Multilingual Domain Adaptation for NMT: Decoupling Language and Domain Information with Adapters

Anonymous ACL submission

Abstract

Adapter layers are lightweight, learnable units inserted between transformer layers. Recent work explores using such layers for neural machine translation (NMT), to adapt pretrained models to new domains or language pairs. We propose strategies to compose language and domain adapters. Our goals are both parameter-efficient adaptation to multiple domains and languages simultaneously, and cross-lingual transfer in domains where parallel data is unavailable for certain language pairs. We find that a naive combination of domain-specific and language-specific adapters often results in translations into the wrong language. We study other ways to combine the adapters to alleviate this issue and maximize cross-lingual transfer. With our best adapter combinations, we obtain improvements of 3-4 BLEU on average for source languages that do not have in-domain data. For target languages without in-domain data, we achieve a similar improvement by combining adapters with back-translation.

1 Introduction

Multilingual Neural Machine Translation (NMT) has made a lot of progress recently (Johnson et al., 2017; Bapna and Firat, 2019; Aharoni et al., 2019; Zhang et al., 2020; Fan et al., 2020a) and is now widely adopted by the community and MT service providers. Multilingual NMT models handle multiple language directions at once and allow for knowledge transfer to low-resource languages. Machine translation systems often need to be adapted to specific domains like legal or medical text. However, when building multilingual systems, data for most language pairs might not exist. We would ideally be able to leverage data in a subset of language pairs to transfer domain knowledge to many others.

A technique for adapting such models to new language-pairs and domains are the recently introduced 'adapter layers' (Bapna and Firat, 2019), lightweight, learnable units inserted between transformer layers. Previous studies have shown it is possible to combine language adapters (Philip et al., 2020), or language and task adapters (Pfeiffer et al., 2020) trained independently, enabling zero-shot compositions of adapters. In this work we analyse how to combine *language adapters* with *domain adapters* in multilingual NMT, and study whether domain knowledge can be transferred across languages.

We show it is hard to decouple language knowledge from domain knowledge when finetuning multilingual MT systems on new domains. In Section 5.2 we demonstrate that adapters learnt on a subset of language pairs fail to generate into languages not in that subset. Such generation into the wrong language is referred to as 'off-target' translation. Our initial results show 'stacking' (or composing) language and domain adapters can improve performance, but combinations of domain and language adapters unseen at training time lead to bad performance. We examine how adapter placement and other techniques can improve the compositionality of language and domain adapters when dealing with source or target languages that do not have in-domain data (which we refer to throughout this work as "**out-of-domain languages**"). Our key contributions are:

- We examine adapter placement for simple bilingual domain adaptation as well as multilingual multi-domain adaptation, and show that encoder-only adapters can be just as effective as default adapters added in every layer.

- We analyse different language and domain adapter combinations that improve performance and reduce off-target translations. Our best results for translation into out-of-domain languages use decoder-only domain adapters, regularisation with domain adapter dropout, and data augmentation with English-centric back-translation.

2 Related Work

Cross-lingual transfer Many works have demonstrated that large pre-trained multilingual models (Devlin et al., 2019; Conneau et al., 2020; Liu et al., 2020) fine-tuned on high-resource languages (or language pairs) can transfer to lower-resource languages in various tasks: Natural Language Inference (Conneau et al., 2018), Question Answering (Clark et al., 2020), Named Entity Recognition (Pires et al., 2019; K et al., 2020), Neural Machine Translation (Liu et al., 2020) and others (Hu et al., 2020).

Domain adaptation in NMT Domain adaptation has been discussed extensively for bilingual NMT models. A typical approach is to fine-tune a model trained on a large corpus of 'generic' data on a smaller in-domain corpus (Luong and Manning, 2015; Neubig and Hu, 2018). A common technique to make use of monolingual in-domain data is to do back-translation (Sennrich et al., 2016a; Berard et al., 2019a; Jin et al., 2020). Multi-domain models can be trained with domain tags (Kobus et al., 2017; Berard et al., 2019a; Stergiadis et al., 2021). In this work we focus on multilingual domain adaptation, where we hope to transfer domain knowledge from one language pair to many others. This setting presents challenges for back-translation, since for n languages and k domains we need to run back-translation $O(n^2k)$ times.

Adapter layers Bapna and Firat (2019) introduce adapter layers for NMT as a lightweight alternative to finetuning. They study both adding language-pair specific adapters to multilingual NMT models to match the performance of a bilingual version, and domain-specific adapters for parameter-efficient domain adaptation. Further, Philip et al. (2020) show that decomposing language-specific adapters into independently trained language adapters improves zero-shot translation in English-centric settings, and can also be used to adapt a model to all language directions in a scalable way. Pfeiffer et al. (2020) study adapter layers in the context of pre-trained Language Models. They compose language adapters trained on masked language modelling in language x and task adapters trained on classification tasks in language y and obtain transfer to classification in language x. Our work pursues a similar objective to Pfeiffer et al. (2020), but for NMT where in addition to encoding sentences we need to generate text for new language and domain combinations.

3 Composing Adapter Modules

Adapter modules (Rebuffi et al., 2017; Houlsby et al., 2019) are randomly initialised modules inserted between the layers of a pre-trained network and fine-tuned on new data. An adapter layer is typically a down projection to a bottleneck dimension followed by an up projection to the initial dimension, which we write as $\mathrm{FFN}(\mathbf{h}) = W_{\mathrm{up}} f(W_{\mathrm{down}}\mathbf{h})$, with $f(\cdot)$ a non-linearity. The bottleneck controls the parameter count of the module; typically NMT requires slightly larger parameter counts than classification to match fine-tuning (Bapna and Firat, 2019; Cooper Stickland et al., 2021). With a residual connection and a near-identity initialization the original model is (approximately) retained at the beginning of optimization, keeping at least the performance of the parent model.

