in Appendix A.

4.1.1 No sharing

We can firstly observe that the modular model without any sharing (LM No sharing) performs worse on zero-shot and low-resource language pairs than the universal model (by 1.48 and 0.93 BLEU points, respectively). However, when looking at the medium-high and high resource directions, the modular model performs achieves a higher translation quality (by 1.10 and 1.18 BLEU points, respectively). The translation quality in the medium-low language pairs is similar between the universal and baseline modular model.

4.1.2 Sharing 2 layers

Compared to the baseline modular model (LM No sharing), the modular model with 2 shared encoder layers (LM EA5–6) performs better on

[2]signature: `refs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0`

zero-shot, low, and medium-low resource language pairs on average, with medium-high and high resource language translation quality only slightly decreasing. Overall, we can observe 0.52 BLEU point increase in translation quality of the shared layer model compared to the modular model.

We can also see that with sharing 2 upper layers in language groups (LM EG5–6) or tiered (LM EG5 EA6), the results are similar, but on average lower by 0.11 and 0.08 BLEU points, respectively. Sharing layers group-specifically gives a similar effect to sharing layers between all languages on average. With group-specific sharing, the lower resource languages have a slightly lower BLEU score, and the higher resource languages have a slightly higher BLEU score compared to the universal layer sharing. We can see the same trend with tiered sharing.

Comparing the language modular models with 2 shared layers to the universal model, the group sharing (LM EG5–6) and tiered (LM EG5 EA6) have slightly worse translation quality in zero- and low-resource language pairs on average, however they outperform the universal model in all of the other higher resource directions. The model with 2 universally shared layers outperforms the universal model in all resource levels. On average, the universally shared modular model (LM EA5–6) outperforms the universal model by 0.67 BLEU points.

4.1.3 Sharing 4 layers

We can see that sharing 4 layers provides better translation quality on average than sharing 2 layers. All of the models (LM EG3–6, LM EG3–4 EA5–6, LM EA3–6) outperform the universal model in all resource types. The universally shared model (LM EA3–6) performs the best out of the three on average in the zero, low, and medium-low resource directions, while the tiered model (LM EG3–4 EA5–6) has the best higher resource performance, even outperforming the baseline modular model, although only by a small margin. Overall, the two aforementioned models have the highest average BLEU score of the language modular models, outperforming the baseline modular model by 0.88 points and the universal model by 1.03 points. Both of them outperform the universal model in the zero-shot direction: the universally shared modular model (LM EA3–6) by 1.11 BLEU points and the tiered modular model (LM EG3–4 EA5–6) by 0.95 BLEU points.

4.1.4 Group modules

When looking at models with group-specific modules (group modular in Table 3), we can see that they outperform the universal model and the baseline language modular model (LM No sharing) on average. The improvement over the baseline modular model comes mostly from the increase in translation quality in low-resource directions and the improvement over the universal model from higher-resource directions, as we also observed in the previous results. We can also observe that the group modular models outperform the universal model at all resource levels.

The group modular model also benefits from having layers shared between all languages. The average BLEU score increases when shared layers are added to the group modular model, which can mainly be attributed to the increase in zero-shot and low resource translation quality.

The group modular model with 4 encoder layers (GM EA3–6) shared is the best performing model in zero-shot and low-resource directions, outperforming the universal model by 1.41 BLEU points in zero-shot and 1.36 BLEU points in low-resource directions on average. On average, it outperforms the baseline language modular model by 1.04 BLEU points and the baseline universal model by 1.19 BLEU points. Complete evaluation results are presented in Appendix B.

Although we used language group modules and language group sharing in our experiments, we failed to find any meaningful effect on the translation quality when translating between language groups versus translating between languages in the same group.

4.2 Sharing between all languages

The previous experiments have shown that group sharing and tiered architectures were only slightly different from sharing between all languages. Furthermore, the number of shared layers affects the result more than the type of sharing. Hence, we continue with experiments on sharing the language modular model layers between all languages to further study the effect of number of encoder layers shared on BLEU scores. The results can be seen in Table 4.

