A Study of Residual Adapters for Multi-Domain Neural Machine Translation

Minh Quang Pham, Josep-Maria Crego, François Yvon, Jean Senellart

A Study of Residual Adapters for Multi-Domain Neural Machine Translation

MinhQuang Pham[†‡], **Josep Crego**[†], **François Yvon**[‡], **Jean Senellart**[†]
[†]SYSTRAN / 5 rue Feydeau, 75002 Paris, France
firstname.lastname@systrangroup.com
[‡]Université Paris-Saclay, CNRS, LIMSI, 91405 Orsay, France
firstname.lastname@limsi.fr

Abstract

Domain adaptation is an old and vexing problem for machine translation systems. The most common and successful approach to supervised adaptation is to fine-tune a baseline system with in-domain parallel data. Standard fine-tuning however modifies all the network parameters, which makes this approach computationally costly and prone to overfitting. A recent, lightweight approach, instead augments a baseline model with supplementary (small) adapter layers, keeping the rest of the model unchanged. This has the additional merit to leave the baseline model intact and adaptable to multiple domains. In this paper, we conduct a thorough analysis of the adapter model in the context of a multidomain machine translation task. We contrast multiple implementations of this idea using two language pairs. Our main conclusions are that residual adapters provide a fast and cheap method for supervised multi-domain adaptation; our two variants prove as effective as the original adapter model and open perspective to also make adapted models more robust to label domain errors.

1 Introduction

Owing to multiple improvements, Neural Machine Translation (NMT) (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017) nowadays delivers useful outputs for many language pairs. However, as many deep learning models, NMT systems need to be trained with sufficiently large amounts of data to reach their best performance. Therefore, the quality of the translation of NMT models is still limited in low-resource language or domain conditions (Duh et al., 2013; Zoph et al., 2016; Koehn and Knowles, 2017). While many approaches have been proposed to improve the quality of NMT models in low-resource domains (see the recent survey of Chu and Wang (2018)), full fine-tuning (Luong and Manning, 2015; Neubig and Hu, 2018) of a generic baseline model remains the dominant supervised approach when adapting NMT models to specific domains.

Under this view, building adapted systems is a two-step process: (a) one first trains NMT with the largest possible parallel corpora, aggregating texts from multiple, heterogeneous sources; (b) assuming that in-domain parallel documents are available for the domain of interest, one then adapts the pre-trained model by resuming training with the sole in-domain corpus. It is a conjecture that the pretrained model constitutes a better initialization than a random one, especially when adaptation data is scarce. Indeed, studies of transfer learning for NMT such as Artetxe et al. (2020); Aji et al. (2020) have confirmed this claim in extensive experiments. Full fine-tuning, that adapts all the parameters of a baseline model usually significantly improves the quality of the NMT for the chosen domain. However, it also yields large losses in translation quality for other domains, a phenomenon referred to as "catastrophic forgetting" in the neural network literature (McCloskey and Cohen, 1989). Therefore, a fully fine-tuned model is *only useful to one target domain*. As the number of domains to handle grows, training, and maintaining a separate model for each task can quickly become tedious and resource-expensive.

Several recent studies (e.g. (Vilar, 2018; Wuebker et al., 2018; Michel and Neubig, 2018; Bapna and Firat, 2019)) have proposed more lightweight schemes to perform domain adaptation, while also preserving the value of pre-trained models. Our main inspiration is the latter work, whose proposal relies on small *adapter components* that are plugged in each hidden layer. These adapters are trained only with the in-domain data, keeping the pre-trained model frozen. Because these additional

Proceedings of the 5th Conference on Machine Translation (WMT), pages 615–626
Online, November 19–20, 2020.

adapters are very small compared to the size of the baseline model, their use significantly reduces the cost of training and maintaining fine-tuned models, while delivering a performance that remains close to that of full fine-tuning.