3.1 Stacking Domain and Language Adapters

In this work we study 'stacking' adapter modules, i.e. each language and domain has a unique adapter module associated with it. When passing a batch with source language x, target language y, and domain z, we only 'activate' the adapters for $\{x, y, z\}$. The encoder adapters for x and decoder adapters for y are activated.

We mostly follow the architecture of Bapna and Firat (2019). Language adapters LA are defined as:

$$\mathrm{LA}(\mathbf{h}_l) = \mathrm{FFN}_{\mathrm{lg}}(\mathrm{LN}_{\mathrm{lg}}(\mathbf{h}_l)) + \mathbf{h}_l \qquad (1)$$

where $\mathbf{h}_l$ is the Transformer hidden state at layer l and $\mathrm{LN}_{\mathrm{lg}}$ is a newly initialised layer-norm. Let $\mathbf{z} = \mathrm{LA}(\mathbf{h}_l)$; when stacking domain and language adapters, the layer output $\mathbf{h}_{l,\mathrm{out}}$ is given by:

$$\mathbf{h}_{l,\mathrm{out}} = \mathrm{FFN}_{\mathrm{dom}}(\mathrm{LN}_{\mathrm{dom}}(\mathbf{z})) + \mathbf{z} \qquad (2)$$

Since we simply apply another adapter on top of the language adapter we refer to this as '**modular**' style.

Pfeiffer et al. (2020) use a different formulation that empirically performed well, with:

$$\mathrm{LA}(\mathbf{h}_l, \mathbf{r}_l) = \mathrm{FFN}_{\mathrm{lg}}(\mathbf{h}_l) + \mathbf{r}_l. \qquad (3)$$

The residual connection $\mathbf{r}_l$ is the output of the Transformer's feed-forward layer whereas $\mathbf{h}_l$ is the output of the subsequent layer normalisation. When stacking domain and language adapters the layer output is given by applying the model's pre-trained layer norm $\mathrm{LN}_{\mathrm{pre}}$,

$$\mathbf{h}_{l,\mathrm{out}} = \mathrm{LN}_{\mathrm{pre}}(\mathrm{FFN}_{\mathrm{dom}}(\mathrm{LA}(\mathbf{h}_l, \mathbf{r}_l)) + \mathbf{r}_l) \quad (4)$$

and using the output of the Transformer's feed-forward layer as a residual instead of the language adapter output. We refer to this as '**MAD-X**' style after Pfeiffer et al. (2020). This leaves the layer output 'closer' to the pre-trained model, with the same layer-norm and residual connection, contrary to Eq. 2 which has a newly initialised layer-norm and a residual connection. For all models without any stacking we obtain layer output as in Eq. 2 or Eq. 4 but replace $\mathrm{LA}(\cdot)$ with the identity operation.

3.2 Improving the Compositionality of Adapters

In our initial experiments (Section 5.2) we found that (unlike Pfeiffer et al., 2020) naive stacking of language and domain adapters does not work very well for unseen combinations of language and domains, and often results in off-target translation (i.e. translations into the wrong language). Therefore, we study several strategies to improve the compositionality of adapters in the context of NMT:

1) Using **decoder-only** domain adapters when translating from an out-of-domain source language into an in-domain[1] target language, and **encoder-only** domain adapters when translating from an in-domain source language into an out-of-domain target language. This means we never stack together a combination of language and domain adapter that was not seen at training time.

2) **Domain adapter dropout** (DADrop). Similar to layer-drop (Fan et al., 2020b) but specialised to adapter layers, or AdapterDrop (Rücklé et al., 2020) but without targeting specific layers, we randomly 'drop' (i.e. skip) the domain adapter[2] and only pass the hidden state through the language adapter. This means the adapter stack in the layer above can more easily adapt to unfamiliar input, and encourages domain and language adapters to be more independent of each other.

3) **Data augmentation**. We often have access to monolingual data in a domain even when no parallel data is available. In this work we leverage English-centric back-translation (BT), i.e. translating monolingual data in some languages into English (thus avoiding the more expensive step of translating from each language into every other language). We examine the ability of such data to help cross-lingual transfer to unseen combinations of source and target language (BT means we have artificial data for every language in combination with English). We briefly explore '*denoising auto-encoder*' style objectives as in unsupervised MT (Lample et al., 2018) or sequence-to-sequence pre-training (Lewis et al., 2020).

4 Experimental Settings

4.1 Data

For bilingual domain adaption we use the same datasets as Aharoni and Goldberg (2020), namely parallel text in German and English from five diverse domains: Koran, Medical, IT, Law and Subtitles. For studying the domain transfer across languages we select four diverse domains that have data available in most language directions: translations of the Koran (**Koran**); medical text from the European Medicines Agency (**Medical**); translation of TED Talks transcriptions (**TED**); various technical IT text, e.g. the Ubuntu manual (**IT**). All data was obtained from the OPUS repository (Tiedemann, 2012). We create validation and test sets of around 2000 sentences each, and avoid overlap with training data (including parallel sentences in any language) with a procedure described in Appendix A. Note that Medical, Koran and IT are from the same source as those of Aharoni and Goldberg (2020), although the train/test splits are different.

Domain	Langs.	Avg size (lines)
ParaCrawl	12	125M
Koran	10	52k
Medical	11	500k
IT	12	196k
TED	12	138k

Table 1: Basic statistics for the datasets we use; number of languages covered, and average number of training examples across all language directions.

4.2 Baselines

For **bilingual** domain adaptation we use a Transformer Base (Vaswani et al., 2017) model trained

[1] Reminder we refer to the subset of languages we have parallel data for in a particular domain as 'in-domain', and all other languages as 'out-of-domain'.

