Language-Family Adapters for Multilingual Neural Machine Translation

Anonymous ACL submission

Abstract

Massively multilingual pretrained models yield state-of-the-art results in a wide range of cross-lingual natural language processing tasks. For machine translation, the de facto way to leverage knowledge of pretrained models is fine-tuning 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 either language-pair specific or language-agnostic adapters while keeping most of the pretrained model's parameters frozen has been proposed as a lightweight alternative. However, the former do not learn useful cross-lingual representations for multiple language pairs, 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. Using language families, our model consistently outperforms other adapter-based approaches and is on par with multilingual fine-tuning, while being more efficient. We also demonstrate that language-family adapters provide an effective method to translate to languages unseen during pretraining and substantially outperform the baselines.[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 NMT is an attractive research field (Firat et al., 2016; Zoph et al., 2016; Johnson et al., 2017; Ha et al., 2016). Multilingual models are also appealing because they are more efficient in terms of the number of model parameters, enabling simple deployment (Aharoni et al., 2019). Massively multilingual pretrained models can be used for multilingual neural machine translation (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).

We identify some major challenges for massively multilingual NMT models. Multilingually fine-tuning pretrained models to create NMT systems has recently been suggested (Tang et al., 2020), yet it requires vast computational resources. Therefore, previous work has focused on efficiently training multilingual NMT models. To fine-tune a pretrained multilingual model for NMT in an efficient way, adapters (Rebuffi et al., 2017; Houlsby et al., 2019) have been proposed (Bapna and Firat, 2019). Fine-tuning a different set of adapters on each language pair, without updating the parameters of the pretrained model, has been shown to improve results for high-resource languages. Low-resource languages do not benefit from this approach though, as adapters are trained with limited data. In a similar vein, 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.

Another issue is that while many-to-one multilingual NMT obtains high-quality translations, one-to-many generally yields smaller improvements. This showcases the difficulty of one-to-many translation, which essentially tries to learn a conditional language model and decode into multiple languages (Arivazhagan et al., 2019; Tang et al., 2020). As one-to-many translation forces different languages into one joint representation space, their linguistic

[1] Our source code is attached and will be made publicly available.

diversity is neglected. 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 automatically clustering the languages to account for linguistic similarities (Tan et al., 2019).

In this work, we propose using *language-family* adapters that enable efficient multilingual NMT for medium- and low-resource languages. Contrary to Stickland et al. (2021), we train a different set of adapters for each language family, to avoid negative transfer. Specifically, we train adapters for NMT on top of mBART-50 (Tang et al., 2020), a model pretrained on monolingual data of 50 languages. The adapters are trained using bi-text from each language family. Our method combines the advantages that adapters offer, due to their modularity, with the benefits of sharing information between similar languages. By taking into account linguistic families, we maximize positive cross-lingual transfer. This way, our model substantially outperforms all relevant baselines.

Our main contributions are:

- A novel, effective approach for multilingual translation which uses a multilingual pretrained model (with monolingual data) and fine-tunes it for each language family using adapters (with parallel data). In the English-to-many setting which we examine, language-family adapters achieve a +1.9 BLEU improvement over language-pair adapters, +1.1 BLEU improvement over language-agnostic adapters and are on par with multilingual fine-tuning, while being more efficient.
- We analyze the effect of our approach when adding *new* languages to mBART-50. LANG-FAMILY adapters improve translation by up to +2.7 BLEU compared to baselines.
- We analyze the importance of script for grouping languages. We also contrast grouping languages based on linguistic knowledge to grouping them using a combination of phylogenetic, typological, statistical and lexical features and provide insights.
- We propose inserting *embedding-layer adapters* to the Transformer to encode lexical information and conduct an ablation study to show that they contribute to better translation scores across all languages.

2 Background

2.1 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). We tackle this issue by training language-family adapters on top of mBART-50. Our approach takes advantage of language families and provides the flexibility necessary to decode into multiple languages. Our approach performs better than other adapter-based methods. It also performs as well as multilingual fine-tuning, while being more efficient.

2.2 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. Stickland et al. (2021) start from a pretrained model and train *language-agnostic* adapters (one set for *all* language pairs) to efficiently fine-tune a model for multilingual NMT.

Scaling language-agnostic adapters to a large number of languages is problematic, as when they are updated with data from multiple languages, negative transfer can occur. In contrast, language-

specific adapters do not face this problem, but at the same item do not allow any sharing between language pairs. Language-family adapters get the best of both worlds and our experimental results show that they lead to higher translation quality.