In this paper, we would like to extend this architecture to improve NMT in several settings that still challenge automatic translation, such as translating texts from multiple topics, genre, or domains, in the face of unbalanced data distributions. Furthermore, as the notion of "domains" is not always well established, another practical setting is the translation of texts mixing several topics/domains. An additional requirement is to translate texts from domains unseen in training, based only on the unadapted system, which should then be made as strong as possible.

In this context, our main contribution is a thorough experimental study of the use of residual adapters for multi-domain translation. We notably explore ways to adjust and/or regularize adapter modules to handle situations where the adaptation data is very small. We also propose and contrast two new variants of the residual architecture: in the first one (*highway residual adapters*), adaptation still affects each layer of the architecture, but its effect is delayed till the last layer, thus making the architecture more modular and adaptive; our second variant (*gated residual adapters*) exploits this modularity and enables us to explore ways to improve performance in the face of train-test data mismatch. We experiment with two language pairs and report results that illustrate the flexibility and effectiveness of these architectures.

2 Residual adapters

In this section, we describe the basic version of the residual adapter architectures (Houlsby et al., 2019; Bapna and Firat, 2019), as well as two novel variants of this model.

2.1 Basic architecture

2.1.1 The computation of adapter layers

Our reference architecture is the Transformer model of Vaswani et al. (2017), which we assume contains a stack of layers both on the encoder and the decoder sides. Each layer contains two subparts, an attention layer, and a dense layer. Details vary from one implementation to another, we simply contend here that each layer $i \in \{1 \dots L\}$ (in the encoder or the decoder) computes a transform of a fixed-length sequence of d-dimensional input vectors h^i into a sequence of output vectors h^{i+1} as follows ($\mathbf{LN}$ denotes the (sub)layer normalization, $\mathbf{ReLU}$ is the "rectified linear unit" operator):

$$
\begin{aligned}
h_0^i &= \mathbf{LN}(h^i) \\
h_1^i &= \mathbf{W}_{db}^i h_0^i + a_1^i \\
h_2^i &= \mathbf{ReLU}(h_1^i) \\
h_3^i &= \mathbf{W}_{bd}^i h_2^i + a_2^i \\
\bar{h}^i &= h_3^i + h^i.
\end{aligned}
$$

Overall, the i^{th} adapter is thus parameterized by matrices $\mathbf{W}_{db}^i \in \mathbb{R}^{d \times b}$,$\mathbf{W}_{bd}^i \in \mathbb{R}^{b \times d}$, bias vectors $b_1^i \in \mathbb{R}^b$, $b_2^i \in \mathbb{R}^d$, with b the dimension of the adapter . For the sake of brevity, we will simply denote $h_3^i = \text{ADAP}^{(i)}(h^i)$, and $\theta_{\text{ADAP}^{(i)}}$ the corresponding set of parameters.

The "adapted" hidden vectors $\bar{h}^i_{1 \le i \le L-1}$, where L is the number of layers, will then be the input of the $(i+1)^{\text{th}}$ layer; $\bar{h}^L$ is passed to the decoder if it belongs to the encoder side, or is the input of output layer if it belongs to the decoder side. Note that zeroing out all adapters enables us to recover the basic Transformer, with $\bar{h}^i = h^i$ for all i.

In the experiments of Section 3, we use $2 \times L = 12$ residual adapters, one for each of the $L = 6$ attention layers of the encoder and similarly for the decoder.[1]

2.1.2 Design space and variants

This general architecture leaves open many design choices pertaining to the details of the network organization, the training procedure, and the corresponding objective function.

The first question is the number of adapter layers. While in principle, all Transformer layers can be subject to adaptation, it is nonetheless worthwhile to consider simpler adaptation schemes, which would only alter a limited number of layers. Such strategy might be especially relevant when the training data contains very small domains, as in the experiments of Section 3, and for which a complete adaptation may not be necessary or/and or prone to overfitting. Likewise, it might be meaningful to explore ways to share subsets of adapters across domains. This, in turn, raises the issue of which layer(s) to adapt, a question that can be approached in the light of recent analyses of Transformers models, which conjecture that the higher layers encode

[1] In the decoder, the stack of self-attention and cross encoder-decoder attention only counts as one attention layer and only corresponds to one residual adapter.

global patterns with a more "semantic" interpretation, while the lower layers encode local patterns akin to morpho-syntactic information (Raganato and Tiedemann, 2018).