[2] We could additionally drop the language adapter, but since this was frozen in many experiments we limit ourselves to domain adapters for simplicity

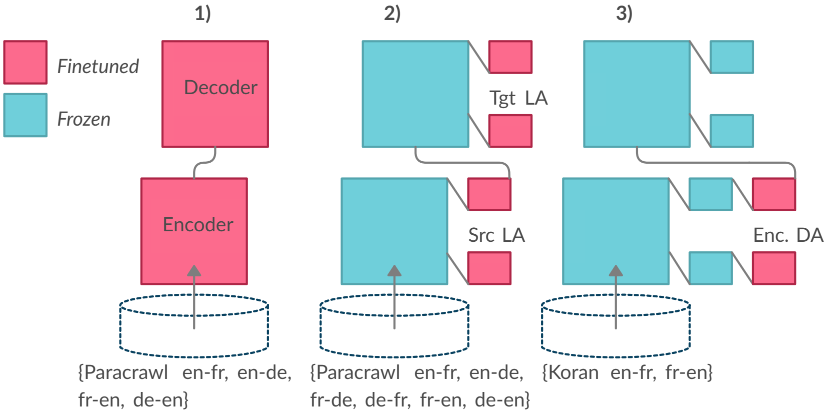
1)
2)
3)
Finetuned
Frozen
Decoder
Encoder
Tgt LA
Src LA
Enc. DA
{Paracrawl en-fr, en-de, fr-en, de-en}
{Paracrawl en-fr, en-de, fr-de, de-fr, fr-en, de-en}
{Koran en-fr, fr-en}

Figure 1: Toy diagram showing one of our proposed pipelines for training language and domain adapters, on a example subset of languages: {en,fr,de}, with 'domain-agnostic' data from ParaCrawl and specialised data from the Koran. Red indicates a fine-tuned model component, blue indicates a frozen component. LA = language adapter, DA = domain adapter. From left to right we show: 1) Training an encoder-decoder model with English-centric ParaCrawl. 2) Training mononlingual language adapters with multiparallel Paracrawl data. 3) Training domain adapters stacked on language adapters in the encoder, on a subset (here {en, fr}) of languages for the domain of interest (e.g. Koran). Here we show domain adapters added only to the encoder, but we consider various other configurations in this work.

for 12 epochs on German to English WMT20 data (47M parallel lines), with a joint BPE (Sennrich et al., 2016b) vocabulary of size 24k with inline casing (Berard et al., 2019b) (i.e. wordpieces are put in lowercase with a special token indicating their case.).

In **multilingual settings** we concentrate on 12 high-resource European languages[3] due to the availability of domain-specific parallel data for most language pairs. Our **baseline model** is a Transformer Base trained on English-centric ParaCrawl v7.1 data (Bañón et al., 2020) with all 12 languages (803M line pairs in total). It is trained with fairseq (Ott et al., 2019) for 800k updates, with a batch size of maximum 4000 tokens and accumulated gradients over 64 steps (Ott et al., 2018).[4] The source/target embeddings are shared and tied with the output layer. We tokenize the data with a shared BPE model of size 64k with inline casing (Berard et al., 2019b) Both the multilingual models and BPE model are trained with temperature-based sampling with $T = 5$ (Arivazhagan et al., 2019). We calculate all BLEU scores with Sacrebleu[5] (Post, 2018). We use adapter bottleneck size of 1024 unless stated otherwise, and when using DADrop (Section 3.2) use a 20% chance of skipping the domain adapter.

We additionally train monolingual language adapters (Philip et al., 2020) for all 12 languages on multi-parallel ParaCrawl data, which we obtain by aligning all languages through their English side, like Freitag and Firat (2020). The adapters are trained for another 1M steps, without accumulated gradients. Finally, for each domain we report the performance of models fine-tuned on all the language-pair directions, which serves as an upper bound for cross-lingual transfer. More training hyper-parameters are given in Appendix A.

4.3 Our model pipelines

Scenario 1): we adapt the English-centric ParaCrawl pre-trained model to all four domains and every language direction simultaneously. This approach tests for parameter-efficient multi-domain, multilingual domain adaptation. There is no cross-lingual transfer needed since all language directions are included in the training data. Results for this scenario and for bilingual domain adaptation are reported in Section 5.1.

Scenario 2): This is described in a toy diagram in Figure 1. We first extend the baseline multilingual English-centric model with 12 (one for each language) monolingual language adapters (Philip

[3] {cs, da, de, en, es, fr, it, nb, nl, pl, pt, sv}

[4] This corresponds to an effective batch size of ≈207k tokens and training length of 7 epochs.

[5] Signature: BLEU+case.mixed+lang.m2m-en+numrefs.1+smooth.exp+tok.13a+version.1.5.0.

et al., 2020) trained on multi-parallel Paracrawl. We then test the cross-lingual domain transfer ability of our proposed combinations of adapters by training on data in a particular domain with a subset of four languages (referred to as '**in-domain**'; in Figure 1 *en* and *fr* would be in-domain). We test our model on all language directions from the set of all twelve languages. This will include cases where we don't have in-domain data for either the source or target language, which we refer to as '**out-of-domain**' (in Figure 1 *de* would be out-of-domain).

Finally, we extend the above mentioned scenario with back-translated (BT) data from *out-of-domain* languages into English. To create the BT data, we use the model with language adapters trained on ParaCrawl **(18)** (which has not seen any in-domain data) on the English-aligned training data for each each language and domain, and use beam search with a beam size of 5. Results for this scenario are reported in Section 5.2.

To train language and domain adapters, we freeze all model parameters except for adapter parameters, and use a fixed learning rate schedule with learning rate 5×10^{-5}. Following Philip et al. (2020), when training language adapters without domain adapters we build homogeneous batches (i.e. only containing sentences for one language direction) and activate only the corresponding adapters. When training language *and* domain adapters together, we build homogeneous batches that only contain sentences for the same combination of language direction and domain.