2.3 Language Families

Extensive work on cross-lingual transfer has demonstrated that leveraging one or more similar languages can improve the performance of a low-resource language 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; Tan et al., 2019; Oncevay et al., 2020; Kong et al., 2021). Linguistic knowledge bases (Littell et al., 2017; Dryer and Haspelmath, 2013) study language variation and can provide insights to phenomena such as negative interference. However, they typically are not used to train multilingual models. To properly model language variation, we cluster languages together using families from WALS (Dryer and Haspelmath, 2013). By taking into account linguistic information, we are able to improve MT.

3 Language-Family Adapters for NMT

Fine-tuning a pretrained model for multilingual NMT provides a good 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.

In this paper, 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. To this end, language-family adapters are trained on a linguistic family on top of mBART-50, a pretrained multilingual model.

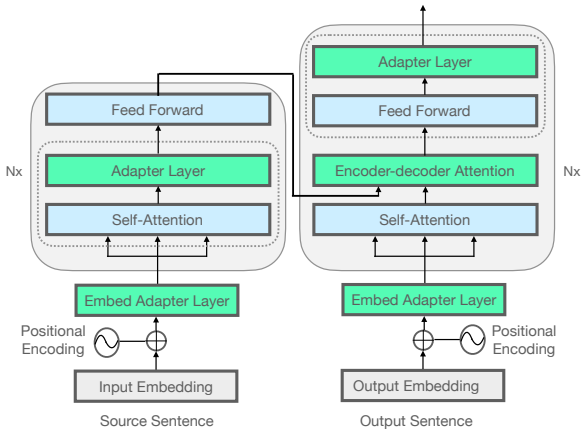


Figure 1: Proposed adapter architecture inside a Transformer model. Adapter layers and the encoder-decoder attention, shown in green, are trained for NMT. Figure best viewed in color.

3.1 Adapter Architecture

Adapters are typically added to a pretrained model in each Transformer layer. An adapter module 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 first 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 of the adapter module and the only tunable parameter. The up-projection is finally combined with a residual connection (He et al., 2016) with z_i:

$$Adapter_i(z_i) = U\,\mathrm{ReLU}(D\,\mathrm{LN}(z_i)) + z_i \quad (1)$$

3.2 Model Architecture

We insert an adapter after each *self-attention* layer in the encoder and after each *feed-forward* layer in the decoder. We also add an adapter after the *embedding layer* of both the encoder and the decoder. This is different from Bapna and Firat (2019), who added adapters only after the feed-forward layer for both the encoder and the decoder.

To train language-family adapters for NMT, we first freeze the pretrained encoder-decoder Transformer. We add adapters and fine-tune them on top of the pretrained model, together with the *encoder-decoder attention* layers (following Stickland et al. (2021); Üstün et al. (2021)). We train the model multilingually on each *language family*. Our proposed model architecture is depicted in Figure 1.

Because we keep the token embeddings of mBART-50 frozen, adding flexibility to the model to encode lexical information of the languages of

Language (code)	Family	Data	Script
⋆Bulgarian (bg)	Balto-Slavic	174k	Cyrillic
Persian (fa)	Indo-Iranian	151k	Arabic
⋆Serbian (sr)	Balto-Slavic	137k	Cyrillic
Croatian (hr)	Balto-Slavic	122k	Latin
Ukrainian (uk)	Balto-Slavic	108k	Cyrillic
Indonesian (id)	Austronesian	87k	Latin
⋆Slovak (sk)	Balto-Slavic	61k	Latin
Macedonian (mk)	Balto-Slavic	25k	Cyrillic
Slovenian (sl)	Balto-Slavic	20k	Latin
Hindi (hi)	Indo-Iranian	19k	Devanagari
Marathi (mr)	Indo-Iranian	10k	Devanagari
⋆Kurdish (ku)	Indo-Iranian	10k	Arabic
⋆Bosnian (bs)	Balto-Slavic	6k	Cyrillic
⋆Malay (ms)	Austronesian	5k	Latin
Bengali (bn)	Indo-Iranian	5k	Bengali
⋆Belarusian (be)	Balto-Slavic	5k	Cyrillic
⋆Filipino (fil)	Austronesian	3k	Latin

Table 1: 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.