A related question concerns the regularization of adapter layers to mitigate overfitting. Reducing the number of adapters, or their dimensions, is simple, but such choices are difficult to optimize numerically – an issue that becomes important as the number of domain grows. Less naive alternatives can also be entertained, such as applying weight decay or layer regularization to the adapter. Implementing these requires to modify the objective function in a way that still allows for a smooth optimization problem. For instance, weight decay applies a penalization on the weights of the adapters, complementing the cross-entropy term with a function of the norm of the parameters:

$$\bar{L} = \frac{1}{\#(x,y)} \sum_{x,y} (-\log(P(y|x))) + \lambda \sum_{i \in \{1,..,6\} \otimes \{enc,dec\}} \|\theta_{\mathrm{ADAP}^{(i)}}\|_2$$

An alternative scheme is *layer regularization*, which penalizes the output of the adapters, corresponding to the following objective:

$$\bar{L} = \frac{1}{\#(x,y)} \sum_{x,y} (-\log(P(y|x)) + \lambda \sum_{i \in \{1,..,6\} \otimes \{enc,dec\}} \|\mathrm{ADAP}^{(i)}(h_i(x,y))\|_2)$$

Finally, another independent design choice relates to the training strategy for adapters. A first option is to generalize supervised domain adaptation to multi-domain adaptation and to proceed in two steps: (a) train a generic model with all the available data; (b) train each adapter layer with domain-specific data, keeping the generic model parameters unchanged. Another strategy is to adopt the view of Dredze and Crammer (2008), where the multi-domain setting is viewed as an instance of multi-task learning (Caruana, 1997) with each domain corresponding to a specific task. This suggests training all the parameters from scratch, as we would do in a multi-task mode. The generic parameters will still depend on all the available data, while each adapter will only be trained with the corresponding in-domain data.

2.2 Highway Residual Adapters

In the basic architecture described in Section 2.1, the computation performed by lower level layers will impact all the subsequent layers. In this section, we introduce an alternative implementation of the same idea, which however delays the adaptation of each layer to the last layer (of the encoder or the decoder) as depicted on Figure 1. While the basic architecture performs adaptation in sequence, we propose here to perform it in parallel. In this version, only the last hidden vector of the encoder (decoder) is thus modified according to:

$$\bar{h}^L = h^L + \sum_{1 \le i \le L} ADAP^i(h^i) \qquad (1)$$

One obvious benefit of this variant is that it allows us to reuse the hidden vectors h^i of all hidden layers when computing an adapted output for several domains during the inference. In this situation, the forward step needs only to compute the hidden vectors h^i once for the inner encoder layers, before an adapted sequence of vectors is computed at the topmost layer. Therefore, we can fine-tune the model to multiple domains at once without recomputing h^i. This variant also opens the way to more parameter sharing across adapters, a perspective that we will not explore further in this work. Instead, we use it to develop a second variation of the adapter model, that is presented in the next section.

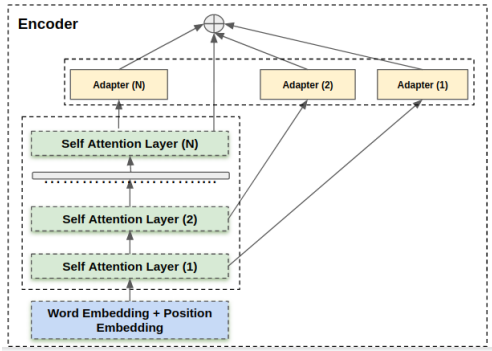

Figure 1: Highway residual adapter network

2.3 Gated Residual Adapters

The basic architecture presented above rests on a rather simplistic view of "domains" as made of well-separated and unrelated pieces of texts that are processed independently during adaptation. Likewise, when translating test documents, one needs to choose between either using one specific domain-adapted model or resorting to the generic model. In

this context, using wrong domain labels can have a strong (negative) effect on translation performance.