5 Results and Discussion

First, in Section 5.1 we discuss different ways of distributing adapter parameters given a fixed parameter budget in bilingual and multilingual settings. In Section 5.2 we analyse the capacity of adapters for domain transfer across languages. We first demonstrate problems with cross-lingual generalisation during domain adaption for 'naive' methods, and then discuss solutions.

5.1 Where are adapters most effective?

Bilingual domain adaptation Before studying multilingual domain adaptation, we validate some of our ideas on a simpler, *bilingual* German → English domain adaptation setting. Table 6 reports the results of this experiment. First, we note that *encoder-only* adapters perform similarly to *encoder & decoder* adapters, while *decoder-only* adapters perform worse.

Moreover, adding adapters to only the last three layers of the encoder almost matches the performance of adapting every layer, while adding adapters to the first three layers decreases performance. We believe this is because the last encoder layer directly influences every layer of the decoder through cross-attention. We find the same trends at a smaller adapter size with a bottleneck dimension of 64 (see Appendix B), and for multilingual models.

The strong performance of encoder-only adapters has interesting implications for inference speed. With an auto-regressive decoder, the computational bottleneck is on the decoder side. The encoder output is computed all at once, while computing the decoder output requires L steps, where L is the output length. Inference speed-ups are modest due to the small size of adapters, e.g. around 10% and 8% faster inference for encoder-only vs decoder only for the models of this section and Section 5.2 respectively.

Multilingual multi-domain models Table 3 reports the results in the more challenging task of adapting a multilingual NMT model to multiple domains and language directions simultaneously. In this scenario, we assume access to in-domain data in all the language directions.

Stacking domain and language adapters (last 3 rows of Table 3) gives the best performance, compared to language adapters with the same parameter budget. We believe this is because it allows the model to (partially) decouple domain information from language-specific information and better exploit the allocated parameter budget. *Encoder-only domain adapters* combined with LA outperform much larger LA, and are faster at inference. Increasing capacity further with encoder + decoder DA improves performance mainly for IT and Koran[6]. The better results for encoder-only DA compared to decoder-only DA are coherent with the results reported for the bilingual setting. However, we will see in the next section that this sometimes comes at the expense of cross-lingual transfer.

5.2 Analysis of cross-lingual domain transfer

To study the capacity of our models to transfer domain knowledge across languages, we do domain

[6] Note that while these models increase the parameter count significantly, the effect on inference speed is limited since only one set of adapters is activated in each batch (assuming we have homogeneous batches.)

ID	Model	IT	Koran	Medical	Subtitles	Law
(1)	No finetuning	35.3	14.8	38.1	26.8	42.4
(2)	Finetuned	43.8	22.7	53	30.9	57.9
(3)	Enc. + dec. adapters $(d = 1024)$	42.9	21.8	51.7	30.5	56
(4)	(3) + MAD-X style	40.6	19.3	48.8	29.8	54.3
(5)	Dec. adapters $(d = 2048)$	42.1	19.8	50.5	29.7	55.1
(6)	Enc. adapters $(d = 2048)$	42.4	21.5	51.9	30.1	56.1
(7)	Last 3 encoder layers only $(d = 4096)$	42.9	21.1	52.1	30.1	56
(8)	First 3 encoder layers only $(d = 4096)$	42.2	20	50.1	28.5	54.9

Table 2: BLEU scores of various domain adaptation strategies for a German → English bilingual model. $(d = N)$ refers to adapters with a bottleneck dimension of size N.

ID	Model	IT	Koran	Medical	TED	Params (M)
(9)	Base (En-centric)	22.9	6.9	24.7	18.8	N/A
(10)	Finetuned	40.7	16.9	42.7	26.7	79
(11)	Single adapter per layer $(d = 1024)$	39.5	15.3	41.8	26.4	12.6
(12)	LA $(d = 1365)$	40.5	16.9	42.0	26.6	202
(13)	LA $(d = 2048)$	41.8	18.9	43.3	27.0	303
(14)	LA + dec. DA $(d = 1024)$	42.0	19.5	43.3	27.4	177
(15)	LA + enc. DA $(d = 1024)$	42.1	20.5	43.5	27.7	177
(16)	LA + enc & dec. DA $(d = 1024)$	42.5	21.2	43.6	27.8	202

Table 3: BLEU scores of various multilingual multi-domain adaptation strategies, i.e. training on all language directions from the 12 languages and all domains. LA = language adapters, DA = domain adapters. 'Params (M)' refers to the number of trainable parameters in millions. Note that unlike in Table 4 the LA here are not pre-trained on ParaCrawl; they are trained jointly with domain adapters.

adaptation on a subset of language pairs and evaluate on all languages. Table 4 shows results for cross-lingual transfer from the subset {en, fr, de, cs} in the Medical domain. We report decomposed BLEU scores for different categories of language-directions depending on whether the source/target language has in-domain parallel data. Appendix C has results in other domains and language subsets; we find similar trends to those reported here.

Off-target translation Directly training vanilla adapters on this subset **(20)** without using pre-trained language adapters results in strong performance in-domain (translating between {en, fr, de, cs}), but produces almost 100% off-target translations when translating into an out-of-domain target language. Using domain *and* language adapters in the naive way **(21)**, stacking them in the encoder and decoder, gives similar performance in-domain, but only slightly better out-of-domain performance.

Improving out-of-domain performance We show in Table 4 various strategies to improve performance on unseen language combinations. We treat fine-tuning the ParaCrawl language adapters (with no DA) as a strong baseline **(23)**, and note this method requires 8× the trainable parameters of decoder/encoder-only DA (and 24× when using back-translation due to training on all 12 languages). Efforts to 'decouple' domain and language improve unseen combinations. For example, using decoder-only domain adapters **(24)** leads to the best performance when translating from out-of-domain into in-domain languages, because the DA and target LA were both seen together at train time. And vice-versa, when translating from in-domain into out-of-domain languages, encoder-only domain adapters work well (although performance is still low).