We can see that, on average, sharing more layers increases the BLEU score steadily until 5 upper encoder layers are shared. Compared to sharing 5 upper layers, sharing all 6 layers slightly de-

Enc. shared layer(s)	Language pair resource					
	zero-shot	low	medium-low	medium-high	high	all
No sharing	32.14	37.19	39.92	44.74	**43.50**	40.02
6	33.07	37.63	40.09	44.67	43.35	40.23
5–6	33.81	38.28	40.32	44.75	43.38	40.54
4–6	34.16	38.43	40.41	44.85	43.43	40.68
3–6	34.73	38.79	**40.91**	44.68	43.36	40.90
2–6	**34.97**	**39.03**	40.81	**44.94**	43.44	**41.03**
1–6	34.61	38.70	40.79	44.60	43.23	40.80

Table 4: Average test set BLEU scores for experiments with encoder layer sharing between all languages in the language modular model.

creases the BLEU scores in all language resource types. This could be attributed to: (1) 1 language-specific layer can better transform the language-specific embeddings to a joint representation than none or (2) more capacity with 5 layers shared and 1 language-specific compared to sharing all 6.

The modular model with encoder layers 2–6 shared provides a very close BLEU score to the best performing model from the previous set of experiments (GM EA3–6). It should be noted however that none of the shared layer models outperform the plain modular model in high resource languages on average, although the difference is quite small. Detailed evaluation results with all translation directions for this model are available in Appendix B.

4.3 Effect of joint embeddings

Since the universal model uses joint embeddings and vocabulary and the modular model uses language-specific embeddings, we investigate whether this could be the reason for the better performance of the latter. We train a modular model with shared embeddings, vocabulary, and encoder layers while still using language-specific decoders. The results in Table 5 show that on average the modular model with shared encoder layers still outperforms the universal model in all resource types even with shared vocabulary and embeddings. Although the selection of training data for the SentencePiece model did not take the language data imbalance into account, we can see that using a unified segmentation model and vocabulary does not significantly decrease the translation quality.

5 Discussion and future work

Multilingual NMT is a complex problem. On the one hand, we face the problem of poor low-resource MT performance of the fully modular model, and on the other hand, we have the capacity issues of the universal model. Our experiments show that we can achieve the best of both worlds with models that combine aspects of both universal and modular NMT architectures.

Although including shared layers in the modular model has kept the translation quality of higher resource language pairs the same or slightly decreased it, there has been a substantial improvement in the translation quality of low and zero resource language pairs compared to the plain modular model. Furthermore, compared to the universal model, these shared layer modular models substantially increase translation quality in all types of language resource directions.

Language-group-specific modules are worth considering as an architecture, as they provide better translation quality in all language resource types compared to the universal model while having fewer parameters in total than models with language-specific modules. Even with language group modules, the zero-shot and low-resource translation quality benefits from layers shared between all languages.

The layer sharing strategy ultimately depends on the available computational and data resources. Having language-specific modules could become memory inefficient in massively multilingual scenarios. Hence, having language group modules or layer sharing is a good compromise between capacity and model size. Approaching the problem from the perspective of the universal model, using some degree of modularization is a good way of increasing capacity without sacrificing zero-shot performance or training time.

Our work also leaves room for future research. While we focused on encoder layer sharing, decoder layer sharing is a direction that we want to investigate in future work comprehensively. Incrementally adding languages is also an important aspect of modular models and should be investigated. In our work, we had a relatively small

Architecture	Language pair resource					
	zero-shot	low	medium-low	medium-high	high	all
Universal	33.62	38.12	39.64	43.64	42.32	39.87
Language modular						
shared enc. + emb. + voc.	**34.65**	**39.01**	40.67	44.43	43.06	40.77
shared enc.	34.61	38.70	**40.79**	**44.60**	**43.23**	**40.80**

Table 5: Average test set BLEU scores for embedding sharing experiments. shared enc. – shared encoder; shared enc. + emb. + voc. – shared encoder, shared embeddings (incl. decoder embeddings) and joint vocabulary.

dataset compared to many state-of-the-art systems, so it would be beneficial to see how our approaches work in a scenario with significantly more data. As previously mentioned, using significantly more languages in the system could also set more constraints on our approaches and would be a promising direction for future works since it could highlight differences between our proposed methods better.

6 Conclusion

In this paper, we propose multiple ways of improving universal models and models with language-specific encoders-decoders by combining features of both. We experimented with language- and language-group-specific modules and sharing layers of the encoders between all languages, groups of languages, or combining them into a tiered architecture. We found that having some layers universally shared (between all languages) benefits the zero-shot and low-resource translation quality of the modular architectures while not hurting the translation quality of high-resource directions. The modular models with some universally shared layers outperform the universal models in all language-resource types (from zero to high). Our best model outperforms the baseline language modular model by 1.04 BLEU points and the universal model by 1.19 BLEU points on average.

References

Arivazhagan, Naveen, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges. 7.

Dabre, Raj, Chenhui Chu, and Anoop Kunchukuttan. 2020. A Survey of Multilingual Neural Machine Translation. *ACM Computing Surveys*, 53(5).

