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

Asa Cooper Stickland[*]   **Alexandre Bérard**   **Vassilina Nikoulina**
University of Edinburgh   NAVER LABS Europe
a.cooper.stickland@ed.ac.uk   first.last@naverlabs.com

Abstract

Adapter layers are lightweight, learnable units inserted between transformer layers. Recent work explores using such layers for neural machine translation (NMT), to adapt pre-trained models to new domains or language pairs, training only a small set of parameters for each new setting (language pair or domain). In this work we study the compositionality of language and domain adapters in the context of Machine Translation. We aim to study, 1) parameter-efficient adaptation to multiple domains and languages simultaneously (full-resource scenario) and 2) cross-lingual transfer in domains where parallel data is unavailable for certain language pairs (partial-resource scenario). We find that in the partial resource scenario a naive combination of domain-specific and language-specific adapters often results in 'catastrophic forgetting' of the missing languages. 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. Supplementary material is available at https://tinyurl.com/r66stbxj.

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 adapting multilingual systems, in-domain data for most language pairs might not exist. We would like to be able to leverage data in a subset of language pairs to transfer domain knowledge to other languages.

Straightforward methods of domain adaptation include fine-tuning (Freitag and Al-Onaizan, 2016) or usage of domain tags (Kobus et al., 2017; Britz et al., 2017) for different domains. For these methods each new domain request would require re-training the whole model, which is a costly procedure. And naive training on a subset of languages typically reduces performance on all other languages (Garcia et al., 2021), a phenomenon known as 'catastrophic forgetting' (McCloskey and Cohen, 1989).

An alternative technique for adapting such models to new language-pairs or domains are 'adapter layers' (Bapna and Firat, 2019), lightweight, learnable units inserted between transformer layers. A previously trained large multilingual model can be adapted to each language-pair by learning only these small units, and keeping the rest of the model frozen. This procedure also allows for the incremental adding of new language pairs and/or domains to the pre-trained model, reducing the cost of adaptation. Previous studies have shown it is possible to combine language-specific (as opposed to language-pair specific) adapters (Philip et al., 2020), or language and task adapters (Pfeiffer et al., 2020) trained independently, enabling zero-shot compositions of adapters. Our ultimate goal is, for ease of deployment and storage, a single model that can handle all languages and domains. In this work we analyse how to combine *language adapters* with *domain adapters* in multilingual NMT, and study to what extent the domain knowledge can be transferred across languages.

First, we show it is hard to decouple language

---

knowledge from domain knowledge when fine-tuning 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. We additionally find combinations of domain and language adapters not seen 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 domain adaptation capacity in the multi-lingual, multi-domain setting. We find that encoder-only adapters can be just as effective as default adapters added in every layer, and that composing domain adapters with language adapters outperforms language adapters alone, although fine-tuning with domain tags performs better for most domains.

- We improve the cross-lingual transfer of domain knowledge for adapters. 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). Although effective, it is expensive to create back-translated data, especially when one needs to cover multiple language pairs. Multi-domain models can be trained with domain tags (Kobus et al., 2017; Britz et al., 2017; Berard et al., 2019a; Stergiadis et al., 2021) that can encode domain-specific information. However, domain tags do not allow *incrementally* adding new domains to a model: each new domain adaptation requires retraining the full model (as opposed to adapter layers that can be trained independently for each language/domain). There are a number of works (Jiang et al., 2020; Britz et al., 2017; Dabre et al., 2020) trying to explicitly decouple domain-specific representations from domain independent representations in bilingual settings. In our work we try to decouple language and domain specific representations through adapter layers.

Adapter layers Bapna and Firat (2019) introduce adapter layers for NMT as a lightweight alternative to fine-tuning. 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. Philip et al. (2020) train adapters for each *language* instead of *language-pair* and show that composing such adapters improves zero-shot translation in English-centric settings, and can adapt a model to all language directions in a scalable way. Pfeiffer et al. (2020, 2021) study adapter layers for pre-trained language models evaluated on NLU tasks. They show it is possible to compose language and task adapters. Combining language adapters trained with a masked language modelling objective for language x and task adapters trained on a classification task in language y can transfer to classification in language x. We have 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.

To the best of our knowledge none of the works above study composing language and domain adapters for generation tasks (such as translation) which is the goal of this work.

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 $\text{FFN}(\mathbf{h}) = W_{\text{up}} f(W_{\text{down}} \mathbf{h})$, 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:

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

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

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

For all models without any stacking we obtain layer output as in Eq. 2 but replace $\text{LA}(\cdot)$ with the identity operation.

Pfeiffer et al. (2020) use a different formulation that empirically performed well for them, but that in initial experiments produced worse results in our setting. We list the corresponding equations and results in Appendix B and Appendix D.

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. We also find empirically that decoder-only adapters work well with back-translation, perhaps because they can 'ignore' the noisy synthetic source-side data.

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).

---

[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

4 Experimental Settings

4.1 Data

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 due to expanding the number of languages and wanting a consistent pipeline for obtaining the data.

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. †: missing nb & da, ‡: missing nb.

4.2 Models

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 (Vaswani et al., 2017) 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). On the recommendation of Marie et al. (2021) we additionally report chrF (Popović, 2015) calculated using Sacrebleu[6] for most models in the Appendix. 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. We report the results of models finetuned on both all the domains simultaneously, or each domain separately, with access to in-domain data available for all the languages. Both serve as a potential upper bound for cross-lingual transfer.