Multi-domain models To study whether jointly training on multiple domains and languages would enable language adapters to be more 'domain-agnostic', we experiment with a setting where we jointly train on all language directions for IT, Koran and TED Talks domains and a subset of languages

ID	Model	All	Out→in	In→out	In→in	Out→out
(17)	Base (En-centric)	26.0	27.4	26.1	27.2	24.5
(18)	(**17**) + ParaCrawl LA	30.5	31.3	30	30.2	30.2
(19)	(**17**) + Finetune (all directions)	43.9	44.5	43.7	43.6	43.6
(20)	(**17**) + Domain adapters only	23.6	38.2	13.7 (12%)	45	13.8 (13%)
(21)	Freeze LA + enc. & dec. DA	27.3	37	20.5 (80%)	44.4	20.2 (83%)
(22)	Unfreeze LA	33.2	36.6	28.8	45.6	30.2
(23)	(**23**) + BT	35.4	33.3	38.4	45	31.9
(24)	Freeze LA + dec. DA	29.5	**41.0**	22.9	42.2	22.3
(25)	(**24**) + BT	**37.2**	38.5	36.7	41.2	**35.4**
(26)	Freeze LA + enc. DA	30.0	34.4	27.3	42.9	24.9
(27)	(**26**) + BT	33.0	35.3	35.1	42.0	27.1
(28)	(**21**) + DADrop	28.5	37.0	23.3 (89%)	43.1	21.7 (88%)
(29)	(**21**) + BT	34.0	36.7	35.9	43.5	27.8
(30)	(**21**) + BT + DADrop	35.2	37.3	36.8	42.5	30.5
(31)	Unfreeze LA + dec. DA	23.1	37.7	16.6 (24%)	43.4	11.3 (49%)
(32)	(**31**) + DADrop	31.4	37.1	24.2	**45.9**	28.0
(33)	(**31**) + DADrop + BT	35.9	34.2	**38.6**	44.0	32.8

Table 4: BLEU score of various models trained on the {en, fr, de, cs} subset of the Medical domain. LA = language adapters, DA = domain adapters. 'Out→in' is the average score when translating from an out-of-domain source language into {en, fr, de, cs}. 'In→out' corresponds to when the out-of-domain language is the target language. 'In→in' refers to average score when source and target are in the set {en, fr, de, cs}. 'Out→out' is the average score when both the source and target language are unseen during domain adaptation. We note percentage of **on-target** (correct language) translations in brackets, when it is less than 90% only.

for Medical. Such models improve out-of-domain performance with decoder-only (**24**) and encoder-only DA (**26**) by respectively 5.3 and 2.4 BLEU on average, and decrease off-target translation (see Appendix C; similar results hold for taking a subset of Koran data instead of Medical). However these scores are still worse than simply using pre-trained 'domain-agnostic' ParaCrawl LA (**18**) and we did not explore this method further.

Data augmentation Data augmentation with English-centric back-translation (BT) is the only technique that improves over pre-trained ParaCrawl LA for out-of-domain target languages. Decoder DA combined with BT (**25**) is the best performing model for out-of-domain, improving 5.2 BLEU on average over the ParaCrawl LA (**18**). Note that with BT, every language has been seen in combination with English model output, so 'out-of-domain' is closer to 'Out→in', where decoder-only DA also performs well. We report results for the other data augmentation methods (see Section 3.2) in Appendix C; these only improve over the ParaCrawl LA baseline in limited settings.

Other techniques We find that randomly dropping domain adapters (DADrop; see Section 3.2) improves zero-shot performance and slightly decreases in-domain performance (**28**). We believe this is caused by a less tight coupling of language and domain information. This technique helps when fine-tuning both LA and DA (**32**), or with encoder + decoder DA (**28**); see Appendix C for more results. We find that modular-style stacking outperforms MAD-X style (see Section 3.1), although the extent of this varies by which subset (in-domain, etc.) of languages we consider (see Appendix C).

Several of our models fine-tune only a single adapter per-layer and use frozen LA. Such models can easily be 'mixed-and-matched' by activating a particular adapter for a particular language pair. For example we could activate model (**24**) on 'Out→in' (out-of-domain source, in-domain target) data, model (**29**) on in-domain data and model (**25**) otherwise. Such models could easily be extended to new domains by training more adapters.

