Language-Family Adapters for Multilingual Neural Machine Translation

Anonymous ACL submission

Abstract

Massively multilingual models, pretrained on monolingual data, yield state-of-the-art results in a wide range of natural language processing tasks. In machine translation, multilingual pretrained models are often fine-tuned on parallel data from one or multiple language pairs. Multilingual fine-tuning improves performance on medium- and low-resource languages but requires modifying the entire model and can be prohibitively expensive. Training a new set of adapters on each language pair or training a single set of adapters on all language pairs (*language-pair* or *language-agnostic* adapters) while keeping the pretrained model's parameters frozen has been proposed as a parameter-efficient alternative. However, the former do not learn cross-lingual representations, while the latter share parameters for all languages and potentially have to deal with negative interference. In this paper, we propose training *language-family adapters* on top of a pretrained multilingual model to facilitate cross-lingual transfer. Our model consistently outperforms other adapter-based approaches. We also demonstrate that language-family adapters provide an effective method to translate to languages unseen during pretraining.[1]

1 Introduction

Recent work in multilingual natural language processing (NLP) has created models that reach competitive performance, while incorporating many languages into a single architecture (Devlin et al., 2019; Conneau et al., 2020). Because of its ability to share cross-lingual representations, which largely benefits lower-resource languages, multilingual neural machine translation (NMT) is an attractive research field (Firat et al., 2016; Zoph et al., 2016; Johnson et al., 2017; Ha et al., 2016; Zhang et al., 2020; Fan et al., 2020). Multilingual models are also appealing because they are more efficient in terms of the number of model parameters, enabling simple deployment (Arivazhagan et al., 2019; Aharoni et al., 2019). Massively multilingual pretrained models can be used for multilingual NMT, if they are fine-tuned in a *many-to-one* (to map any of the source languages into a target language, which is usually English) or *one-to-many* (to translate a single source language into multiple target languages) fashion (Aharoni et al., 2019; Tang et al., 2020). Recently, a large dataset was proposed, which enables training a *many-to-many* (multiple source to multiple target languages) NMT model (Fan et al., 2020).

Multilingual pretrained models generally permit improving translation on low-resource language pairs. Specializing the model to a specific language pair further boosts performance, but is computationally expensive. For example, mBART-50 (Tang et al., 2020), pretrained on monolingual data of 50 languages, still has to be fine-tuned for NMT, which poses a computational overhead.

To avoid fine-tuning large models, previous work has focused on efficiently building multilingual NMT models. Adapters (Rebuffi et al., 2017; Houlsby et al., 2019), which are lightweight feedforward layers added in each Transformer (Vaswani et al., 2017) layer, have been proposed as a parameter-efficient method for fine-tuning. In machine translation, training a different set of adapters on each language pair on top of a pretrained multilingual NMT model, without updating the parameters of the underlying model, has shown to improve results for high-resource languages (Bapna and Firat, 2019). Low-resource languages do not benefit from this approach though, as adapters are trained with limited data. In a similar vein, Cooper Stickland et al. (2021) fine-tune a pretrained model for multilingual NMT using a single set of adapters, trained on all languages. Their approach manages to narrow the gap but still does not perform on par with multilingual fine-tuning.

Many-to-one and one-to-many NMT force lan-

[1] Our source code is attached and will be released.

guages into a joint space (in the encoder or decoder side) and neglect diversity. One-to-many NMT faces the difficulty of learning a conditional language model and decoding into multiple languages (Arivazhagan et al., 2019; Tang et al., 2020). To better model the target languages, recent approaches propose exploiting both the unique and the shared features (Wang et al., 2018), reorganizing parameter-sharing (Sachan and Neubig, 2018), decoupling multilingual word encodings (Wang et al., 2019a), or accounting for linguistic similarities (Tan et al., 2019; Fan et al., 2020).

In this work, we propose using *language-family* adapters that enable efficient multilingual NMT. We train adapters for NMT on top of mBART-50 (Tang et al., 2020), a model pretrained on monolingual data on 50 different languages. The adapters are trained using bi-text from each language family, while the pretrained model is not updated. Families are formed based on linguistic knowledge bases. Our approach improves positive cross-lingual transfer, compared to language-pair adapters (Bapna and Firat, 2019), which ignore similarities between languages, and language-agnostic adapters (Cooper Stickland et al., 2021), which are trained on all languages and can suffer from negative interference (Wang et al., 2020).

Our main contributions are: **1)** A novel, effective approach for multilingual translation which trains adapters on top of a pretrained model for each language family. In the English-to-many setting which we examine, language-family adapters achieve a $+1$ BLEU improvement over language-pair adapters and $+2.7$ BLEU improvement over language-agnostic adapters when evaluated on 16 medium- and low-resource language pairs from OPUS-100. **2)** We propose inserting *embedding-layer adapters* into the Transformer to encode lexical information and conduct an ablation study to show that they contribute to better translation scores across all languages. **3)** We contrast grouping languages based on linguistic knowledge to grouping them based on clustering the representations of a multilingual pretrained language (PLM) model using a Gaussian Mixture Model (GMM) and provide insights. **4)** We analyze the effect of our approach when evaluating on languages that are new to mBART-50.

2 Background

Massively Multilingual Models. Multilingual masked language models have pushed the start-of-the-art on cross-lingual language understanding by training a single model for many languages (Devlin et al., 2019; Conneau and Lample, 2019; Conneau et al., 2020). Encoder-decoder Transformer (Vaswani et al., 2017) models that are pretrained using monolingual corpora from multiple languages, such as mBART (Liu et al., 2020), have also shown to outperform strong baselines in NMT. Recently, mBART-50 (Tang et al., 2020) was introduced, pretrained in 50 languages and multilingually fine-tuned for NMT. However, while multilingual models are known to outperform strong baselines and simplify model deployment, they are susceptible to negative interference/transfer (McCann et al., 2018; Arivazhagan et al., 2019; Wang et al., 2019b; Conneau et al., 2020) and catastrophic forgetting (Goodfellow et al., 2014) when the parameters of a multilingual model are shared across a large number of languages. Negative transfer affects the translation quality of high-resource (Conneau et al., 2020), but also low-resource languages (Wang et al., 2020). Our approach takes advantage of language families and provides the flexibility necessary to decode into multiple languages, improving results in both low- and mid-resource scenarios compared to related methods.