Therefore, we would like to design a version of residual adapters that is more robust to such domain errors. This variant, called the *gated residual adapter model*, relies on the training of a supplementary component that will help decide whether to activate, on a word per word basis, a given residual layer and to regulate the strength of this activation. To this end, we extend the highway version of residual adapters as follows.

Formally, we replace the adapter computation of equation (1) and take the adapted hidden (topmost) layer to be computed as (this is for domain k):

$$\bar{h}^L = h^L + \sum_{1 \leq i \leq L} \text{ADAP}_k^i(h^i) \odot z_k(h^L), \quad (2)$$

where the scalar $z_k(h^L[t]) \in [0, 1]$ measures the relatedness of the t^{th} word w_t to domain k. The more likely w_t is in domain k, the larger $z_k(h^L[t])$ should be; conversely, for words[2] that are not typical of any domain k (eg. function words), adaptation is minimum and the corresponding adapted encoder output ($\bar{h}^L[t]$) will remain close to the output of the generic model ($h^L[t]$). In our implementation, we incorporate two domain classifiers on top of the encoder and the decoder, that take the last hidden layer of the encoder (resp. decoder) as input and use the posterior probability $P(k|h^L[t])$ of domain k as the value for $z_k(h^L[t])$.

Training gated residual adapters thus comprises three steps, instead of two for the baseline version:

1. learn a generic model with mixed corpora from multiple domains.

2. train a domain classifier on top of the encoder and decoder; during this step, the parameters of the generic model are frozen. This model computes the posterior domain probability $P(k|h^L[t])$ for each word w_t, based on the representation computed by the last layer.

3. train the parameters of adapters with in-domain data separately for each domain, while freezing all the other parameters.

[2]The term "word" is employed here by mere convenience, as systems only manipulate sub-lexical BPE units; furthermore, the values of the hidden representations h^i at position t depend upon all the other positions in the sentence.

3 Experimental settings

3.1 Data and metrics

We perform our experiments with two translation pairs involving multiple domains: English-French (En→Fr) and English-German (En→De). For the former pair, we use texts[3] initially from 6 domains, corresponding to the following data sources: the UFAL Medical corpus V1.0 (MED)[4], the European Central Bank corpus (BANK) (Tiedemann, 2012); The JRC-Acquis Communautaire corpus (LAW) (Steinberger et al., 2006), documentations for KDE, Ubuntu, GNOME and PHP from Opus collection (Tiedemann, 2009), collectively merged in a IT-domain, Ted Talks (TALK) (Cettolo et al., 2012), and the Koran (REL). Complementary experiments also use v12 of the News Commentary corpus (NEWS). Corpus statistics are in Table 1.

En→De is a much larger task, for which we use corpora distributed for the News task of WMT20[5] including: European Central Bank corpus (BANK), European Economic and Social Committee corpus (ECO), European Medicines Agency corpus (MED)[6], Press Release Database of European Commission corpus, News Commentary v15 corpus, Common Crawl corpus (NEWS), Europarl v10 (GOV), Tilde MODEL - czechtourism (TOUR)[7], Paracrawl and Wikipedia Matrix (WEB). Statistics are in Table 2.

We randomly select in each corpus a development and a test set of 1,000 lines each and keep the rest for training.[8] Development sets help choose the best model according to the average BLEU score (Papineni et al., 2002).[9]

3.2 Baseline architectures

Using Transformers (Vaswani et al., 2017) implemented in OpenNMT-tf[10] (Klein et al., 2017), we train the following baselines:

- a generic model trained on a concatenation of all corpora, denoted **Mixed**;

[3]Most corpora are available from the Opus web site: http://opus.nlpl.eu

[4]https://ufal.mff.cuni.cz/ufal_medical_corpus

[5]http://www.statmt.org/wmt20/news.html

[6]https://tilde-model.s3-eu-west-1.amazonaws.com/Tilde_MODEL_Corpus.html

[7]https://tilde-model.s3-eu-west-1.amazonaws.com/Tilde_MODEL_Corpus.html

[8]Scripts to replicate these experiments are available at urlhttps://github.com/qmpham/experiments.git.