interest is crucial, especially for low-resource, as well as unseen languages (languages that are not part of the pretraining corpus of mBART-50). Lexical cross-lingual information could be encoded by learning new embeddings for the languages we focus on (Artetxe et al., 2020). However, since token embeddings make up the majority of the parameters of the pretrained model, this would make the approach computationally expensive. Instead, we propose adding an adapter after the *embedding* layer, in both the encoder and the decoder (*embed adapter layer* in Fig. 1). The adapter layer receives as input the embedding-layer representation of each sequence and aims to capture token-level cross-lingual transformations. Our approach is inspired by *invertible adapters* (Pfeiffer et al., 2020) and simplifies their structure. The vocabulary of mBART-50 (and the embedding layer) was created using a very large number of languages and scripts. This permits us to fine-tune the model to unseen languages. However, we note that expanding the model to scripts that do not exist in the vocabulary of mBART-50 is not possible with our approach.

4 Experimental Setup

Data. We use TED talks (Qi et al., 2018) as the bitext for the 17 languages we fine-tune mBART-50 on. We choose 17 languages, 9 of which are present during pretraining, while 8 are new to mBART-50. These languages belong to 3 language families, namely Balto-Slavic, Austronesian and Indo-Iranian. We report in Table 1 information about the amount of parallel data available for each language, as well as their scripts and family.

Baselines. We compare the proposed language-family adapters with **1)** *multilingual fine-tuning* of mBART-50 (ML-FT), **2)** fine-tuning with *language-pair adapters* (LANG-PAIR), and **3)** fine-tuning with *language-agnostic adapters* (LANG-AGNOSTIC). ML-FT trains *all* layers of mBART-50 for NMT, using the data of all 17 languages. The second baseline, fine-tuning with LANG-PAIR adapters, is similar to Bapna and Firat (2019): a different set of adapters is trained for each language pair on top of mBART-50, so no parameters are shared for differing language pairs. Finally, the third baseline fine-tunes using all data with LANG-AGNOSTIC adapters (similar to Stickland et al. (2021)). This approach trains a single set of adapters using parallel data from all languages. Baselines **2)**, **3)** and our proposed approach train the same layers: adapters (as specified in §3.2) and the encoder-decoder attention. Baseline **1)** trains all layers.

Training details. We start from the trained mBART-50 model checkpoint[2], which is publicly available. We extend its embedding layers with randomly initialized vectors for an extra set of 8 languages, unseen during pretraining. We reuse the 250k sentencepiece (Kudo and Richardson, 2018) model of the original mBART and mBART-50 models. We use the fairseq (Ott et al., 2019) library for all experiments. We select the final models using validation perplexity. We use beam search with size 5 for decoding and evaluate BLEU scores using SacreBLEU[3] (Post, 2018). We also compute CHRF (Popović, 2015) and COMET (Rei et al., 2020) scores. For the latter, we rely on the `wmt-large-da-estimator-1719` pretrained model. Results using all metrics are reported in Appendix A.4.

We freeze all layers of mBART, except for the encoder-decoder attention. We add adapters and fine-tune the model for NMT. We fine-tune the LANG-FAMILY, LANG-AGNOSTIC adapters in a multilingual, one-to-many ($1 \rightarrow xx$) setup, using English as the source language. LANG-PAIR adapters are fine-tuned bilingually for each language pair. The adapters have a bottleneck dimension of 2048. We train each model for 130k updates with a batch size of 1024 tokens per GPU, using 2 NVIDIA-V100 GPUs. We evaluate models

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

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

	en → xx																	
	Balto-Slavic									Austronesian			Indo-Iranian					
	bg⋆	sr⋆	hr	uk	sk⋆	mk	sl	bs⋆	be⋆	id	ms⋆	fil⋆	fa	hi	mr	ku⋆	bn	Avg
Fine-tuning all layers																		
ML-FT	35.2	25.0	30.7	23.2	23.7	27.5	20.7	28.9	12.0	35.0	24.0	13.3	18.3	18.1	10.1	4.3	9.6	21.2
Fine-tuning adapters + encoder-decoder attention																		
Lang-pair	35.5	24.2	30.3	22.5	23.6	23.9	19.3	22.1	8.9	33.4	20.6	10.9	17.4	19.9	9.9	0.2	9.5	19.5
Lang-agnostic	34.0	24.4	28.7	21.7	22.0	27.0	21.6	28.2	8.7	34.1	23.3	11.6	17.8	18.3	9.9	4.3	9.6	20.3
Lang-family	35.2	24.6	**30.8**	22.5	**24.2**	27.0	**22.0**	**28.6**	**12.5**	**35.3**	**24.5**	**12.2**	**18.1**	19.2	**11.0**	**6.0**	**10.6**	**21.4**