Adapters for NMT. Adapters are parameter-efficient modules that are typically added to a pretrained Transformer and are fine-tuned on a downstream task, while the pretrained model is frozen. Bapna and Firat (2019) add *language-pair* adapters to a pretrained multilingual NMT model (one set for *each* language pair), to recover performance for high-resource language pairs. Cooper Stickland et al. (2021) start from an unsupervised pretrained model and train *language-agnostic* adapters (one set for *all* language pairs) to improve performance in multilingual NMT. Philip et al. (2020) train *monolingual* adapters to permit zero-shot translation to directions that were not seen during training, while Üstün et al. (2021) propose *denoising adapters*, i.e., adapters trained using monolingual data, for unsupervised multilingual NMT.

We identify some challenges in relevant previous works (Bapna and Firat, 2019; Cooper Stickland et al., 2021). Scaling language-agnostic adapters to a large number of languages is problematic, as when they are updated with data from multiple

languages, negative transfer occurs. In contrast, language-pair adapters do not face this problem, but at the same time do not allow any sharing between language pairs. Language-family adapters intuitively strike a balance, providing a trade-off between the two approaches, and our experiments show that they lead to higher translation quality.

Language Families. Extensive work on cross-lingual transfer has demonstrated that jointly training a model using similar languages can improve low-resource results in several NLP tasks, such as part-of-speech or morphological tagging (Täckström et al., 2013; Cotterell and Heigold, 2017), entity linking (Tsai and Roth, 2016; Rijhwani et al., 2019), and machine translation (Zoph et al., 2016; Johnson et al., 2017; Neubig and Hu, 2018; Oncevay et al., 2020). Linguistic knowledge bases (Littell et al., 2017; Dryer and Haspelmath, 2013) study language variation and can provide insights to phenomena such as negative interference. Languages can be clustered together using linguistic information, forming language families. Tan et al. (2019) and Kong et al. (2021) leverage families for multilingual NMT, the former by training language-family NMT models from scratch, the latter by training a separate shallow decoder for each family. Instead, our approach keeps a pretrained model frozen and only trains language-family adapters, which is parameter-efficient.

3 Language-Family Adapters for NMT

Fine-tuning a pretrained model for multilingual NMT provides a competitive performance, yet is computationally expensive, as all layers of the model need to be updated. A parameter-efficient alternative suggests fine-tuning a pretrained multilingual model for NMT with data from all languages of interest using adapters (language-agnostic adapters), while keeping the pretrained model unchanged. However, as multiple languages share the same parameters in a single set of adapters, capacity issues arise. Languages are also grouped together, even though they might be different in terms of geographic location, script, syntax, typology, etc. As a result, linguistic diversity is not modeled adequately and translation quality degrades.

We address the limitations of previous methods by proposing language-family adapters for multilingual NMT. We exploit linguistic knowledge to enable cross-lingual transfer between related languages and avoid negative interference. Our approach is to train adapters using language pairs of a linguistic family on top of mBART-50, while keeping the pretrained model frozen.

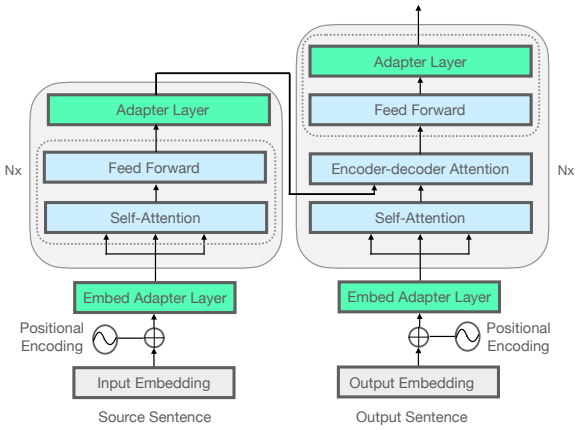

Figure 1: Proposed adapter architecture inside a Transformer model. Adapter layers, shown in green, are trained for NMT. Figure best viewed in color.

3.1 Adapter Architecture

Adapters are usually added to each Transformer layer. An adapter uses as input the output of the previous layer. Formally: Let z_i be the output of the i-th layer, of dimension h. We apply a layer-normalization (Ba et al., 2016), followed by a down-projection $D \in R^{h \times d}$, a ReLU activation and an up-projection $U \in R^{d \times h}$, where d is the bottleneck dimension and the only tunable hyper-parameter. The up-projection is combined with a residual connection (He et al., 2016) with z_i according to the following equation: $Adapter_i(z_i) = U\,\text{ReLU}(D\,\text{LN}(z_i)) + z_i$. This follows Bapna and Firat (2019). Adapters are randomly initialized.

3.2 Embedding-layer Adapter

Because we keep the token embeddings of mBART-50 frozen, adding flexibility to the model to encode lexical information of the languages of interest is crucial, especially for unseen languages (not part of its pretraining corpus). Lexical cross-lingual information could be encoded by learning new embeddings for the unseen languages (Artetxe et al., 2020) but this would be computationally expensive. We instead add an adapter after the *embedding* layer, in both the encoder and the decoder, which receives as input the lexical representation of each sequence and aims to capture token-level cross-lingual transformations.

Our approach draws inspiration from Pfeiffer

et al. (2020) and simplifies the invertible adapters structure. We use the large vocabulary of mBART-50 to extend the model to unseen languages. We note that adding scripts that do not exist in the vocabulary of mBART-50 is not possible with our approach. We point out that Chronopoulou et al. (2020); Pfeiffer et al. (2021); Vernikos and Popescu-Belis (2021) have recently proposed approaches to permit fine-tuning to unseen languages/scripts when using PLMs and we leave further exploration to future work.