We train the same model (i.e. with access to all languages) with domain tags: one special token per domain prepended to each source sequence (Kobus et al., 2017). We also measure the cross-lingual transfer ability of domain tags, by training a model with domain tags on all 4 domains but with in-domain data in only 4 languages (fr, de, cs and en). Because the latter model exhibits catastrophic forgetting issues in the other languages, we also train the same model with ParaCrawl data in all language directions (with a "paracrawl" domain tag). ParaCrawl line pairs are sampled with probability 0.5. More training hyper-parameters are given in Appendix A.

4.3 Our model pipelines

We perform two series of experiments.

Multilingual multi-domain models. Firstly, we experiment with different ways of multi-domain adaptation of multilingual models. We adapt the English-centric ParaCrawl pre-trained model to four domains (Koran, Medical, IT and TED) and every language direction simultaneously. We test models with language adapters, language + domain adapters, and domain tags. There is no cross-lingual domain transfer needed[7] since all language

---

[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.

[6]Signature: chrF2+numchars.6+space.false+version.1.5.1

[7]There is obviously cross-lingual domain transfer that may take place when all the domains are trained jointly, but we do
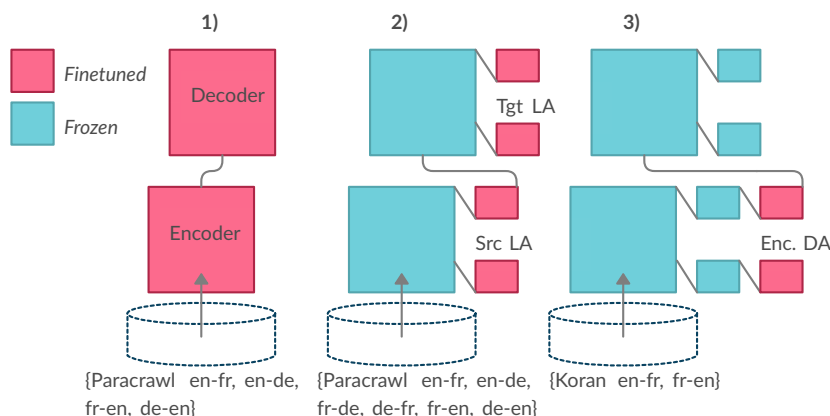
Figure 1: Toy diagram showing one of our proposed pipelines for training language and domain adapters, on a example subset of languages: {en,fr,de}, with 'domain-agnostic' data from ParaCrawl and specialised data from the Koran. Red indicates a fine-tuned model component, blue indicates a frozen component. 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.

directions are included in the training data. Results for this scenario are reported in Section 5.1.

Cross-lingual domain transfer. In the second experiment we try to decouple the notion of domain from language via analysing the zero-shot composition of domain and language adapters. 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 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 (11) (which has not seen any in-domain data) on the English-aligned training data for 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 the results of experiments testing the domain adaptation capacity of various models, assuming access to data for all language pairs. In Section 5.2 we analyse domain transfer across languages with adapters and other methods. We first demonstrate problems with cross-lingual generalisation during domain adaption for 'naive' methods, and then propose potential solutions. Note we concentrate on the medical domain and a particular language subset for convenience. Appendix D has results in

---

not explicitly study this in the first experiment.

ID	Model	IT	Koran	Medical	TED	Params (M)
(1)	Base (En-centric)	23.2	7.0	25.7	19.0	N/A
(2)	Finetuned	40.8	16.0	42.7	26.6	79
(3)	Finetuned + domain tags	**43.6**	**20.3**	**46.0**	27.2	79
(4)	Single adapter per layer ($d = 1024$)	39.6	14.7	41.8	26.2	12.6
(5)	LA ($d = 1365$)	42.0	17.6	43.7	26.8	202
(6)	LA ($d = 2048$)	42.2	18.1	43.8	26.9	303
(7)	LA + dec. DA ($d = 1024$)	42.1	18.5	43.6	27.1	177
(8)	LA + enc. DA ($d = 1024$)	42.3	19.3	43.8	27.5	177
(9)	LA + enc & dec. DA ($d = 1024$)	42.7	20.1	44.0	**27.7**	202

Table 2: BLEU scores averaged across all the language-directions for 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 3 the LA here are not pre-trained on ParaCrawl; they are trained jointly with domain adapters.

ID	Model	All	In→in	Out→in	In→out	Out→out
	Oracles					
(10)	Finetune (all langs)	44.3	43.9	44.8	44.2	44.2
(3)	FT (all langs & domains) + dom. tags	46.0	45.3	46.3	45.9	46.0
	Baselines					
(1)	Base (En-centric)	25.7	27.0	27.2	25.9	24.3
(11)	**(1)** + ParaCrawl LA	30.2	29.6	30.8	30.0	30.0
	Straightforward Methods					
(12)	**(1)** + Domain adapters only	23.0	44.7	37.8	13.4 (7%)	13.4 (11%)
(13)	Freeze LA + enc. & dec. DA	26.9	44.0	36.7	20.1 (71%)	19.9 (76%)
(14)	Freeze LA + enc. DA	29.6	42.6	34.0	27.0 (89%)	24.6 (88%)
(15)	Freeze LA + dec. DA	29.0	41.7	**40.7**	22.5 (77%)	22.0 (77%)
(16)	FT (all domains) + dom. tags	15.6	**46.8**	13.2 (55%)	12.0 (1%)	10.7 (2%)
	Improving Off-target Translation					
(17)	**(16)** + ParaCrawl	34.7	42.2	39.6	32.4	31.0
(18)	**(13)** + BT	33.9	43.2	36.8	35.9	28.0 (85%)
(19)	**(14)** + BT	32.5	41.8	35.0	34.7	26.8 (83%)
(20)	**(15)** + BT	**36.9**	40.9	38.2	36.4	**35.1**
(21)	**(13)** + DADrop	28.0	42.6	36.7	22.9 (82%)	21.5 (82%)
(22)	**(13)** + BT + DADrop	34.8	42.2	37.0	36.5	30.2
(23)	Unfreeze LA + dec. DA	14.3	45.8	36.6	0.0 (1%)	0.0 (2%)
(24)	**(23)** + DADrop	31.1	45.5	36.9	23.9 (82%)	27.8
(25)	**(23)** + DADrop + BT	35.2	44.5	33.4	**38.2**	31.8