Table 2: Test set BLEU scores when translating out of English ($en \rightarrow xx$). Languages are presented by decreasing amount of parallel data per language family. LANG-PAIR stands for language-pair, LANG-AGNOSTIC for language-agnostic, and LANG-FAMILY for language-family adapters. Languages denoted with ⋆ are *unseen* from mBART-50 during pretraining. Results in bold are significantly different ($p < 0.01$) when compared to the best adapter baseline.

after 5k training steps. To balance high and low-resource language pairs, we use temperature-based sampling (Arivazhagan et al., 2019) with $T = 5$. We use 0.1 dropout. We otherwise use the same hyper-parameters as Tang et al. (2020) and report them in Appendix A.2.

5 Results and Discussion

5.1 Main results

Table 2 shows translation results for 17 languages in terms of BLEU, using parallel data to fine-tune mBART-50 in the $en \rightarrow xx$ direction. We also evaluate the models using CHRF and COMET and report results in Appendix A.4.

Compared to fine-tuning with LANG-PAIR adapters, our approach (LANG-FAMILY) largely improves results, with an average +1.9 BLEU performance boost (+3.2 CHRF, +13.0 COMET) across all languages. This shows that cross-lingual representations from similar languages are beneficial to a multilingual model in a medium- or low-resource setup. As the LANG-PAIR approach trains a set of adapters on each language pair, it does not take advantage of cross-lingual signal, which could lead to a positive transfer. The effectiveness of our approach is pronounced in low-resource languages, where we observe up to +6.5 BLEU compared to LANG-PAIR. Moreover, LANG-PAIR adapters train 160M parameters for each language pair, or 2.7B parameters for the 17 languages examined. This scales linearly with the number of languages used, which makes this approach computationally burdensome. Our model trains 160M parameters per language family, or 480M in total, achieving more efficient fine-tuning.

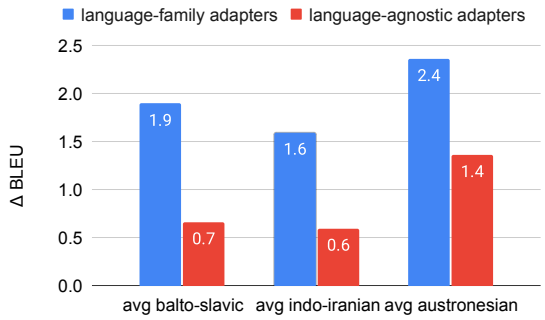
The most related baseline, LANG-AGNOSTIC, provides decent results, with a 20.3 average BLEU score across all language pairs, while it updates 160M parameters. Since it trains adapters using all parallel data, it is nonetheless susceptible to negative interference. Training a set of adapters jointly on languages from different linguistic families hinders the decoding ability of the multilingual model, as languages are fighting for model capacity. A good parameter-performance trade-off is achieved using LANG-FAMILY adapters, our proposed approach. As we fine-tune a model on multiple languages which are similar to each other using adapters, our model combines family-specific information and multilingual learning, achieving 21.4 BLEU, or a +1.1 improvement compared to LANG-AGNOSTIC (+1 CHRF, +3.7 COMET).

Finally, ML-FT provides an average BLEU score of 21.2 across all language pairs. This approach requires training the entire mBART-50 model (680M parameters) to parallel data of all 17 languages. Language-family (LANG-FAMILY) adapters, our proposed approach, is on par or even outperforms ML-FT (on par in terms of BLEU and CHRF, +1 COMET), while being more computationally efficient. Although LANG-FAMILY adapters keep the token embedding layer frozen, they manage to adapt to both seen and unseen languages and provide high-quality translations.