Escolano, Carlos, Marta R. Costa-jussà, and José A. R. Fonollosa. 2019. From Bilingual to Multilingual Neural Machine Translation by Incremental Training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 236–242, Stroudsburg, PA, USA. Association for Computational Linguistics.

Escolano, Carlos, Marta R. Costa-jussà, José A. R. Fonollosa, and Mikel Artetxe. 2020. Multilingual Machine Translation: Closing the Gap between Shared and Language-specific Encoder-Decoders. 4.

Fan, Angela, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond English-Centric Multilingual Machine Translation. 10.

Habash, Nizar and Jun Hu. 2009. Improving Arabic-Chinese Statistical Machine Translation using English as Pivot Language. In *EACL 2009 - 4th Workshop on Statistical Machine Translation, Proceedings of theWorkshop*.

Johnson, Melvin, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. 11.

Kingma, Diederik P. and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Koehn, Philipp. 2005. Europarl : A Parallel Corpus for Statistical Machine Translation. *MT Summit*, 11.

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

Liao, Junwei, Yu Shi, Ming Gong, Linjun Shou, Hong Qu, and Michael Zeng. 2021. Improving Zero-shot Neural Machine Translation on Language-specific Encoders- Decoders. In *2021 International Joint*

Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 7.

Lyu, Sungwon, Bokyung Son, Kichang Yang, and Jaekyoung Bae. 2020. Revisiting Modularized Multilingual NMT to Meet Industrial Demands. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5905–5918, Online, 11. Association for Computational Linguistics.

Ott, Myle, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A fast, extensible toolkit for sequence modeling. In *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Demonstrations Session*.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 311, Morristown, NJ, USA. Association for Computational Linguistics.

Post, Matt. 2018. A Call for Clarity in Reporting BLEU Scores. In *WMT 2018 - 3rd Conference on Machine Translation, Proceedings of the Conference*, volume 1.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, volume 3.

Steinberger, Ralf, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. 2006. 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*.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-December.

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

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 2017-December.

Zhang, Biao, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving Massively Multilingual Neural Machine Translation and Zero-Shot Translation.

A Number of parameters

The number of parameters of the models can be seen in Table 6.

Architecture	Total params.	Inference params.
Universal	60,526,080	60,526,080
Group modular		
EA3–6	108,442,624	60,526,080
EA5-6	114,747,392	60,526,080
No sharing	121,052,160	60,526,080
Language modular		
EA3-6	250,938,368	52,331,008
EA5-6 EG3-4	257,243,136	52,331,008
EG3–6	263,547,904	52,331,008
EA5-6	282,462,208	52,331,008
EA6 EG5	285,614,592	52,331,008
EG5-6	288,766,976	52,331,008
No sharing	313,986,048	52,331,008

Table 6: Number of parameters

B Detailed evaluation results

Tables 7, 8, 9, 10, and 11 provide detailed evaluation results for selected experiments.

src	tgt					
	en	de	da	fr	es	pt
en	–	38.84	40.39	48.60	51.07	45.32
de	46.41	–	32.44	38.60	39.08	34.41
da	45.60	30.57	–	36.77	37.32	32.77
fr	49.28	32.19	31.65	–	42.95	39.65
es	52.06	32.66	32.63	44.02	–	41.13
pt	49.17	31.37	31.74	43.25	44.09	–

Table 7: Universal model test set BLEU scores.

src	tgt					
	en	de	da	fr	es	pt
en	–	1.30	2.14	1.25	1.30	-0.30
de	1.44	–	0.98	1.31	1.15	-0.38
da	0.56	-0.32	–	-1.56	-1.60	-2.93
fr	1.07	0.73	1.03	–	1.04	0.16
es	1.61	0.98	1.17	0.50	–	0.12
pt	-1.49	-2.84	-2.55	-0.77	-0.60	–

Table 8: Improvement of the baseline language modular model over the universal model on test set in BLEU points.

src	tgt					
	en	de	da	fr	es	pt
en	–	0.76	1.78	1.44	0.55	1.29
de	1.00	–	1.52	1.12	1.13	1.37
da	0.98	0.91	–	1.41	0.87	1.28
fr	0.79	0.82	1.62	–	0.75	1.51
es	1.31	1.11	1.87	1.25	–	0.98
pt	1.38	1.14	1.65	1.34	0.95	–

Table 9: Improvement of the group modular model with layers 3–6 shared (group modular EA3–6) over the universal model on test set in BLEU points.

src	tgt					
	en	de	da	fr	es	pt
en	–	0.84	1.75	1.49	1.10	-0.62
de	1.40	–	1.30	1.19	1.43	-0.44
da	2.30	1.25	–	1.93	1.59	0.35
fr	0.94	0.88	2.10	–	1.26	0.18
es	1.70	1.06	1.79	1.26	–	0.22
pt	1.73	0.80	1.70	1.07	1.33	–

Table 10: Improvement of the modular model with layers 2–6 shared (EA2–6) over the universal model on test set in BLEU points.