3.3 Model Architecture

To train a model for multilingual NMT, we leverage mBART-50, a model pretrained on data from 50 languages using a denoising auto-encoding objective. We want to fine-tune this model on a variety of language pairs, by leveraging similarities between languages. Our model aims to provide a parameter-efficient alternative to traditional fine-tuning of the entire pretrained model.

To this end, we insert adapters after each *feed-forward* layer both in the encoder and in the decoder, following Bapna and Firat (2019) and we also add embedding-layer adapters. We freeze the pretrained encoder-decoder Transformer and fine-tune *only* the adapters on NMT. We leverage the knowledge of the pretrained model, but encode additional cross-lingual information on each language family using adapters. We fine-tune a new set of adapters multilingually on each *language family* and evaluate the performance on mid- and low-resource language pairs. Our proposed model architecture is depicted in Figure 1.

4 Experimental Setup

Data. We initially fine-tune the model on TED talks (Qi et al., 2018), using data from 17 languages paired to English. We then scale to a larger parallel dataset, using OPUS-100 (Zhang et al., 2020) for the same languages paired to English (with the only exception being English-Filipino, which does not appear in OPUS-100). For the TED experiments, we choose 17 languages, 9 of which were present during pretraining, while 8 are new to mBART-50. For OPUS-100, we use the same 16 languages (without Filipino), 9 of which were present during pretraining and 7 are new. In both sets of experiments, the languages belong to 3 language families, namely Balto-Slavic, Austronesian and Indo-Iranian. The parallel data details are reported in Table 1.

Language (code)	Family	Train Set	
		TED	OPUS-100
⋆Bulgarian (bg)	BS	174k	1M
Persian (fa)	I	151k	1M
⋆Serbian (sr)	BS	137k	1M
Croatian (hr)	BS	122k	1M
Ukrainian (uk)	BS	108k	1M
Indonesian (id)	A	87k	1M
⋆Slovak (sk)	BS	61k	1M
Macedonian (mk)	BS	25k	1M
Slovenian (sl)	BS	20k	1M
Hindi (hi)	I	19k	534k
Marathi (mr)	I	10k	27k
⋆Kurdish (ku)	I	10k	45k
⋆Bosnian (bs)	BS	6k	1M
⋆Malay (ms)	A	5k	1M
Bengali (bn)	I	5k	1M
⋆Belarusian (be)	BS	5k	67k
⋆Filipino (fil)	A	3k	-

Table 1: Languages used in the experiments. ⋆ indicates languages that are *unseen* from mBART-50, i.e., they do not belong to the pretraining corpus. *BS* stands for Balto-Slavic, *I* for Indoiranian, *A* for Austronesian.

Baselines. We compare the proposed language-family adapters with **1)** *language-agnostic* (LANG-AGNOSTIC) and **2)** *language-pair adapters* (LANG-PAIR). While the adapters are trained using parallel data, mBART-50 (pretrained on monolingual data) is not updated. Moreover, we compare our approach to multilingual fine-tuning (ML-FT), although it requires fine-tuning the entire model and is thus not directly comparable to the parameter-efficient approaches we study. We show this result in the Appendix.

The first baseline, LANG-AGNOSTIC adapters, fine-tunes a set of adapters using data from all languages (similar to Cooper Stickland et al., 2021). The second baseline, LANG-PAIR adapters, follows Bapna and Firat (2019): a new set of adapters is trained for each language pair, so no parameters are shared between different language pairs.

Training details. We start from the mBART-50 checkpoint[2]. We extend its embedding layer with randomly initialized vectors to account for the new languages. We reuse the 250k sentencepiece (Kudo and Richardson, 2018) model of mBART-50. We use the fairseq (Ott et al., 2019) library for all experiments. We select the final models using validation perplexity. If the model is trained on multiple languages (using mixed mini-batches), we use the

[2] https://github.com/pytorch/fairseq

Model	Balto-Slavic									Austro-nesian			Indo-Iranian					
	bg⋆	sr⋆	hr	uk	sk⋆	mk	sl	bs⋆	be⋆	id	ms⋆	fil⋆	fa	hi	mr	ku⋆	bn	AVG
OPUS-100																		
Lang-pair	**27.8**	17.5	23.7	**17.7**	25.0	**35.0**	**24.1**	**21.0**	10.1	28.0	24.5	-	**10.5**	15.6	17.0	14.1	13.0	20.3
Lang-agnostic	21.6	19.7	21.4	13.8	24.1	28.9	19.6	19.5	11.3	28.6	21.8	-	8.1	16.9	17.8	12.8	11.2	18.6
Lang-family	25.4	**20.9**	23.7	15.1	**27.7**	31.9	22.6	20.3	**15.2**	**31.3**	**25.4**	-	9.8	**18.7**	**25.0**	**15.3**	12.9	**21.3**
TED																		
Lang-pair	**35.7**	21.1	30.5	21.1	**24.2**	27.0	21.4	**28.6**	**12.5**	**35.4**	**23.4**	**12.2**	14.0	14.1	10.0	**4.9**	9.0	20.3
Lang-agnostic	31.7	24.0	29.7	21.9	20.6	26.5	20.2	27.8	7.7	33.8	22.1	11.6	17.0	15.5	7.0	3.3	6.0	19.2
Lang-family	33.8	**25.1**	30.5	22.2	22.8	**28.0**	21.5	27.8	9.5	34.7	22.0	11.5	**17.5**	**19.8**	10.3	4.1	**11.6**	**20.7**

Table 2: Test set BLEU scores when translating out of English ($en \rightarrow xx$) on OPUS-100 and TED. LANG-PAIR stands for language-pair, LANG-AGNOSTIC for language-agnostic, and LANG-FAMILY for language-family adapters. Languages denoted with ⋆ are new to mBART-50. Results in bold are significantly different ($p < 0.01$) to the best adapter baseline.

overall perplexity. We use beam search with size 5 for decoding and evaluate BLEU scores using SacreBLEU [3] for OPUS-100 and SacreBLEU without tokenization for TED (Post, 2018). We also compute COMET (Rei et al., 2020) scores using the *wmt-large-da-estimator-1719* pretrained model. Results are reported in the Appendix.