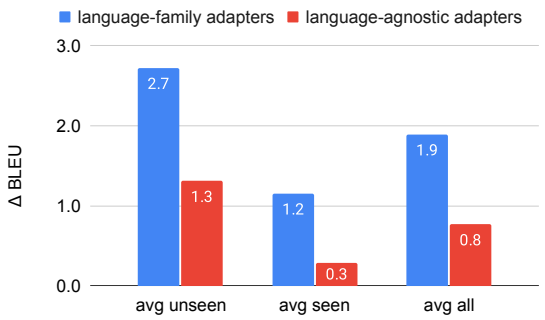
5.2 Performance according to language family

To evaluate the contribution of grouping languages based on linguistic information, we compute the difference of LANG-FAMILY adapters compared to the LANG-PAIR baseline *per language family* in terms of BLEU score. We show the results in Figure 2a. The LANG-PAIR baseline is displayed as the x-axis in Figures 2a, b.

Compared to the LANG-AGNOSTIC baseline, LANG-FAMILY adapters boost the translation scores for *all* language families (+1.2 BLEU for Balto-Slavic, +1 for Indo-Iranian, +1 the Austronesian) by a similar degree. This is the case



(a) Grouping based on language family.



(b) Grouping based on "seen" (existing in the pretraining corpus), or "unseen" language.

Figure 2: Average translation scores (BLEU) of language-family adapters for $en \rightarrow xx$, (a) according to language family and (b) based on whether the language is seen or unseen during pretraining. We show the difference versus the language-pair adapters baseline (depicted as the x-axis).

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.

Compared to the LANG-PAIR baseline, Austronesian LANG-FAMILY adapters provide an impressive +2.4 BLEU performance boost. This particular language family largely benefits from sharing between similar languages, probably because mBART-50 has been exposed to just one Austronesian language during pretraining, namely Indonesian (`id`). As a result, the model has a limited knowledge of how to encode Austronesian languages. LANG-FAMILY (and, to a lesser extent, LANG-AGNOSTIC) adapters permit cross-lingual transfer between related languages and can provide more accurate translations. There is also a clear, although less pronounced improvement for Balto-Slavic and Indo-Iranian languages, which is reasonable, as there are many languages from these families in the pretraining corpus of mBART-50.

5.3 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 2b. We observe that LANG-FAMILY adapters boost the translation quality considerably (+2.7 BLEU) 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. While the performance of LANG-AGNOSTIC adapters is hindered by negative transfer, sharing information between languages is still preferable to training LANG-PAIR adapters, as it leads to a better translation quality, as shown by a +1.3 BLEU improvement.

LANG-FAMILY adapters that are fine-tuned on *seen* languages yield a smaller but clear improvement (+1.2 BLEU). LANG-AGNOSTIC adapters, however, fail to learn multilingual representations that would provide extra information to the model for the languages of interest. Therefore, they perform on par with LANG-PAIR adapters.

Overall, fine-tuning with LANG-PAIR adapters is the weakest baseline. This intuitively makes sense, as LANG-PAIR adapters do not harness linguistic similarities and encode each language independently, with a different set of parameters. On medium and low-resource language pairs LANG-PAIR adapters are not expressive enough due to data scarcity.

5.4 Performance based on dataset size

LANG-FAMILY adapters, our proposed approach, outperforms LANG-AGNOSTIC, the most related baseline by +1.1 on average on all language pairs. We want to quantify the improvement based on the parallel data size. Results for language pairs based on their dataset sizes are displayed in Table 3.

Our method outperforms the baseline by a larger margin in the extremely low-resource language pairs (+1.4 BLEU points for languages with 3-10k sentences) and the improvement becomes smaller, when there are larger parallel sets available (+0.9

BLEU points for language pairs with more than 100k sentences). Our method is well fit for low-resource languages, as it enables positive transfer. In low-resource setups, the translation model heavily relies on cross-lingual representations from similar language pairs to provide decent translations.

Data	Lang-family	Lang-agnostic	ΔBLEU
>100k	26.2	25.3	0.9
10k-100k	20.7	19.6	1.1
3-10k	20.7	19.6	1.4
All	21.4	20.3	1.1

Table 3: Difference of LANG-FAMILY adapters (our approach) compared to LANG-AGNOSTIC in terms of BLEU when grouping languages according to dataset sizes.