Table 3: BLEU score of various models trained on the {en, fr, de, cs} subset of the Medical domain, except 'Oracle' models trained on all language pairs. 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.

other domains and language subsets, and also chrF (Popović, 2015) scores; we find similar trends to those reported in Section 5.2.

5.1 Multilingual multi-domain models

Table 2 reports the results from the 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, and so we are testing the capacity of various models for domain adaptation, rather than cross-lingual transfer. Models are compared against a baseline (1) not trained on in-domain data.

We report the results for naive fine-tuning on the concatenation of in-domain parallel datasets for all the languages and all the domains (2). On all domains we improve on these results by fine-tuning with domain tags (3) (a similar result to Jiang et al. (2020) in the bilingual setting). Fine-tuning with domain tags (3) outperforms the model with stacked adapters (9). A fine-grained comparison of these models is in Figure 5 in the Appendix. For the IT and Medical domains the model with tags (3) is clearly better for all language directions. For the lowest resource domains, Koran and TED, most of the differences are not statistically significant, except for English-centric language pairs for TED, where the adapter model (9) is better. Exploring the combination of domain tags and adapters could be an interesting future research direction.

Stacking domain and language adapters (9) results in better performance than a model with the same parameter budget devoted to language adapters only (5). We believe this is because it allows the model to (partially) decouple domain from language-specific information, and better exploit the allocated parameter budget. Even a higher capacity language adapter model (6) does not perform as well.

We also note that usage of encoder-only domain adapters (8) outperforms the decoder-only domain adapter model (7). This is perhaps because the encoder representations influence the whole model (it is directly connected to the decoder at all layers with encoder-decoder attention) as opposed to the the decoder adapters that only impact decoder representations. We find a similar trend in bilingual domain adaptation, see Appendix C.

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. This implies devoting more capacity to encoder adapters would achieve similar performance and faster inference (more details in Appendix C).

5.2 Cross-lingual Domain Transfer

To study the capacity of our models to transfer domain knowledge across languages, we perform domain adaptation using parallel datasets for a *subset* of language pairs, and evaluate on the test sets available for *all* language pairs. In this section we report the results for adaptation to the *medical* domain using the subset of all the language directions including {en, fr, de, cs} languages (Table 3). We refer to these languages as *in-domain* languages, and *out-of-domain* languages would include all the other languages, {de, nl, sv, es, it, pt, pl } (referred to as *In* and *Out* respectively in Table 3).

We report BLEU scores averaged across test sets of different categories of language-directions depending on whether the source/target language was observed during the domain adaptation training: In→in for language pairs observed during DA, Out→out for fully zero-shot DA performance, and In→out, Out→in for translation directions combining *in-domain* and *out-of-domain* languages.

First, we report the results for *Oracle* models providing an upper bound for the scores models could achieve with access to in-domain data for *all* the languages: model (10) was fine-tuned on *medical* data for all the language directions[8], and a model with domain tags (3) discussed in section 5.1.

Baseline models include the default multilingual English-centric model (1), as well as model (11) with language adapters trained on multi-parallel ParaCrawl data. Comparing against this baseline shows us improvements from domain-specific (rather than language-specific) information.

Straightforward Methods We train several 'straightforward' adapter models for the subset of *in-domain* languages on the top of the baseline model, one with no language adapters, model (12), and model (13) with domain *and* language adapters (where language adapters are frozen), stacking them in the encoder and decoder.

---

[8]This is different from the model (3) which was fine-tuned on *all* the domains and all the language directions.

Both of these models achieve good scores when translating into *in-domain* languages (the In→in and Out→in categories), on par or better than *Oracle* scores and much higher than the baselines. On the other hand they suffer from significant drops in performance when translating into *out-of-domain* languages (the In→out and Out→out categories).

The model (**16**) trained with tags on a subset of *in-domain* languages suffers from the same low performance translating into *out-of-domain* languages and additionally has low performance with out-of-domain *source* languages.

Looking closer at the translations of the above models, we see that many translations are either generated in English, copy the source language, or mix words between English and the true target language; see Table 4 in the Appendix for illustrative examples. We refer to this phenomenon as "off-target" translation. We report the percentage of translations generated in the correct target language in Table 3 when it is lower than 90% [9].

We believe this phenomenon is partly due to decoder domain adapters having never been exposed to *out-of-domain* language generation. Encoder domain adapters seem to be less sensitive to composition with new language adapters (as observed by Pfeiffer et al. (2020) for NLU tasks, and Table 3 in the Out→In column).