To train the models, we freeze mBART-50. We fine-tune the LANG-FAMILY, LANG-AGNOSTIC adapters in a multilingual, one-to-many setup, using English as the source language. LANG-PAIR adapters are fine-tuned for each language pair. All models have a bottleneck dimension of 512. We otherwise use the same hyperparameters as Tang et al. (2020) and report them in the Appendix.

5 Results and Discussion

5.1 Main results

Table 2 shows translation results for a subset of languages of OPUS-100 and TED in terms of BLEU using parallel data to fine-tune mBART-50 in the $en \rightarrow xx$ direction. We also report COMET scores in the Appendix.

Our approach (LANG-FAMILY) consistently improves results, with an average +1 BLEU performance boost across all languages compared to fine-tuning with LANG-PAIR adapters on OPUS-100 and +2.7 improvement compared to LANG-AGNOSTIC adapters. This shows that representations from similar languages are beneficial to a multilingual model in a low- and medium-resource setup. As the LANG-PAIR approach trains a new set of adapters on each language pair, it completely ignores similarities between languages and does not benefit from potential positive transfer, obtaining worse translations.

[3] Signature "BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.5.1"

	Parameters	Runtime	GPUs
LANG-AGNOSTIC	27M	35h	8
LANG-FAMILY	81M	78h	8
LANG-PAIR	432M	192h	8

Table 3: Parameters used by our approach and the baselines to train on OPUS-100.

We note that our approach is also a lot more efficient than the LANG-PAIR approach, as it requires only 20% of the parameters of LANG-PAIR adapters.

The most related baseline, LANG-AGNOSTIC, which groups all languages together, provides 18.6 BLEU score on average on OPUS-100. Training a set of adapters jointly on languages from different linguistic families hinders the decoding ability of the model, as negative interference affects its cross-lingual ability. Our approach instead fine-tunes a model on multiple languages which are similar to each other, yielding a +2.7 improvement compared to LANG-AGNOSTIC when trained and evaluated in OPUS-100. This showcases the utility of our model, which leverages both multilingual learning and language-family specific representations.

Our approach also outperforms both baselines on TED. It yields a +1.5 performance boost compared to LANG-AGNOSTIC and +0.4 BLEU compared to LANG-PAIR. This shows that selectively sharing languages together is more beneficial compared to grouping them all together or training each language pair separately. Although the improvement is marginal compared to LANG-PAIR, we note that our approach is also a lot more efficient in terms of parameters and easy to deploy.

	Balto-Slavic				Austro-nesian		Indo-Iranian			
	bg	hr	mk	be	id	ms	fa	ku	bn	AVG-16
Lang-Agnostic w/o emb adapter	21.3	21.5	28.3	10.5	28.7	21.5	7.6	12.4	10.9	18.1
Lang-Agnostic with emb adapter (baseline)	21.6	21.4	28.9	11.3	28.6	21.8	8.1	12.8	11.2	**18.6**
Lang-family w/o emb adapter	24.3	22.6	31.2	13.4	31.4	25.2	9.0	13.7	12.2	20.6
Lang-family with emb adapter (ours)	25.4	23.7	31.9	15.2	31.3	25.4	9.8	15.3	12.9	**21.3**

Table 4: Ablation of the proposed architecture for $en \rightarrow xx$ (Bleu scores) on OPUS-100. We present results only for a subset of languages per language family. Full results can be found in the Appendix.

5.2 Computational cost

We show in Table 3 the number of trainable parameters used for each approach. The mBART-50 model has 680M parameters. Our approach trains parameters that add up to just 11.9% of the full model. Lang-Agnostic is the most efficient approach, requiring just 8.4% trainable parameters. However, there is a cost in terms of performance compared to our model. Finally, training Lang-Pair adapters is relatively expensive (52.2% of the trainable parameters of mBART-50). All in all, our Lang-family approach provides a trade-off between performance and efficiency in terms of model parameters and is an effective method of adapting pretrained multilingual models to mid- and low-resource languages.

5.3 Embedding-layer adapter

In our proposed approach, the encoder and decoder embeddings are not updated during fine-tuning. We hypothesize that the model cannot learn useful lexical representations that would be fed to the encoder or decoder and finally result in better decoding to the target languages. To overcome this issue, we introduce an adapter after the *encoder embedding layer*, as well as after the *decoder embedding layer*. We do not tie these adapter layers, since they only add up a small number of parameters (1M each, i.e., 0.1% of mBART-50 parameters).

As we can see in Table 4, we get consistent gains across almost all language pairs by adding these adapters, for both our model and the Lang-Agnostic baseline. The former yields a $+0.5$ performance boost, while the latter a $+0.7$ improvement in terms of BLEU. While the gains are modest, they are consistent and come at a very small computational overhead. For some languages, such as Kurdish (which is an unseen language for mBART-50), results improve by $+1.6$ when using embedding-layer adapters. Since Kurdish is not part of mBART-50 pretraining corpus, encoding token-level representations is in this case more challenging and embedding-layer adapters allows the model to specialize in this language.

5.4 Automatic clustering of languages

Gaussian Mixture Model. For our main set of experiments, we used language families from WALS. However, not all languages in a single language family share the same linguistic properties (Ahmad et al., 2019). Moreover, it can often be the case that a language pair with data from a specific domain (e.g., Twitter) might not have a positive overlap, if trained together with a language pair from a distant domain (e.g., EuroParl). Training a model with data from heterogeneous domains leads to degraded performance, as was recently shown for the task of language modeling (Chronopoulou et al., 2021).