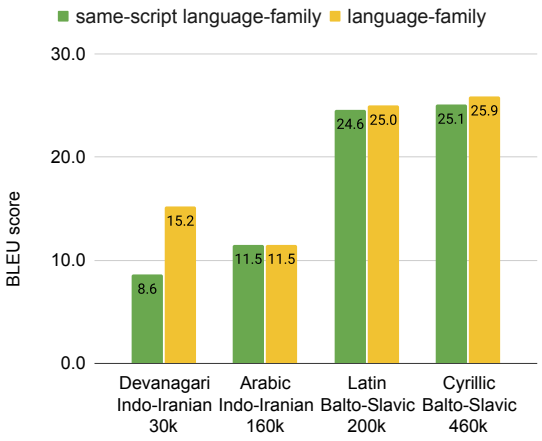


Figure 3: BLEU scores on Indo-Iranian and Balto-Slavic, using either *same-script language-family* or *language-family adapters*. From left to right on x-axis, we go from low-resource to high-resource same-script families.

5.5 Performance based on script

When training multilingual models, a natural question that arises is whether script mismatch hinders cross-lingual transfer. We try to answer that by training language-family adapters in two distinct setups: one that takes into consideration script (by grouping together languages of the *same-script language-family*) and another that clusters related languages of different scripts together (*language-family*). We present the results in Figure 3.

We first focus on Indo-Iranian languages. `Hi` and `mr` use the Devanagari script, while `fa` and `ku` use the Arabic script. For the Devanagari-script languages, BLEU scores greatly improve (by +6.6) when *all* Indo-Iranian languages are used, independent of their script. As the Devanagari Indo-Iranian languages are low-resource (30k parallel sentences), we hypothesize that the adapters are undertrained. However, using all Indo-Iranian languages means we have more parallel data available (190k sentences). The pronounced improvement is probably due to the additional data provided. For the Arabic script Indo-Iranian languages, script does not play a crucial role. Since this language group is medium-resource (160k sentences), training adapters using same-script languages performs comparably with training adapters on the whole language family.

We then present results on Balto-Slavic languages. `Hr,sl,sk` use the Latin script, while `bg,sr,uk,mk,be` use the Cyrillic script. For both subgroups of this language family, which are medium-resource (>100k of parallel data), using same-script languages to fine-tune the adapter models again performs comparably to using all languages of this language family.

We find that only taking into consideration same-script languages of a linguistic family is harmful for the translation quality of a model, if there is limited data available. When enough data is available, grouping together same-script languages is on par or slightly worse than using all of the data for a language family.

Unseen Language	**LANGRANK with $K = 5$**				
Bulgarian (bg)	mk	bg	sr	uk	sl
Serbian (sr)	hr	sr	sl	be	mk
Slovak (sk)	sk	hr	sr	fa	bg
Bosnian (bs)	hr	sr	bg	sl	uk
Belarusian (be)	uk	sr	bg	sl	hr
Malay (ms)	id	fa	bg	ms	sr
Kurdish (ku)	ku	fa	bg	uk	hr

Table 4: Ranking of candidates for cross-lingual transfer using LANGRANK.

	bg	sr	sk	bs	be	ms	ku	avg
LR	**35.8**	**25.5**	23.8	**29.1**	10.9	**24.7**	5.1	22.1
Family	35.2	24.6	**24.2**	28.6	**12.5**	**24.5**	**6.0**	22.2

Table 5: BLEU scores using adapters trained on LANGRANK clusters (*LR*), compared to our approach, that trains adapters using linguistic families (*family*).

5.6 Grouping languages using LANGRANK

For our main set of experiments, we used language families from WALS. However, not all languages in a single language family share the same linguis-

	en → *xx*									
	BALTO-SLAVIC			AUSTRONESIAN			INDO-IRANIAN			AVG-17
	bg	hr	be	id	ms	fil	fa	ku	bn	
Ft enc-dec attention, add adapters after:										
(1): self-att (enc), ffn (dec)	34.4	30.0	12.0	34.3	23.6	12.1	17.7	4.9	10.2	20.8
(2): self-att (enc), ffn (dec) + enc-dec emb	**35.2**	**30.8**	**12.5**	**35.3**	**24.5**	12.2	**18.1**	**6.0**	**10.6**	**21.4**

Table 6: Ablation of the proposed architecture for $en \rightarrow xx$ (BLEU scores). *Self-att* stands for the self-attention layer, *ffn* for the feed-forward layer, *emb* for the token embedding layer, *enc* for the encoder and *dec* for the decoder. Results in bold are significantly different ($p < 0.01$) when compared to model (1). We present results only for 3 languages per language family due to space constraints. Results on all languages can be found in Appendix A.5.

tic properties (Ahmad et al., 2019). LANGRANK (Lin et al., 2019) represents languages as a set of attributes that include typological information and corpus statistics and, for a given language, ranks the languages that are most helpful for cross-lingual transfer. They suggest that dataset size and word-level overlap might be better indicators than linguistic families for transfer learning in NMT.