To investigate this, we train models (**14**) and (**15**) with encoder-only and decoder-only adapters. Figure 2 compares the performances of these models as well as model (**13**) trained with encoder and decoder domain adapters, (**14**), (**15**) against the baseline model (**11**). The decoder-only model (**15**) can better translate *from* out-of-domain languages and the encoder-only model (**14**) slightly improves for translations *into* out-of-domain languages. However the problem of off-target translation persists for both models and neither improves over ParaCrawl LA (**11**). Therefore, we conclude that a straightforward combination of domain and language adapters leads to catastrophic forgetting both in the encoder and the decoder, but the encoder is less important for this effect.

Effect of data augmentation   We train models (**17**),(**18**) (**19**), (**20**) with additional data (either a

---

[9]This percentage is computed against the reference translations that were correctly tagged by 'langdetect', a Python language identifier (https://pypi.org/project/langdetect/). This is to exclude very short and numerical examples which can be quite frequent in some domains.

portion of ParaCrawl data, or back-translation of in-domain data) to alleviate potential forgetting of representations for *out-of-domain* languages. All of these models improve the translation quality into *out-of-domain* languages. The model with tags (**17**) reaches competitive results and can be considered as a strong baseline.

For models with back-translation data, the decoder-only adapter (**20**) model outperforms the encoder-only adapter (**19**) model on out-of-domain target languages (as opposed to the case without BT) and has the strongest results overall on translating into out-of-domain languages. While the BT models are trained on exactly the same data, this effect is possibly due the encoder adapters being more influenced by potentially noisy synthetic source-side data, whereas decoder adapters are more influenced by clean reference translations. The decoder-only BT model (**20**) improves over the baseline for all the language directions except for translation into English; see Figure 3.

We report results for the other data augmentation methods (see Section 3.2) in Appendix D; these only improve over the ParaCrawl LA baseline in limited settings.

Domain adapter dropout   Models (**21**) and (**22**) trained with dropping domain adapters (DADrop; see Section 3.2) also allow to reduce catastrophic forgetting, although only combining DADrop with Data Augmentation (model (**22**)) allows to solve the problem of off-target translation. We also note slight decreases in *in-domain* performance for those models, perhaps due to underfitting.

Increasing adaptation capacity   When naively increasing model capacity by unfreezing LA stacked with decoder DA (**23**), the model seems to mostly devote this capacity to In→in category, and suffers on other language pair groups. This trend seems to be similar to the un-augmented model with tags (**16**). However, once regularized with DADrop (**24**), and augmented with back-translation (**25**) it reaches very competitive results.

Figure 4 shows fine-grained results for different models with DADrop, back-translation and unfrozen LA. Back-translation improves performance on the Out→out and In→out groups, but decreases performance on the Out→in group. Finally, unfreezing language adapters decreases the performance on Out→in but improves on the Out→out group.
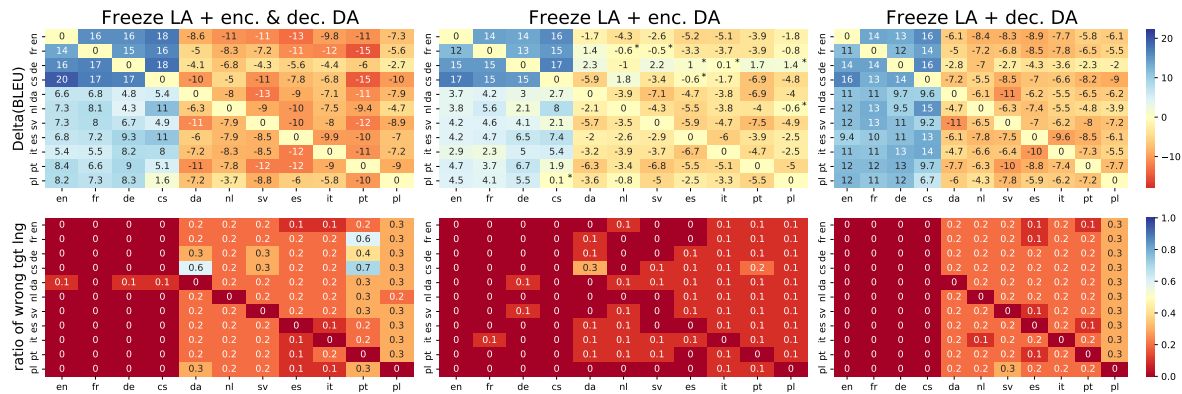
Figure 2: Comparing models with encoder-decoder adapters, encoder-only adapters and decoder-only adapters. x-axis shows the target language and y-axis shows the source language. Languages are grouped so the in-domain languages are in the top left corner. Top: Difference in BLEU compared to the baseline (11) (negative scores indicate a decrease w.r.t. the baseline, "*" indicates *not* statistically significant). Bottom: proportion translating into the wrong target language. Best viewed in .pdf form.
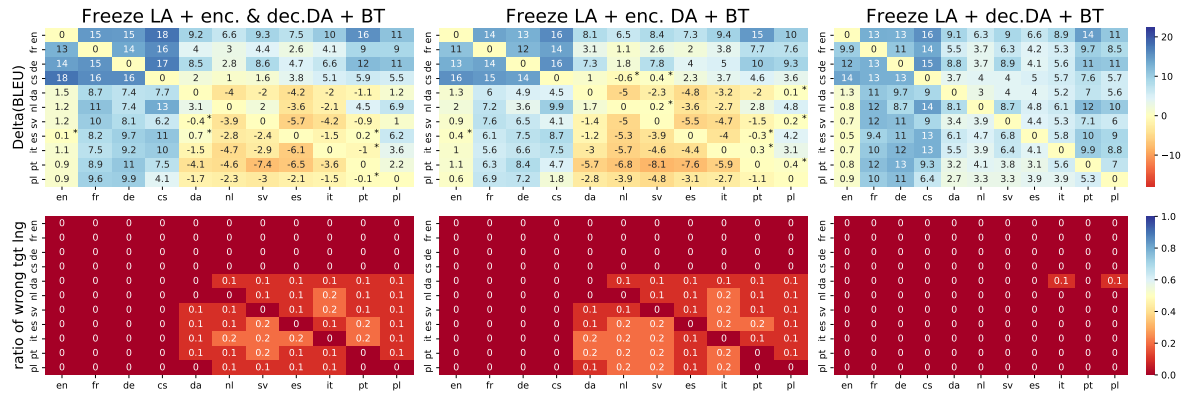