A data-driven approach may be able to induce the similarities between languages in an unsupervised way. To this end, we group languages together using Gaussian Mixture Model (GMM) clustering of text representations obtained from a PLM (Aharoni and Goldberg, 2020). We used released code by the authors of the paper.[4]

We use XLM-R (Conneau et al., 2020), a multilingual PLM and specifically the *xlmr-roberta-base* HuggingFace (Wolf et al., 2020) checkpoint. We encode 500 sequences of 512 tokens from each language (using the OPUS-100 dataset) to create sentence representations, by performing average pooling of the last hidden state. We then use PCA projection of dimension 100 and fit the sentence representations to a GMM with 3 components (3 Gaussian distributions, i.e., clusters). As this is a soft assignment, every language belongs with some probability to one or more clusters. We present the confusion matrix in the Appendix. For simplicity, we map each language to just one cluster based on where the majority of its samples are assigned to.

[4] https://github.com/roeeaharoni/unsupervised-domain-clusters

Language Groups				bg	id	fa	be	ku	AVG
ling. family	<be, bg, sr, hr, uk, sk, mk, sl, bs>	<id, ms>	<ku, fa, hi, mr, bn>	25.4	31.3	9.8	15.2	15.3	**21.3**
GMM	<bg, sr, hr, uk, sk, mk, sl, bs>	<**ku**, id, ms>	<**be**, fa, hi, mr, bn>	23.9	29.7	9.2	14.9	14.3	19.4
random	<bg, hr, mk, bs, be, ms, hi, mr, ku>	<sl, id>	<sr, uk, sk, fa, bn>	22.9	27.8	7.0	12.1	15.0	18.4

Table 5: Evaluation of different methods to form language families for $en \rightarrow xx$ (BLEU) on OPUS-100. We present results only for a subset of languages. Full results are shown in the Appendix.

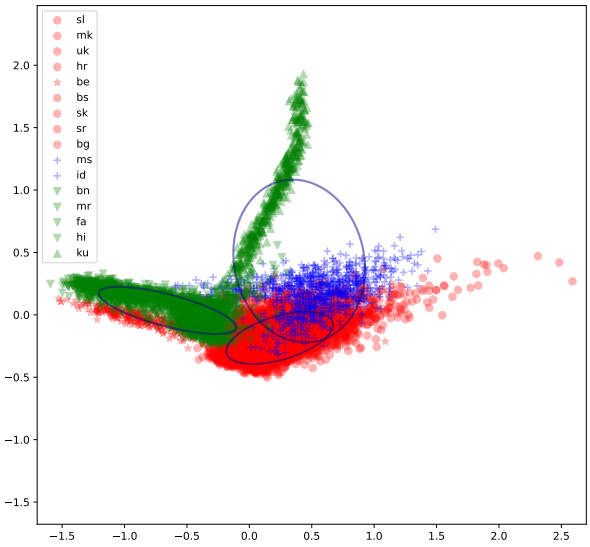

Figure 2: 2D projection of the GMM clustering. We use *red* (and the circle or star sign) for languages that belong to the Balto-Slavic family (according to linguistic knowledge), *blue* (plus sign) for Austronesian, *green* (triangle or reversed triangle sign) for Indo-Iranian languages. Notice that the languages that are primarily "mis-allocated" (i.e., belong to a linguistic family different to the cluster they are assigned to) are Belarusian (star sign) and Kurdish (triangle sign).

We confirm that XLM-R has a strong cross-lingual ability, as it was able to separate the languages to clusters, which correspond almost exactly to the families formed by linguistic knowledge.

In Figure 2, we show a 2-dimensional projection of the GMM clustering and in Table 5 (second row), the language groups that were formed by the same clustering. We notice that the GMM clusters are for the most part corresponding to the groups formed by linguistic knowledge. In our experiments, only Belarusian and Kurdish were allocated to groups that contained languages that are largely different. We believe that this might be caused by some domain mismatch, for example between the parallel dataset of Belarusian and those of the other Balto-Slavic languages. We have randomly sampled a subset of data for all languages, therefore we believe that the clustering obtained is robust to data variations.

Results. We see in Table 5 that training adapters using language groups computed by GMM clustering yields worse translation scores in terms of BLEU compared to language groups based on linguistic similarities. We conclude that perhaps it could be helpful to automatically cluster languages together, in the absence of linguistic knowledge bases. However, when such resources are available, leveraging them provides a more accurate translation model.

Moreover, randomly clustering languages together is ineffective, as expected. We believe that this shows that taking into account linguistic similarities or similar cross-lingual representations is beneficial when training a multilingual model for NMT. By ignoring relations between languages, negative interference can occur and hinder the overall translation quality of the model.

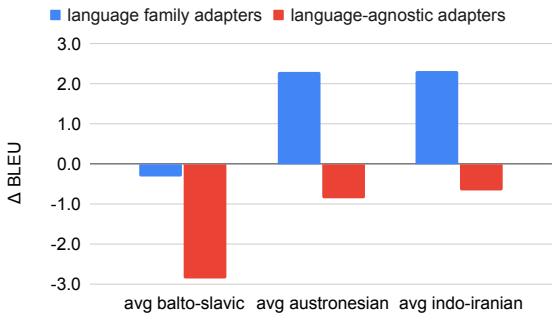

Figure 3: Grouping based on language family using OPUS-100. Difference (in terms of BLEU) compared to the performance of the LANG-PAIR model is shown.

6 Analysis

6.1 Performance according to language family

To evaluate the contribution of grouping languages based on linguistic information, we compute the difference of LANG-FAMILY adapters compared to the LANG-AGNOSTIC baseline *per language family* in terms of BLEU score. We show the results in Figure 3. The LANG-PAIR baseline is displayed as the x-axis in the same Figure.

Compared to the LANG-AGNOSTIC baseline, LANG-FAMILY adapters perform better in all lan-

guage families. On Balto-Slavic, our approach performs on par with the LANG-PAIR baseline (<0.5 BLEU difference). On both Austronesian and Indo-Iranian, our approach largely outperforms (more than +2 BLEU) both the LANG-PAIR and the LANG-AGNOSTIC baselines. This is arguably the case because LANG-AGNOSTIC adapters, trained using parallel data from all languages, group dissimilar languages together and do not take into account language variation. Training adapters using languages with common linguistic properties results in consistently improved translations.