We want to empirically assess whether using the candidate languages indicated by LANGRANK for each of the unseen languages is more helpful than using linguistic families. We train LANG-FAMILY adapters for each unseen language, with the group of languages created using LANGRANK with $K = 5$, where K is the number of transfer languages. As we see in Table 4, the unseen language itself is not always predicted by the ranking model (e.g., `be`). We nonetheless always use the unseen language to train the respective adapter (e.g., for `be`, we use `be,uk,sr,bg,sl` and `hr`).

We see in Table 5 that training adapters using language groups computed by LANGRANK (denoted as LR) performs on par with our proposed approach (denoted as *family*), which uses language groups defined from linguistic families in WALS. We note that features used by LANGRANK, such as typology, are not available for many languages. Training adapters according to the candidate languages that LANGRANK predicts also means that a different set of adapters needs to be trained for every language of interest. For these reasons, our approach is more efficient and applicable to a wider range of languages.

5.7 Embedding-layer adapter

When mBART-50 is fine-tuned for NMT, encoder embeddings are tied to decoder embeddings. This is usually the case for NMT models, as weight tying reduces their size without harming their performance (Press and Wolf, 2017). In our proposed approach, the encoder and decoder embeddings are also not updated during fine-tuning. We hypothesize that this hinders the translation ability of the model, as it struggles to encode token-level representations that would be fed to the encoder or decoder and finally result in better decoding to the target languages.

To encode useful lexical representations, we introduce an adapter after the *encoder embedding layer*, as well as after the *decoder embedding layer*. We do not tie the adapter layers, since they only add up a small number of parameters (5M each, i.e., 0.7% of mBART-50 parameters). As we can see in Table 6, we get consistent gains across almost all language pairs by adding these adapters.

6 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 avoid negative transfer and perform on par with multilingual fine-tuning, while being more efficient. Moreover, our model outperforms established baselines, such as language-pair adapters and language-agnostic adapters. Finally, for languages new to mBART-50, our approach provides an effective way of leveraging shared cross-lingual information between similar languages, largely improving translations versus the baseline.

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.

References

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

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2440–2452.

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 Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637.

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 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548.

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 Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

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

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

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

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

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

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*, pages 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 Conference of the European Chapter of the Association for Computational Linguistics*, pages 1613–1624.

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

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135.

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 Conference of the European Chapter of the Association for Computational Linguistics*, pages 8–14.

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*, pages 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 Conference on Empirical Methods in Natural Language Processing*, pages 875–880.

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

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 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

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 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 392–395.

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

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 157–163.

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 529–535.

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 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

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 Conference on Machine Translation: Research Papers*, pages 261–271.

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 Conference of the European Chapter of the Association for Computational Linguistics*, pages 3440–3453.

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*, pages 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 Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 963–973.

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 589–598.

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*.

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 Conference on Empirical Methods in Natural Language Processing*, pages 2955–2960.

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 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450.

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

A Appendix

A.1 Dataset statistics

We present the statistics of all parallel data used in our set of experiments in Table 7. We note that the number of train, validation and test set presented refers to sentences.

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

Table 7: Dataset stats. We use data from TED (Qi et al., 2018) and TED2020 (Reimers and Gurevych, 2020).

A.2 Hyperparameters

We list the hyperparameters we used to train both our proposed model and the baselines in Table 8. We do not tune the hyperparameters. We note that the adapters were used for the LANG-PAIR, LANG-AGNOSTIC baselines and the LANG-FAMILY proposed approach (not for the ML-FT baseline).

A.3 Average runtime and Parameters

We present in Table 9 the average runtime for our proposed model and the baselines, as well as model parameters that are updated during training. We note that our proposed model has to run for each language family, while the LANG-PAIR baseline has to run for each language pair separately.

A.4 Evaluation of main results using 3 metrics

We evaluate the translations of our model (LANG-FAMILY adapters) and all the baselines using additional metrics besides BLEU, namely COMET (Rei et al., 2020) and CHRF (Popović, 2015). 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. CHRF uses character n-grams (instead of word n-grams in BLEU) to compare the translated output with the reference and, compared to BLEU, it can better match morphological variants of words. Our results are summarized in Table 10. We see that both CHRF and COMET correlate with BLEU in our experiments.