Figure 3: Comparing adapter models trained with back-translation. Top: Difference in BLEU compared to the baseline (11) ("*" indicates *not* statistically significant). Bottom: proportion translating into the wrong target language. See Figure 2 for more details.

Adapters vs. tags As mentioned previously, model (17) with tags augmenteed with ParaCrawl reaches competitive scores overall. Note that this model was trained on a concatenation of *all* the domains, unlike the models with adapters which were trained *only* on the medical domain. Therefore it has been exposed to more data overall. On the other hand, several of our models fine-tune only a single adapter per-layer and use frozen LA. Thus, encoder-only or decoder-only models only require 6.3 million tunable parameters, compared to 79 million for tag-based models. Additionally adapter models can easily be 'mixed-and-matched' by activating a particular adapter for a particular language pair. For example we could activate model (15) on 'Out→in' (out-of-domain source, in-domain target) data, model (18) on in-domain data and model (20) otherwise. Such models could easily be extended to

new domains by training more adapters, in contrast to tag-based models which update all parameters for each domain adaptation request.

6 Conclusion

In this work we studied multilingual domain adaptation both in the full resource setting where in-domain parallel data is available for all the language pairs, as well as the partial resource setting, where in-domain data is only available for a small set of languages.

In particular, we study how to better compose 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 learnt in the partial resource scenario struggle to generate into languages they were not trained
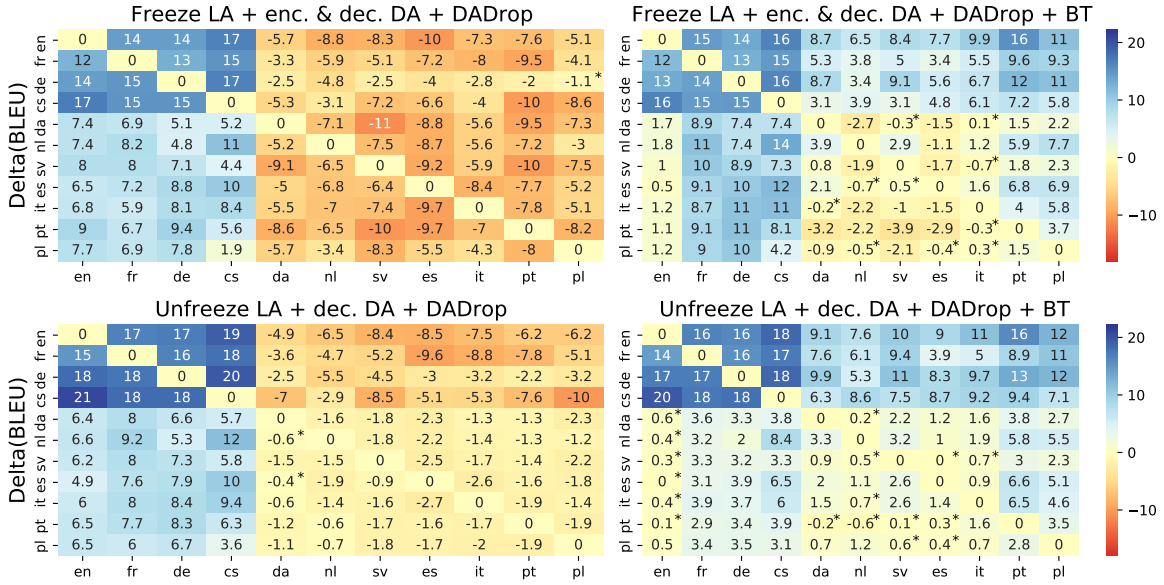
Figure 4: Comparing models with DADrop, back-translation and unfrozen language adapters. Difference in BLEU compared to the baseline (11) ("*" indicates *not* statistically significant). See Figure 2 for more details.

on, even though the original model they are inserted in was trained on those languages. We found that randomly dropping the domain adapter and back-translation can regularize the training and lead to less catastrophic forgetting for when generating into out-of-domain languages, although they do not fully solve the problem of off-target translation.

We experimented with different adapter placement and found that devoting additional capacity to encoder adapters can lead to better results compared to when the same capacity is shared between the encoder and the decoder. Similarly, in the partial resource scenario, models with encoder-only domain adapters suffer less from catastrophic forgetting when translating into out-of-domain languages. In contrast, decoder-only domain adapters perform well when translating *from* out-of-domain into in-domain languages, and combine well with back-translation, perhaps due to their ability to ignore noisy synthetic source data.