We also observe that LANG-AGNOSTIC adapters perform worse than LANG-PAIR adapters when evaluated on each language family. This intuitively makes sense, as multilingual approaches only surpass the performance of bilingual ones in very low-resource scenarios. For most of the language pairs examined in our setup, we have 1M parallel sentences available, therefore a bilingual model yields a higher translation accuracy. Of course, training such a model is a lot more computationally expensive than both our approach and the LANG-AGNOSTIC baseline and the number of parameters grows linearly with the number of language pairs (see Table 3).

6.2 Performance on seen *vs* unseen languages

We also evaluate the performance of language-family adapters on languages that are not included in the mBART-50 pretraining data (*unseen*), compared to results on languages that belong to its pretraining corpus (*seen*). We present the results in Figure 4. We observe that LANG-FAMILY adapters boost the translation quality compared to the LANG-PAIR adapter baseline (depicted as the x-axis) on *unseen* languages. As the pretrained model has no knowledge of these languages, LANG-FAMILY adapters provide useful cross-lingual signal. LANG-FAMILY adapters that are fine-tuned on *seen* languages also yield an improvement compared to baselines.

LANG-AGNOSTIC adapters perform significantly worse than both our approach and the LANG-PAIR baseline. This might be the case because of negative transfer between unrelated languages, that are clustered and trained together using the LANG-AGNOSTIC model. This issue is prevalent for both the seen and the unseen languages.

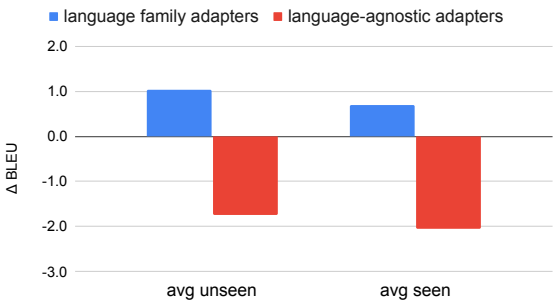

Figure 4: Grouping based on "seen" (existing in the pretraining corpus), or "unseen" language using OPUS-100. Difference (in terms of BLEU) compared to the performance of the LANG-PAIR model is shown.

7 Conclusion

We have presented a novel approach for fine-tuning a pretrained multilingual model for NMT using language-family adapters. Our approach can be used for multilingual NMT, combining the modularity of adapters with effective cross-lingual transfer between related languages. We have shown that language-family adapters perform better than both language-agnostic and language-pair adapters, while being computationally efficient. We also contrast our method of grouping languages together based on language families to an automatic way of clustering data from different languages, using the representations of a pretrained multilingual model and discuss the potential utility of this method. Finally, for languages new to mBART-50, we have shown that our approach provides an effective way of leveraging shared cross-lingual information between similar languages, considerably improving translations versus the baselines.

In the future, a more elaborate approach to encode lexical-level representations could further boost the performance of language-family adapters. We also hypothesize that the effectiveness of our model could be leveraged for other cross-lingual tasks, such as natural language inference, document classification and question-answering.

8 Limitations and Risks

Our work uses a large multilingual pretrained model. As unsupervised models are trained on large chunks of monolingual data from a lot of languages in the Internet, they encode biases that could harm marginalized populations (Bender et al., 2021). The NMT model we propose could be used to translate toxic text. However, we believe that adapters could potentially be used to specialize a multilingual pretrained model to not generate translations that contain hateful or toxic language.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Wasi Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard Hovy, Kai-Wei Chang, and Nanyun Peng. 2019. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2440–2452, Minneapolis, Minnesota. Association for Computational Linguistics.

Naveen Arivazhagan, 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.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Alexandra Chronopoulou, Matthew E. Peters, and Jesse Dodge. 2021. Efficient hierarchical domain adaptation for pretrained language models.

Alexandra Chronopoulou, Dario Stojanovski, and Alexander Fraser. 2020. Reusing a Pretrained Language Model on Languages with Limited Corpora for Unsupervised NMT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2703–2711, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*.

Asa Cooper Stickland, Xian Li, and Marjan Ghazvininejad. 2021. Recipes for adapting pre-trained monolingual and multilingual models to machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3440–3453, Online. Association for Computational Linguistics.

Ryan Cotterell and Georg Heigold. 2017. Cross-lingual character-level neural morphological tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 748–759, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology.

Angela Fan, 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. *CoRR*, abs/2010.11125.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2014. An empirical investigation of catastrophic forgeting in gradient based neural networks. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Thanh-Le Ha, Jan Niehues, and Alexander H. Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *CoRR*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 2790–2799.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Xiang Kong, Adithya Renduchintala, James Cross, Yuqing Tang, Jiatao Gu, and Xian Li. 2021. Multilingual neural machine translation with deep encoder and multiple shallow decoders. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1613–1624, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *CoRR*.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

Arturo Oncevay, Barry Haddow, and Alexandra Birch. 2020. Bridging linguistic typology and multilingual machine translation with multi-view language representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2391–2406, Online. Association for Computational Linguistics.

Myle Ott, 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 *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4465–4470, Online. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525.

Shruti Rijhwani, Jiateng Xie, Graham Neubig, and Jaime Carbonell. 2019. Zero-shot neural transfer for cross-lingual entity linking. In *The AAAI Conference on Artificial Intelligence*.

Devendra Sachan and Graham Neubig. 2018. Parameter sharing methods for multilingual self-attentional translation models. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 261–271, Brussels, Belgium. Association for Computational Linguistics.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 963–973, Hong Kong, China. Association for Computational Linguistics.

Y. Tang, C. Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *ArXiv*, abs/2008.00401.

Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual wikification using multilingual embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 589–598, San Diego, California. Association for Computational Linguistics.

Ahmet Üstün, Alexandre Berard, Laurent Besacier, and Matthias Gallé. 2021. Multilingual unsupervised neural machine translation with denoising adapters. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 6650–6662.