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	3e-05
Temperature sampling	5
Adapter dim.	2048

Table 8: Fairseq hyperparameters used for our set of experiments.

Approach	Runtime	Parameters
LANG-FAMILY (ours)	20	160M
LANG-AGNOSTIC	36	160M
LANG-PAIR	10	160M
ML-FT	40	680M

Table 9: Average runtime per approach (hours) and trainable parameters.

A.5 Full results of ablation study

We report in Table 11 the results of the ablation study on all 17 languages.

Lang. Family		ML-FT			LANG-PAIR			LANG-AGNOSTIC			LANG-FAMILY (ours)		
		BLEU	CHRF	COMET	BLEU	CHRF	COMET	BLEU	CHRF	COMET	BLEU	CHRF	COMET
BALTO-SLAVIC	bg	35.2	58.7	64.9	35.5	58.7	67.9	34.0	56.9	60.2	35.2	58.0	65.9
	sr	25.0	44.3	67.0	24.2	42.4	64.2	24.4	43.5	67.8	24.6	43.8	69.0
	hr	30.7	55.9	74.3	30.3	54.7	75.3	28.7	55.1	77.5	30.8	55.7	78.5
	uk	23.2	45.7	48.2	22.5	44.8	48.0	21.7	45.2	48.8	22.5	45.9	50.7
	sk	23.7	47.7	55.4	23.6	46.7	55.0	22.0	45.7	50.8	24.2	47.3	58.1
	mk	27.5	52.7	54.0	23.9	48.3	41.2	27.0	52.4	55.4	27.0	53.1	58.5
	sl	20.7	44.8	43.9	19.3	42.0	33.9	21.6	44.9	47.8	22.0	45.8	52.0
	bs	28.9	53.8	72.9	22.1	45.7	39.0	28.2	52.3	74.2	28.6	53.7	76.5
	be	12.0	32.0	-8.4	8.9	26.8	-54.6	8.7	27.4	-42.6	12.5	30.2	-29.3
AUSTRONESIAN	id	35.0	58.9	61.2	33.4	58.1	61.4	34.1	59.0	61.8	35.3	59.1	61.9
	ms	24.0	50.0	51.3	20.6	46.3	27.0	23.3	49.1	48.8	24.5	49.9	52.0
	fil	13.3	38.8	-24.8	10.9	35.8	-33.6	11.6	37.9	-24.1	12.2	38.2	-26.1
INDO-IRANIAN	fa	18.3	40.0	37.8	17.4	38.8	37.8	17.8	39.2	38.3	18.1	39.1	36.5
	hi	18.1	37.8	14.1	19.9	37.0	15.2	18.3	37.5	10.0	19.2	38.6	10.7
	mr	10.1	29.3	-21.1	9.9	28.0	-25.2	9.9	28.6	-22.3	11.0	29.9	-16.0
	ku	4.3	24.4	-37.2	0.2	9.4	-94.1	4.3	24.2	-41.8	6.0	25.9	-33.8
	bn	9.6	31.3	-19.7	9.5	28.5	-29.4	9.6	29.7	-22.2	10.6	31.5	-14.4
AVG		**21.2**	**43.9**	31.4	19.5	40.7	19.4	20.3	42.9	28.7	**21.4**	**43.9**	**32.4**

Table 10: Test set BLEU, CHRF and COMET scores when translating out of English. 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. Best results on average are indicated with bold.

	en → xx																	
	Balto-Slavic									**Austronesian**			**Indo-Iranian**					
	bg	sr	hr	uk	sk	mk	sl	bs	be	id	ms	fil	fa	hi	mr	ku	bn	AVG
Ft enc-dec attention, add adapters after:																		
(1): self-att (enc), ffn (dec)	34.4	24.2	30.0	21.6	23.5	26.7	21.1	27.5	12.0	34.3	23.6	12.1	17.7	19.6	10.9	4.9	10.2	20.8
(2): (1) + enc-dec emb	35.2	24.6	30.8	22.5	24.2	27.0	22.0	28.6	12.5	35.3	24.5	12.2	18.1	19.2	11.0	6.0	10.6	21.4

Table 11: Full results for Table 6 in the main paper. Ablation of the proposed model architecture (LANG-FAMILY adapters) for $en \rightarrow xx$.