Finally we note that a model fine-tuned with domain tags serves as a very competitive baseline for multilingual domain adaptation. On the other hand, domain adaptation with adapters offers modularity, and allows incrementally adapting to new domains without retraining the full model. Future research directions could explore multi-task training combining parallel and monolingual in-domain data in other ways to alleviate the need for back-translation.

Our work is the first attempt to combine domain adapters and language adapters for a generation task (NMT). Although such combinations have shown to be successful for NLU tasks, obtaining good representations for generating unseen target languages proves to be a difficult problem. We believe a fine-grained study of where to use language or domain-specific capacity could lead to better cross-lingual domain transfer in future. Finally, we provide supplementary material to facilitate reproducibility.[10]

Acknowledgements

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.

[10]https://tinyurl.com/r66stbxj

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.

Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark. 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.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A comprehensive survey of multilingual neural machine translation. *arXiv preprint arXiv:2001.01115*.

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 Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

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.

Xavier Garcia, Noah Constant, Ankur P. Parikh, and Orhan Firat. 2021. Towards continual learning for multilingual machine translation via vocabulary substitution. *CoRR*, abs/2103.06799.

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.

Haoming Jiang, Chen Liang, Chong Wang, and Tuo Zhao. 2020. Multi-domain neural machine translation with word-level adaptive layer-wise domain mixing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1834, Online. Association for Computational Linguistics.

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*.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. 2021. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. *CoRR*, abs/2106.15195.

M. McCloskey and N. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165.

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, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. 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.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. 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, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, Long Beach, California, USA.

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

For **bilingual** domain adaptation we use a Transformer Base (Vaswani et al., 2017) model trained 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.). 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 multilingual settings we use the following 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. Both validation sets and test sets contain around 2000 examples for every domain and language-pair.

B MAD-X Style Stacking

Pfeiffer et al. (2020) use the following stacking formulation,

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

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 $\mathrm{LN}_{\mathrm{pre}}$,

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

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. 4 but replace $\mathrm{LA}(\cdot)$ with the identity operation.

C Additional Results for Bilingual Domain Adaptation

Before studying multilingual domain adaptation, we validate some of our ideas on a simpler, *bilingual* German → English domain adaptation setting. Table 11 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.

Table 12 presents results of bilingual domain adaption 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 7 models **(38)** and **(39)**. 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.
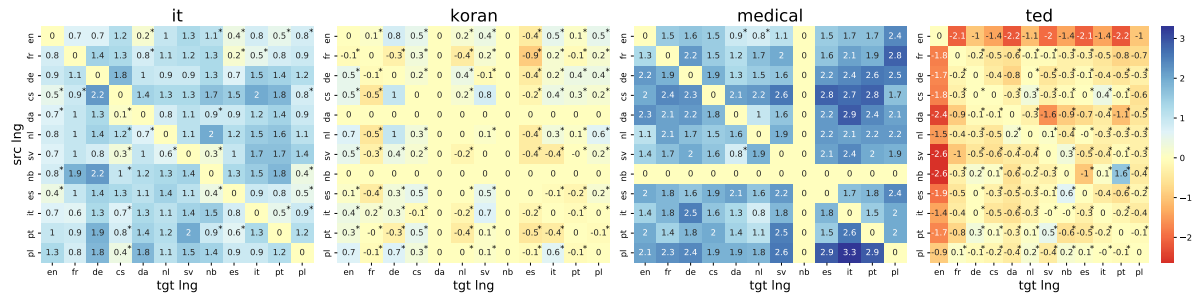
Figure 5: Difference in BLEU score for each domain between the model trained with adapters (9) and model trained with domain tags (3), for the multilingual multi-domain models. A positive number corresponds to the case where model (3) has higher score than the model (9). The "*" indicates the cases when the difference is *not* statistically significant.

D Additional Results for Cross-lingual Transfer

Does language diversity increase 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. 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.

Additional results and metrics We present additional results for the setting discussed in Section 5.2 of the main paper in Table 9 (Koran domain), Table 10 (Koran results for the romance language subset), and Table 7 (additional Medical results). We use the chrF metric as discussed in the main paper, and find the conclusions based on BLEU score are unchanged. 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. Additionally we see as before that combining BT with decoder-only DA works the best, and achieves the highest overall performance. We report on-target (correct language) percentage for all medical domain models in Table 8.

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 (42) and (41)) 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 (43), (44) 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 (11).

source (fr)	ref (pt)	(12)	(13)
La durée du traitement dépend de la nature et de la sévérité de l' infection et de la réponse observée.	A duração do tratamento depende da natureza e da gravidade da infecção e da resposta verificada.	The duration of treatment depends on the nature and severity of the infection and on the response observed.	A duração do tratamento depende da nature e severidade da infecção e da resposta observed.
Insuman Comb 50 40 UI/ ml suspension injectable en flacon	Insuman Comb 50 40 UI/ ml, Suspensão injectável num frasco para injectáveis	Insuman Comb 50 40 IU/ ml suspension injectable en flacon	Insuman Comb 50 40 IU/ ml suspension for injection in vial
A quoi ressemble TAXOTERE et contenu de l' emballage extérieur TAXOTERE 80 mg, solution à diluer pour perfusion est une solution visqueuse, limpide, jaune à jaune marron.	Qual o aspecto de TAXOTERE e contéudo da embalagem TAXOTERE 80 - mg concentrado para solução para perfusão é uma solução viscosa transparente amarela ou amarela- acastanhada.	What TAXOTERE looks like and contents of the pack TAXOTERE 80 mg concentrate para solution for infusion is a solution visqueuse, limpida, de jaune à marron.	TAXOTERE 80 mg, Diluted for Solution for Infusion é uma solution viscous, limpa, yellow to marrom.