Ashish Vaswani, 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*.

Giorgos Vernikos and Andrei Popescu-Belis. 2021. Subword mapping and anchoring across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2633–2647, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Philip Arthur, and Graham Neubig. 2019a. Multilingual neural machine translation with soft decoupled encoding. In *International Conference on Learning Representations*.

Yining Wang, Jiajun Zhang, Feifei Zhai, Jingfang Xu, and Chengqing Zong. 2018. Three strategies to improve one-to-many multilingual translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2955–2960, Brussels, Belgium. Association for Computational Linguistics.

Zirui Wang, Zihang Dai, Barnabas Poczos, and Jaime Carbonell. 2019b. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

A Appendix

A.1 Dataset statistics

First, we show the script and language family (according to linguistic information) of each language used in our set of experiments in Table 6. We also present in detail the statistics of all parallel data used in our set of experiments in Table 8. We note that the number of train, validation and test set presented refers to sentences.

The TED dataset we used can be downloaded using the command `wget http://phontron.com/data/ted_talks.tar.gz` while the OPUS-100 can be downloaded from the following link. We include the urls in the main repository of the submitted code, in the `download_data.sh` file.

Language (code)	Family	Script
⋆Bulgarian (bg)	Balto-Slavic	Cyrillic
Persian (fa)	Indo-Iranian	Arabic
⋆Serbian (sr)	Balto-Slavic	Cyrillic
Croatian (hr)	Balto-Slavic	Latin
Ukrainian (uk)	Balto-Slavic	Cyrillic
Indonesian (id)	Austronesian	Latin
⋆Slovak (sk)	Balto-Slavic	Latin
Macedonian (mk)	Balto-Slavic	Cyrillic
Slovenian (sl)	Balto-Slavic	Latin
Hindi (hi)	Indo-Iranian	Devanagari
Marathi (mr)	Indo-Iranian	Devanagari
⋆Kurdish (ku)	Indo-Iranian	Arabic
⋆Bosnian (bs)	Balto-Slavic	Cyrillic
⋆Malay (ms)	Austronesian	Latin
Bengali (bn)	Indo-Iranian	Bengali
⋆Belarusian (be)	Balto-Slavic	Cyrillic
⋆Filipino (fil)	Austronesian	Latin

Table 6: Languages that are used in the experiments. ⋆ indicates languages that are *unseen* from mBART-50, i.e., they do not belong to the pretraining corpus. Filipino is only used in the TED experiments.

Adapter size	Dropout	Lang-Family	Lang-Agnostic
128	0.1	16.8	10.1
128	0.3	16.4	9.5
256	0.1	19.0	14.9
256	0.3	18.6	14.0
512	0.1	**20.7**	**19.2**
512	0.3	19.9	18.5

Table 7: Hyperparameter tuning for dropout, adapter bottleneck size on TED. Average performance (on all language pairs using TED) per model. We chose the best-performing combination of dropout and bottleneck size for our experiments.

A.2 Training details

We train each model for 130k updates with a batch size of 900 tokens per GPU for OPUS-100 and 1024 tokens per GPU for TED. We use 8 NVIDIA-V100 GPUs for OPUS-100 and 2 GPUs for TED (much smaller dataset). We evaluate models after 5k training steps. We use early stopping with a patience of 5. To balance high and low-resource language pairs, we use temperature-based sampling (Arivazhagan et al., 2019) with $T = 1.5$. We include the command we used to train the language-family adapter in OPUS-100 in the `train_lang_family_adapter.sh` script in the main repository of the uploaded code.

A.3 Evaluation of main results using 2 metrics

We evaluate the translations of our model (LANG-FAMILY adapters) and all the baselines trained on OPUS-100 using COMET (Rei et al., 2020). COMET leverages progress in cross-lingual language modeling, creating a multilingual machine translation evaluation model that takes into account both the source input and a reference translation in the target language. We rely on `wmt-large-da-estimator-1719`. COMET scores are not bounded between 0 and 1; higher scores signify better translations. Our results are summarized in Table 10. We see that COMET correlates with BLEU in our experiments.

A.4 Hyperparameters

We tune the dropout and the adapter bottleneck size on TED. We use values 0.1, 0.3 for the dropout and 128, 256, 512 for the bottleneck size. We list the hyperparameters we used to train both our proposed model and the baselines in Table 9.

A.5 Embedding-layer results

We report in Table 11 the results of the ablation study concerning the use of *embedding-layer* adapters on all languages.

A.6 Position of adapter in the encoder layer

We had initially (in the first submission of our paper) inserted the adapter layer in the encoder side of the Transofmer *before* the feed-forward layer. However, more careful experimentation showed that the best model architecture for our setup is adding the adapter *after* the feed-forward layer, following Bapna and Firat (2019). We show results of both model architectures in Table 13. We note that

Language	Source	Train	Valid	Test	Source	Train	Valid	Test
Bulgarian (bg)	TED	174k	4082	5060	OPUS-100	1M	2k	2k
Persian (fa)	TED	151k	3930	4490	OPUS-100	1M	2k	2k
Serbian (sr)	TED	137k	3798	4634	OPUS-100	1M	2k	2k
Croatian (hr)	TED	122k	3333	4881	OPUS-100	1M	2k	2k
Ukrainian (uk)	TED	108k	3060	3751	OPUS-100	1M	2k	2k
Indonesian (id)	TED	87k	2677	3179	OPUS-100	1M	2k	2k
Slovak (sk)	TED	61k	2271	2445	OPUS-100	1M	2k	2k
Macedonian (mk)	TED	25k	640	438	OPUS-100	1M	2k	2k
Slovenian (sl)	TED	20k	1068	1251	OPUS-100	1M	2k	2k
Hindi (hi)	TED	19k	854	1243	OPUS-100	534k	2k	2k
Marathi (mr)	TED	10k	767	1090	OPUS-100	27k	2k	2k
Kurdish (ku)	TED	10k	265	766	OPUS-100	45k	2k	2k
Bosnian (bs)	TED	6k	474	463	OPUS-100	1M	2k	2k
Malay (ms)	TED	5k	539	260	OPUS-100	1M	2k	2k
Bengali (bn)	TED	5k	896	216	OPUS-100	1M	2k	2k
Belarusian (be)	TED	5k	248	664	OPUS-100	67k	2k	2k
Filipino (fil)	TED2020	3k	338	338	OPUS-100	-	-	-

Table 8: Dataset details for TED (Qi et al., 2018; Reimers and Gurevych, 2020) and OPUS-100 (Zhang et al., 2020).