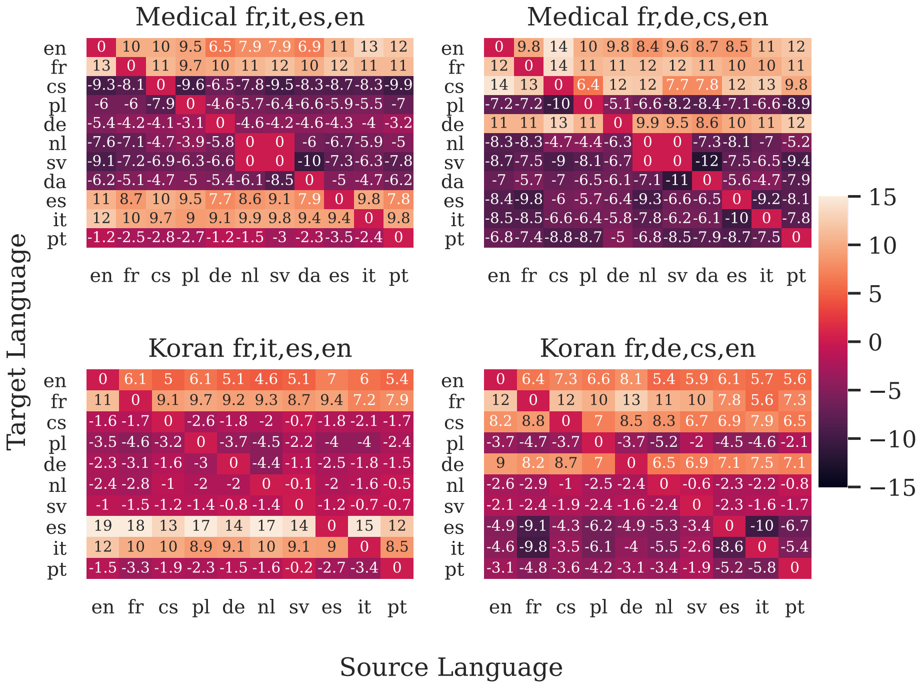

Figure 2: Analysis of cross-lingual transfer for the decoder-only domain adapter (with pre-trained ParaCrawl language adapters) fine-tuned on a subset of language for the Medical (top) or Koran (bottom) domains. Models are either trained on romance languages (fr,it,es,en; left) or a mix of language families (fr,de,cs,en; right). All numbers are BLEU score improvement over a model with ParaCrawl language adapters that have not been fine-tuned on in-domain data. Other than en & fr, language families are grouped together. Best viewed in .pdf form.

Cross-lingual transfer analysis We conduct a preliminary analysis of whether language diversity is important for cross-lingual transfer. We compare models trained on a mix of language families (fr, de, cs, en) and mostly romance languages (fr, it, es, en) to test whether diversity of languages in our in-domain training set improved transfer. Figure 2 shows how the performance changes for each language pair. When translating from out-of-domain source languages into in-domain target languages, training on diverse languages gives better performance (compare *en* and *fr* rows in Figure 2).

However, for out-of-domain source and target languages (*da, nl, pl, pt, sv*) the romance languages performed roughly the same or better. It is not clear why this is the case, but perhaps the DA are more language-agnostic when trained on similar languages as they devote less capacity to language information. When using back-translation or when training multi-domain models (which see all language directions for three domains as explained previously), training with multiple families of languages outperforms romance-only training for out-of-domain languages as expected. We summarise these results in Appendix C.

6 Conclusion

This work studies composing language and domain adapter modules in the context of NMT. We find that while adapters for encoder architectures like BERT can be safely composed, this is not true for NMT adapters: domain adapters struggle to generate into languages they were not trained on, even though the original model they are inserted in was trained on those languages (but not with in-domain data). Naive fine-tuning, or stacking language adapters with domain adapters at every layer results in low cross-lingual generalisation.

We can improve performance when translating an out-of-domain source language into an in-domain target language by using domain adapters in the decoder only. Data augmentation with back-translation improves a similar model when both source and target are out-of-domain. Overall decoupling domain and language information is required for strong cross-lingual generalisation. When used carefully we believe using adapters for multilingual domain adaptation represent a convenient and effective method with reasonable cross-lingual generalisation and are easily extensible to new domains.

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 (Volume 1: Long Papers)*. 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.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

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.

Alexandre Berard, Ioan Calapodescu, Marc Dymetman, Claude Roux, Jean-Luc Meunier, and Vassilina Nikoulina. 2019a. Machine translation of restaurant reviews: New corpus for domain adaptation and robustness. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 168–176, Hong Kong. Association for Computational Linguistics.

Alexandre Berard, Ioan Calapodescu, and Claude Roux. 2019b. Naver Labs Europe's systems for the WMT19 machine translation robustness task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 526–532, Florence, Italy. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A Benchmark for Information-Seeking Question Answering in Typologically Diverse Languages. In *Transactions of the Association of 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, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of EMNLP 2018*, pages 2475–2485.

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.

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.

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. 2020a. Beyond english-centric multilingual machine translation.

Angela Fan, Edouard Grave, and Armand Joulin. 2020b. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations*.

Markus Freitag and Orhan Firat. 2020. Complete multilingual neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 550–560, Online. Association for Computational Linguistics.

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 *International Conference on Machine Learning*, volume 97, pages 2790–2799, Long Beach, California, USA.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *CoRR*, abs/2003.11080.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. A simple baseline to semi-supervised domain adaptation for machine translation.

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.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain Control for Neural Machine Translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria. INCOMA Ltd.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations (ICLR)*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. 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.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *International Workshop on Spoken Language Translation*.

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.

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.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. 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.

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.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. 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, Belgium, Brussels. 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*, pages 506–516, Long Beach, California, USA.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2020. Adapterdrop: On the efficiency of adapters in transformers. *CoRR*, abs/2010.11918.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the*

54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Emmanouil Stergiadis, Satendra Kumar, Fedor Kovalev, and Pavel Levin. 2021. Multi-domain adaptation in neural machine translation through multidimensional tagging.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

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.

A Data and Hyper-parameters

We share embeddings between encoder and decoder. We use the Adam optimizer (Kingma and Ba, 2014) with an inverse square root learning rate schedule for pre-training, and a fixed learning rate schedule for training adapters. We speed up training with 16 bit floating point arithmetic. We use label smoothing 0.1 and dropout 0.1. We train for either 20 epochs or 1 million updates, whichever corresponds to the smallest number of training updates. We use early stopping, checking performance after each epoch or every 100,000 training steps, and use average validation negative-log-likelihood on all of the training data (but not out-of-domain language data) as our criteria for choosing the best model. We otherwise use default Fairseq (Ott et al., 2019) parameters. We train all models on a single Nvidia V100 GPU, and training takes between 8 and 36 hours depending on dataset size.

In order to create validation and test splits that had no overlap with training data in any language, we first set aside a number of English sentences. Then we aligned all language pairs to these sentences, i.e. the German to French test set is composed of German and French sentences that share the same English sentence. Finally we remove all sentences in any language from the train splits of all parallel data if those sentences are aligned with any English sentences in the subset we set aside for validation/test splits.