Table 4: Some examples of translations generated by straight-forward adapter training settings, in this case from a known source language, *fr* into a target language unseen during domain adaptation, *pt*, and for the medical domain.

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).

ID	Model	IT	Koran	Medical	TED	Params (M)
(1)	Base (En-centric)	.456	.300	.488	.456	N/A
(2)	Finetuned	.623	.394	.625	.513	79
(3)	Finetuned + domain tags	**.645**	**.433**	**.646**	.517	79
(4)	Single adapter per layer ($d = 1024$)	.612	.382	.619	.512	12.6
(5)	LA ($d = 1365$)	.632	.408	.630	.517	202
(6)	LA ($d = 2048$)	.634	.411	.631	.517	303
(7)	LA + dec. DA ($d = 1024$)	.633	.412	.629	.518	177
(8)	LA + enc. DA ($d = 1024$)	.634	.426	.631	.522	177
(9)	LA + enc & dec. DA ($d = 1024$)	.636	.429	.632	**.523**	202

Table 6: chrF scores of various multilingual multi-domain adaptation strategies, i.e. training on all language directions from the 12 languages and all domains.

ID	Model	All	In→in	Out→in	In→out	Out→out
	Oracles					
(10)	Finetune (all langs)	.635	.631	.638	.635	.635
(3)	FT (all langs & domains) + domain tags	.646	.641	.648	.646	.646
	Baselines					
(1)	Base (En-centric)	.488	.500	.497	.493	.475
(11)	**(1)** + ParaCrawl LA	.537	.532	.540	.538	.535
	Straightforward Methods					
(12)	**(1)** + Domain adapters only	.400	.635	.575	.295	.287
(13)	Freeze LA + enc. & dec. DA	.468	.631	.566	.401	.400
(16)	FT (all domains) + dom. tags	.277	**.650**	.236	.283	.193
	Improving Off-target Translation					
(17)	**(16)** + ParaCrawl	.570	.622	.601	.556	.544
(26)	Unfreeze LA	.551	.639	.574	.514	.535
(27)	**(34)** + BT	.571	.636	.556	.594	.548
(15)	Freeze LA + dec. DA	.492	.613	**.605**	.432	.421
(20)	**(15)** + BT	**.586**	.608	.590	.584	**.577**
(28)	**(15)** + BT + DADrop	.584	.604	.587	.583	.576
(14)	Freeze LA + enc. DA	.518	.623	.548	.506	.477
(19)	**(14)** + BT	.548	.620	.567	.574	.498
(29)	**(14)** + BT + DADrop	.561	.614	.570	.579	.527
(30)	Freeze LA + enc. first 3 layers DA	.440	.618	.525	.379	.373
(31)	Freeze LA + enc. last 3 layers DA	.512	.622	.576	.472	.465
(21)	**(13)** + DADrop	.490	.621	.567	.447	.429
(32)	**(13)** + BT	.559	.626	.581	.582	.511
(22)	**(13)** + BT + DADrop	.569	.619	**.583**	.587	.534
(33)	**(13)** + BT + MAD-X style	.540	.625	.575	.561	.477
(23)	Unfreeze LA + dec. DA	.221	.641	.573	.010	.007
(24)	**(23)** + DADrop	.528	.639	.577	.452	.514
(25)	**(23)** + DADrop + BT	.573	.636	.559	.595	.550

Table 7: chrF score of various models trained on the {en, fr, de, cs} subset of the Medical domain. Some models are also included in the main paper.

ID	Model	All	In→in	Out→in	In→out	Out→out
	Oracles					
(10)	Finetune (all langs)	95%	95%	95%	95%	95%
(3)	FT (all langs & domains) + domain tags	95%	95%	96%	95%	95%
	Baselines					
(1)	Base (En-centric)	91%	93%	92%	91%	89%
(11)	(1) + ParaCrawl LA	95%	96%	96%	95%	95%
	Straightforward Methods					
(12)	(1) + Domain adapters only	40%	95%	93%	7%	11%
(13)	Freeze LA + enc. & dec. DA	81%	95%	93%	71%	76%
(16)	FT (all domains) + dom. tags	25%	96%	55%	1%	2%
	Improving Off-target Translation					
(17)	(16) + ParaCrawl	93%	95%	93%	93%	92%
(34)	Unfreeze LA	94%	95%	95%	93%	95%
(35)	(34) + BT	94%	96%	95%	95%	94%
(15)	Freeze LA + dec. DA	84%	95%	95%	77%	77%
(20)	(15) + BT	94%	95%	95%	94%	94%
(36)	(15) + BT + DADrop	94%	95%	95%	94%	94%
(14)	Freeze LA + enc. DA	90%	95%	93%	89%	88%
(19)	(14) + BT	90%	96%	94%	94%	83%
(37)	(14) + BT + DADrop	93%	95%	95%	94%	91%
(38)	Freeze LA + enc. first 3 layers DA	59%	95%	93%	35%	43%
(39)	Freeze LA + enc. last 3 layers DA	88%	95%	94%	83%	86%
(21)	(13) + DADrop	86%	95%	93%	82%	82%
(18)	(13) + BT	91%	95%	95%	94%	85%
(22)	(13) + BT + DADrop	93%	95%	95%	94%	91%
(40)	(13) + BT + MAD-X style	89%	95%	95%	92%	80%
(23)	Unfreeze LA + dec. DA	36%	96%	95%	1%	2%
(24)	(23) + DADrop	90%	95%	95%	82%	92%
(25)	(23) + DADrop + BT	94%	95%	95%	94%	94%

Table 8: On-target translation percentages of various models trained on the {en, fr, de, cs} subset of the Medical domain.