Lang. pair	Universal	Group modular			Language modular						
		EA3–6	EA5–6	–	EA3–6	EG3–4 EA5–6	EG3–6	EA5–6	EG5 EA6	EG5–6	–
en–de	38.84	39.6	39.57	39.77	39.96	40.11	39.8	39.67	39.96	39.83	**40.14**
de–en	46.41	47.41	47.25	47.32	47.76	47.8	47.78	**47.88**	47.56	47.72	47.85
en–da	40.39	42.17	41.99	42.37	42.36	42.65	42.5	42.52	42.45	**42.68**	42.53
da–en	45.6	46.58	46.77	46.62	47.86	**47.91**	47.52	46.93	47	47.23	46.16
en–fr	48.6	50.04	50.04	49.9	49.78	**50.15**	49.78	49.77	50.08	49.84	49.85
fr–en	49.28	50.07	49.84	50.32	50.43	50.56	50.49	**50.57**	50.27	50.45	50.35
en–es	51.07	51.62	52.03	52.01	51.92	52.22	52.34	52.18	52.03	52.07	**52.37**
es–en	52.06	53.37	53.27	53.58	53.72	53.77	53.84	**53.89**	53.69	53.7	53.67
en–pt	45.32	**46.61**	46.49	46.12	45.11	44.73	44.58	45.04	45.07	44.54	45.02
pt–en	49.17	**50.55**	50.39	50.53	50.13	49.95	49.95	48.97	48.82	48.87	47.68
de–da	32.44	33.96	33.66	33.56	34.08	**34.11**	33.67	33.93	33.75	33.58	33.42
da–de	30.57	31.48	31.42	31.21	**31.89**	31.53	31.27	30.85	30.8	30.95	30.25
de–fr	38.6	39.72	39.7	39.7	39.56	39.92	39.72	39.77	39.72	**39.97**	39.91
fr–de	32.19	**33.01**	32.72	32.93	32.68	32.98	32.97	32.64	32.89	32.83	32.92
de–es	39.08	40.21	40.12	40.2	39.94	**40.44**	40.28	40.18	40.07	40.06	40.23
es–de	32.66	**33.77**	33.61	33.29	33.44	33.63	33.76	33.66	33.55	33.45	33.64
de–pt	34.41	**35.78**	35.72	35.14	34.27	34.35	34.28	34.59	34.33	34.18	34.03
pt–de	31.37	**32.51**	32.35	32.17	31.55	31.51	31.52	30.38	30.03	30.02	28.53
da–fr	36.77	38.18	37.91	37.94	37.99	38	**38.26**	37.03	36.78	36.82	35.21
fr–da	31.65	33.27	32.54	31.49	**33.67**	33.11	32.8	33.65	33.37	32.66	32.68
da–es	37.32	38.19	38.31	37.84	38.47	38.56	**38.59**	37.39	37.09	37.52	35.72
es–da	32.63	34.5	33.41	32.46	34.52	**34.81**	33.78	34.62	34.14	34.23	33.8
da–pt	32.77	**34.05**	33.78	33.5	33.19	32.57	32.72	31.79	31.66	31.74	29.84
pt–da	31.74	**33.39**	32.57	31.24	32.76	32.34	32.38	31.44	31.13	30.86	29.19
fr–es	42.95	43.7	43.78	43.78	43.86	**44.18**	44.09	43.73	43.83	43.86	43.99
es–fr	44.02	**45.27**	44.74	44.76	45.18	45.21	45.08	44.88	45.14	44.98	44.52
fr–pt	39.65	**41.16**	41.08	40.84	40.13	39.57	39.79	39.88	39.97	39.64	39.81
pt–fr	43.25	**44.59**	44.27	44.24	44.19	43.94	43.79	43.16	43.14	42.99	42.48
es–pt	41.13	42.11	42.15	**42.38**	41.65	41.39	41.36	41.19	41.42	41.04	41.25
pt–es	44.09	45.04	44.88	44.78	**45.09**	44.95	44.61	44.06	44.1	44.46	43.49

Table 11: Test set BLEU scores for the main experiments. The best result of each row is in bold.