B Additional Results for Bilingual Domain Adaptation

Table 6 presents results of bilingual domain adaption explored in the main paper but with smaller adapter bottleneck dimension. The same trends emerge: encoder-only adapters perform better, and the last three layers of the encoder are better than the first three. The last three encoder layers also perform better than the first three for a multilingual model, see Table 9 models **(93)** and **(94)**. Interestingly the multilingual last three encoder layer DA model is roughly halfway between encoder-only and decoder-only on Out→in and In→out performance, suggesting it might be a useful compromise between the two.

C Additional Results for Cross-lingual Transfer

Table 5 compares models trained on a mix of language families (fr, de, cs, en) and mostly romance languages (fr, it, es, en) to test whether diversity of languages in our in-domain training set improved transfer. Positive numbers in this table indicate diversity of training languages improves performance. As noted in the main paper, diversity helps for translating out-of-domain languages into in-domain. We have unclear results for when both source and target are out-of-domain; it seems when using back-translation (BT), i.e. when all languages have been seen (albeit with artificial English parallel data) diversity helps, but without BT it mostly hurts performance. We speculate that training on mostly romance languages means the domain adapter encodes less 'language information', but leave further exploration to future work.

Model	Out→ {en,fr}	Out→Out
Koran		
LA + Dec. DA	0.6	-0.9
LA + Dec. DA	0.3	-0.3
Unfr. LA + Dec. DA	0.1	-0.4
LA + Enc & Dec. DA	1.3	-1.3
Koran + BT		
LA + Dec. DA	0.4	0.2
LA + Dec. DA	1.9	0.9
Unfr. LA + Dec. DA	0.3	0
LA + Enc & Dec. DA	0.7	0.3
Medical		
LA + Dec. DA	0.8	-2.4
LA + Dec. DA	3.2	0.2
Unfr. LA + Dec. DA	0.2	0.1
LA + Enc & Dec. DA	3.2	-1.1
Medical + BT		
LA + Dec. DA	0.4	2.9
LA + Dec. DA	1.3	1.6
Unfr. LA + Dec. DA	0.5	3.1
LA + Enc & Dec. DA	0.5	1.5

Table 5: Difference in average BLEU score between models trained on a diverse subset of languages and models trained on mostly romance languages. Data source is noted in bold. Refer to the main paper for model definitions. Out→ {en,fr} corresponds to translation from an out-of-domain source language into {en,fr}. 'Out→Out' is the average score when both the source and target language are unseen during domain adaptation (choosing languages unseen by either subset).

We present additional results for the setting discussed in Section 5.2 of the main paper in Table 7

ID	Model	IT	Medical	Koran	Subtitles	Law
(34)	No finetuning	35.3	14.8	38.1	26.8	42.4
(35)	Finetuned	43.8	22.7	53	30.9	57.9
(36)	Enc. + dec. adapters (d=64)	40	18.7	47.3	29.4	51.5
(37)	Dec. adapters (d=128)	39	17.5	46	28.8	50.6
(38)	Enc. adapters (d=128)	40	18.9	47.3	29.2	51.5
(39)	Last 3 encoder layers only (d=256)	40	19	47.3	29	51.1
(40)	First 3 encoder layers only (d=256)	39.5	18	46	28.8	49.5

Table 6: BLEU scores of various domain adaptation strategies for a German → English bilingual model. $(d = N)$ refers to adapters with a bottleneck dimension of size N.

(Koran domain), Table 8 (Koran results for the romance language subset), and Table 9 (additional Medical results). For the Koran domain, we see similar trends with decoder-only domain adapters (DA) performing best on out-of-domain source to in-domain target languages, and vice versa for encoder-only DA. We briefly experiment with denoising objectives, where we simply copy target data in out-of-domain languages to the source side (and optionally add 'noise' to the source side, e.g. swap tokens or mask tokens (Lewis et al., 2020)). Although we got reasonable improvements (models **(56)** and **(51)**) for out-of-domain target languages, we were mostly unable to improve over the pre-trained ParaCrawl LA, and so concentrate on back-translation.

We experiment with a setting where we jointly train on all language directions for IT, Koran and TED Talks domains and a subset of languages for Medical, and similarly with only a subset of Koran (models **(47)**, **(48)** etc.). These models stack language and domain adapters. Such models don't require any pre-trained LA, and improve out-of-domain performance and decrease off-target translation compared to freezing ParaCrawl LA and training DA. However these scores are still worse than simply using pre-trained 'domain-agnostic' ParaCrawl LA **(43)**.

ID	Model	All	Out→in	In→out	In→in	Out→Out
(41)	Base (En-centric)	7	7.5	7	7.7	6.4
(42)	Finetune (all directions)	23.8	19.9	26.8	20.6	25.7
(43)	**(41)** + ParaCrawl LA	9	9.2	9	9	8.7
(44)	Domain adapters only	6.9	13.5	0.4	23.0	0.3
(45)	Freeze LA + enc. & dec. DA	8.6	13.5	4.1	21.7	3.1
(46)	Unfreeze LA	11.8	12.9	8.6	23.5	8.7
(47)	Multi-domain dec. DA	9.9	12.9	7.8	17.0	6.4
(48)	Multi-domain enc. DA	10.6	13.1	9.0	17.6	7.1
(49)	Multi-domain enc. & dec. DA	10.3	14.0	7.7	18.2	6.2
(50)	Freeze LA + dec. DA	9.7	**16.2**	5.2	18.2	4.8
(51)	**(50)** + Mono data	11.3	14.4	8.7	16.2	8.6
(52)	**(50)** + MAD-X style	9	15.4	4.5	17.1	4.3
(53)	**(50)** + BT	13.7	14.4	13.1	17.5	**12.1**
(54)	**(50)** + BT + DADrop	**13.8**	14.7	13.2	17.7	**12.1**
(55)	Freeze LA + enc. DA	9.6	10.9	8.3	20.3	5.4
(56)	**(55)** + Mono data	10.6	10.5	10	17.3	8.3
(57)	**(55)** + MAD-X style	7.5	11.0	3.8	19.4	2.8
(58)	**(55)** + BT	12	12.1	12.7	19.5	8.3
(59)	**(55)** + BT + DADrop	12.3	12.2	13.5	18.9	8.8
(60)	Enc. & dec. DA + BT	13.3	14	13.9	20.8	9.3
(61)	**(60)** + DADrop	**13.8**	14.2	**14.8**	20.9	9.8
(62)	Unfreeze LA + dec. DA	7.6	13.6	0.51	**24.6**	1.6
(63)	**(62)** + DADrop	11.5	13.5	6.8	24.5	8.4