ID	Model	All	In→in	Out→in	In→out	Out→out
	Oracles					
(10)	Finetune (all langs)	.461	.437	.427	.487	.477
(3)	FT (all langs & domains) + domain tags	.433	.423	.409	.455	.438
	Baselines					
(1)	Base (En-centric)	.300	.307	.299	.306	.294
(11)	(1) + ParaCrawl LA	.334	.330	.328	.340	.335
	Straightforward Methods					
(12)	(1) + Domain adapters only	.246	.451	.349	.163	.150
(13)	Freeze LA + enc. & dec. DA	.165	.449	.137	.144	.089
	Improving Off-target Translation					
(16)	FT (all dom.) + dom. tags	.166	.436	.162	.143	.081
(17)	FT (all dom. + ParaCrawl) + dom. tags	.359	.410	.375	.351	.332
(34)	Unfreeze LA	.352	.454	.351	.322	.335
(15)	Freeze LA + dec. DA	.304	.404	**.385**	.249	.244
(41)	(15) + Mono data	.355	.390	.373	.342	.336
(20)	(15) + BT	.381	.399	.371	.387	.375
(36)	(15) + BT + DADrop	**.382**	.402	.373	.388	**.376**
(14)	Freeze LA + enc. DA	.319	.438	.328	.315	.266
(42)	(14) + Mono data	.347	.410	.338	.353	.324
(19)	(14) + BT	.365	.432	.353	.385	.330
(37)	(14) + BT + DADrop	.368	.425	.354	**.394**	.336
(18)	(13) + BT	.374	.434	.366	**.394**	.341
(22)	(13) + BT + DADrop	.381	.436	.369	.406	.349
(23)	Unfreeze LA + dec. DA	.224	.457	.351	.088	.138
(24)	(23) + DADrop	.339	**.458**	.354	.288	.320
(43)	Multi-domain dec. DA	.326	.403	.360	.304	.285
(44)	Multi-domain enc. DA	.337	.412	.360	.327	.297
(45)	Multi-domain enc. & dec. DA	.327	.417	.369	.302	.279

Table 9: chrF 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	In→in	Out→in	In→out	Out→out
(13)	Freeze LA + enc. & dec. DA	.309	.491	.362	.267	.229
(21)	**(13)** + DADrop	.311	.490	.360	.270	.233
(34)	Unfreeze LA	.357	.515	.395	.303	.307
(35)	**(34)** + BT	.372	.529	.364	.370	.319
(15)	Freeze LA + dec. DA	.332	.463	**.418**	.268	.262
(20)	**(15)** + BT	**.382**	.460	.408	.362	**.345**
(14)	Freeze LA + enc. DA	.320	.491	.345	.296	.249
(19)	**(14)** + BT	.359	.492	.374	.354	.298
(23)	Unfreeze LA + dec. DA	.322	.519	.399	.216	.267
(24)	**(23)** + DADrop	.353	.515	.399	.291	.303
(25)	**(23)** + DADrop + BT	.377	**.522**	.372	**.373**	.327
(18)	**(13)** + BT	.368	.490	.388	.363	.308
(22)	**(18)** + DADrop	.373	.494	.389	.369	.314

Table 10: chrF 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	IT	Koran	Medical	Subtitles	Law
(46)	No fine-tuning	35.3	14.8	38.1	26.8	42.4
(47)	Fine-tuned	43.8	22.7	53	30.9	57.9
(48)	Enc. + dec. adapters ($d = 1024$)	42.9	21.8	51.7	30.5	56
(49)	**(48)** + MAD-X style	40.6	19.3	48.8	29.8	54.3
(50)	Dec. adapters ($d = 2048$)	42.1	19.8	50.5	29.7	55.1
(51)	Enc. adapters ($d = 2048$)	42.4	21.5	51.9	30.1	56.1
(52)	Last 3 encoder layers only ($d = 4096$)	42.9	21.1	52.1	30.1	56
(53)	First 3 encoder layers only ($d = 4096$)	42.2	20	50.1	28.5	54.9

Table 11: 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	Medical	Koran	Subtitles	Law
(54)	No fine-tuning	35.3	14.8	38.1	26.8	42.4
(55)	Finetuned	43.8	22.7	53	30.9	57.9
(56)	Enc. + dec. adapters (d=64)	40	18.7	47.3	29.4	51.5
(57)	Dec. adapters (d=128)	39	17.5	46	28.8	50.6
(58)	Enc. adapters (d=128)	40	18.9	47.3	29.2	51.5
(59)	Last 3 encoder layers only (d=256)	40	19	47.3	29	51.1
(60)	First 3 encoder layers only (d=256)	39.5	18	46	28.8	49.5

Table 12: 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.