Hyperparameter	Value
Checkpoint	mbart50.pretrained
Architecture	mbart_large
Optimizer	Adam
β_1, β_2	0.9, 0.98
Weight decay	0.0
Label smoothing	0.2
Dropout	0.1
Attention dropout	0.1
Batch size	1024 tokens
Update frequency	2
Warmup updates	4k
Total number of updates	130k
Max learning rate	1e-04
Temperature sampling	5
Adapter dim.	512

Table 9: Fairseq hyperparameters used for our set of experiments.

both model architectures include embedding-layer adapters, and 1 adapter/transformer layer in both the encoder and the decoder.

A.7 Results using GMM, random clustering and language families

Full results of Table 5 can be seen in Table 12.

Lang	LANG-FAMILY BLEU	LANG-FAMILY COMET	LANG-PAIR BLEU	LANG-PAIR COMET	ML-FT BLEU	ML-FT COMET
bg	25.4	67.2	27.8	72.1	28.0	76.5
sr	20.9	44.3	17.5	38.2	21.1	48.4
hr	23.7	55.0	23.7	53.1	24.5	55.1
uk	15.1	-17.0	17.7	14.4	17.1	35.9
sk	27.7	54.3	25.0	50.1	30.5	64.9
mk	31.9	62.9	35.0	64.1	35.6	62.1
sl	22.6	48.9	24.1	65.8	24.5	64.3
bs	20.3	44.1	21.0	37.1	22.1	50.8
be	15.2	-10.2	10.1	-21.6	17.9	36.6
id	31.3	60.1	28.0	64.0	31.5	60.1
ms	25.4	53.5	24.5	66.1	25.5	68.0
fa	9.8	-23.5	10.5	-22.1	9.5	-15.0
hi	18.7	39.1	15.6	-19.1	18.4	36.4
mr	25.0	67.0	17.0	9.0	24.7	58.1
ku	15.3	-18.5	14.1	-12.9	15.6	-9.1
bn	12.9	-16.0	13.0	-24.1	14.1	-8.5
avg	21.3	32.0	20.3	27.1	22.5	42.8

Table 10: Test set BLEU and COMET scores when translating out of English using OPUS-100. Languages are presented by decreasing amount of parallel data per language family. LANG-PAIR stands for language-pair adapters, LANG-AGNOSTIC for language-agnostic, while LANG-FAMILY for language-family adapters. ML-FT stands for multilingual fine-tuning of the entire mBART-50 model.

	bg*	sr*	hr	uk	sk*	mk	sl	bs*	be*	id	ms*	fa	hi	mr	ku*	bn	AVG
Lang-agnostic w/o emb	21.3	19.0	21.5	13.9	23.6	28.3	19.1	18.9	10.5	28.7	21.5	7.6	16.1	16.9	12.4	10.9	18.1
Lang-agnostic with emb	21.6	19.7	21.4	13.8	24.1	28.9	19.6	19.5	11.3	28.6	21.8	8.1	16.9	17.8	12.8	11.2	18.6
Lang-family w/o emb	24.3	20.4	22.6	14.8	26.3	31.2	21.9	20.6	13.4	31.4	25.2	9.0	18.3	23.7	13.7	12.2	20.6
Lang-family with emb	25.4	20.9	23.7	15.1	27.7	31.9	22.6	20.3	15.2	31.3	25.4	9.8	18.7	25.0	15.3	12.9	**21.3**

Table 11: Full results of the ablation of the proposed architecture for $en \rightarrow xx$ (BLEU scores) on OPUS-100. Bold results indicate best performance on average.

	bg	sr	hr	uk	sk	mk	sl	bs	be	id	ms	fil	fa	hi	mr	ku	bn	AVG
GMM	23.9	17.7	24.4	11.0	19.3	22.9	19.0	23.6	14.9	29.7	23.4	-	9.2	18.8	25.5	14.3	13.2	19.4
random	22.9	18.8	23.5	10.0	22.5	31.9	21.1	20.1	12.1	25.8	24.9	-	5.0	18.6	22.9	15.0	8.1	18.4

Table 12: Evaluation of different methods to form language families for $en \rightarrow xx$ (BLEU) on OPUS-100.

	bg	sr	hr	uk	sk	mk	sl	bs	be	id	ms	fil	fa	hi	mr	ku	bn	AVG
Lang. family before ff	32.7	24.6	30.2	21.7	21.5	26.8	20.4	27.2	9.2	33.8	22.8	11.7	17.1	19.1	9.5	4.2	9.8	20.1
Lang. family after ff	33.8	25.1	30.5	22.2	22.8	28.0	21.5	27.8	9.5	34.7	22.0	11.5	17.5	19.8	10.3	4.1	11.6	**20.7**
Lang. agnostic before ff	30.7	23.4	28.8	21.3	19.7	25.3	19.7	26.5	8.1	32.9	22.2	11.5	16.5	14.8	6.5	3.3	6.0	18.7
Lang. agnostic after ff	31.7	24.0	29.7	21.9	20.6	26.5	20.2	27.8	7.7	33.8	22.1	11.6	17.0	15.5	7.0	3.3	6.0	**19.2**

Table 13: BLEU scores on TED of our proposed approach and the LANG-AGNOSTIC baseline, using two different model architectures. *Before ff* refers to adding an adapter in the encoder before the feed-forward layer. Bold results indicate best performance on average.