Table 7: BLEU score of various models trained on the {en, fr, de, cs} subset of the Koran domain. LA = language adapters, DA = domain adapters. 'Out→in' is the average score when translating from an out-of-domain source language into {en, fr, de, cs}. 'In→out' corresponds to when the out-of-domain language is the target language. 'In→in' refers to average score when source and target are in the set {en, fr, de, cs}. 'Out→Out' is the average score when both the source and target language are unseen during domain adaptation. 'Mono data' refers to adding copied monolingual data for out-of-domain languages, and additionally multiparallel ParaCrawl data in small amounts.

ID	Model	All	Out→in	In→out	In→in	Out→Out
(64)	Freeze LA + enc. & dec. DA	10.0	13.9	5.7	26.7	3.6
(65)	(**64**) + DADrop	10.0	13.8	5.8	26.7	3.6
(66)	Unfreeze LA	13	16.5	7.3	31.6	7
(67)	(**66**) + BT	14.4	12.9	**13.4**	**35.0**	8.1
(68)	Freeze LA + dec. DA	11.6	**19.4**	5.6	24.1	5.1
(69)	(**68**) + BT	14.2	17.7	11.6	23.5	**9.7**
(70)	Freeze LA + enc. DA	9.9	11.8	7.1	26.1	4.2
(71)	(**70**) + BT	12.1	13.4	10.9	26.2	6.3
(72)	Unfreeze LA + dec. DA	11.9	17.7	3.7	32.9	5.6
(73)	(**72**) + DADrop	13.3	17.5	7.2	32.1	7.2
(74)	(**72**) + DADrop + BT	**14.5**	13.6	**13.4**	33.6	8.6
(75)	Enc. & dec. DA + BT	13.3	15.6	11.9	26.9	7.2
(76)	(**75**) + DADrop	13.8	16.3	12.2	27.7	7.4

Table 8: BLEU score of various models trained on the mostly romance language {en, fr, it, es} subset of the Koran domain. LA = language adapters, DA = domain adapters. ‘Out→in’ is the average score when translating from an out-of-domain source language into {en, fr, it, es}. ‘In→out’ corresponds to when the out-of-domain language is the target language. ‘In→in’ refers to average score when source and target are in the set {en, fr, it, es}. ‘Out→Out’ is the average score when both the source and target language are unseen during domain adaptation.

ID	Model	All	Out→in	In→out	In-domain	Out→Out
(77)	Base (En-centric)	26.0	27.4	26.1	27.2	24.5
(78)	**(77)** + ParaCrawl LA	30.5	31.3	30	30.2	30.2
(79)	**(77)** + Finetune (all directions)	43.9	44.5	43.7	43.6	43.6
(80)	**(77)** + Domain adapters only	23.6	38.2	13.7 (12%)	45	13.8 (13%)
(81)	Freeze LA + enc. & dec. DA	27.3	37	20.5 (80%)	44.4	20.2 (83%)
(82)	Unfreeze LA	33.2	36.6	28.8	45.6	30.2
(83)	**(83)** + BT	35.4	33.3	**38.4**	45	31.9
(84)	Multi-domain dec. DA	31.9	37.3	28.6	42	27.5
(85)	Multi-domain enc. DA	32.3	38.4	28.9	42.4	27.3
(86)	Multi-domain enc. & dec. DA	31.3	38.8	26.8	42.4	25.9
(87)	Freeze LA + dec. DA	29.5	**41.0**	22.9	42.2	22.3
(88)	**(87)** + BT	**37.2**	38.5	36.7	41.2	**35.4**
(89)	**(87)** + BT + DADrop	36.9	37.8	36.6	40.5	35.3
(90)	Freeze LA + enc. DA	30.0	34.4	27.3	42.9	24.9
(91)	**(90)** + BT	33.0	35.3	35.1	42.0	27.1
(92)	**(90)** + BT + DADrop	34.1	35.7	35.8	41.4	29.6
(93)	Freeze LA + enc. first 3 layers DA	24	31.5	18.2	41.8	17.6
(94)	Freeze LA + enc. last 3 layers DA	29.8	37.3	24.8	42.7	24.2
(95)	**(81)** + DADrop	28.5	37.0	23.3 (89%)	43.1	21.7 (88%)
(96)	**(81)** + BT	34.0	36.7	35.9	43.5	27.8
(97)	**(81)** + BT + DADrop	35.2	37.3	36.8	42.5	30.5
(98)	**(81)** + BT + MAD-X style	32.3	37	33.6	43.5	24.8
(99)	Unfreeze LA + dec. DA	23.1	37.7	16.6 (24%)	43.4	11.3 (49%)
(100)	**(99)** + DADrop	31.4	37.1	24.2	**45.9**	28.0
(101)	**(99)** + DADrop + BT	35.9	34.2	38.6	44.0	32.8

Table 9: BLEU score of various models trained on the {en, fr, de, cs} subset of the Medical domain. Some results are also included in the main paper.