[9]We use truecasing and the multibleu script.

[10]https://github.com/OpenNMT/OpenNMT-tf

MED	LAW	BANK	IT	TALK	REL	NEWS
2609 (0.68)	190 (0.05)	501 (0.13)	270 (0.07)	160 (0.04)	130 (0.03)	260 (0)

Table 1: Corpora statistics for En→Fr : number of parallel lines ($\times 10^3$) and proportion in the basic domain mixture (which does not include the NEWS domain). MED is the largest domain, containing almost 70% of the sentences, while REL is the smallest, with only 3% of the data.

BANK	ECO	MED	GOV	NEWS	TOUR	WEB
4 (0.00022)	2857 (0.15)	347 (0.018)	1828 (0.095)	3696 (0.19)	7 (0.00039)	10473 (0.54)

Table 2: Corpora statistics for En→De: number of parallel lines ($\times 10^3$) and proportion in the basic domain mixture. WEB is the largest domain, containing about 54% of the sentences, while BANK and TOUR are very small.

- a fine-tuned model (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016), based on the **Mixed** system, further trained on each domain with early stopping when the development BLEU score stops increasing during 3 consecutive epochs.

For all En→Fr models, we set the embeddings size and the hidden layers size to 512. Transformers use multi-head attention with 8 heads in each of the 6 layers; the inner feedforward layer contains 2,048 cells. Residual adapters additionally use an adaptation block in each layer, composed of a 2-layer perceptron, with an inner ReLU activation function operating on normalized entries of dimension $b = 1024$. Bapna and Firat (2019) showed that the performance of adapted models increases with respect to the size of the inner dimension and obtained performance close to the full fine-tuned model with $b = 1024$, which is twice as large as the dimension of a Transformer layer. We used the same setting in our experiments.

Training uses a batch size of 12,288 tokens; optimization uses Adam with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$ and Noam decay ($warmup\_steps = 4,000$), and a dropout rate of 0.1 for all layers. For the **Mixed** model, we use an initial learning rate of 1.0 and take the concatenation of the validation sets of 6 domains for development. In the fine-tuning experiments, we continue training using **Mixed** as starting point, using the same learning rate schedule, and continuing the incrementation of the number of steps. In the multi-task training, we use the same learning rate schedule as for **Mixed**: for each iteration, we sample a domain a probability proportional to its size; we then sample a batch of 12,288 tokens that is used to update the shared parameters and the parameters of the corresponding adapter.

Models for En→De are larger and rely on embeddings as well as hidden layers of size 1024; each Transformers layer contains 16 attention heads; the inner feedforward layer contains 4,096 cells. Adapter modules have the same architecture as for the other language pair, except for their size, which is doubled ($b = 2,048$).

3.3 Multi-domain systems

In this section, we evaluate several proposals from the literature on multi-domain adaptation and compare them to full fine-tuning on the one hand, and to two variants of the residual adapter architecture on the other hand. The reference methods included in our experiments are the following:

- a system using "domain control" (Kobus et al., 2017). In this approach, domain information is introduced either as an additional token for each source sentence (**DC-Tag**) or in the form of a supplementary feature for each word (**DC-Feat**);

- a system using lexicalized domain representations (Pham et al., 2019): word embeddings are composed of a generic and a domain-specific part (**LDR**);

- the three proposals of Britz et al. (2017). **TTM** is a feature-based approach where the domain tag is introduced as an extra word *on the target side*. The training uses reference tags and inference is performed with predicted tags, just like for regular target words. **DM** is a multi-task learner where a domain classifier is trained on top of the MT encoder, so as to make it aware of domain differences; **ADM** is the adversarial version of **DM**, pushing the encoder towards learning domain-independent source representations. These methods only use domain labels in training.

Model / Domain	MED	LAW	BANK	TALK	IT	REL	AVG
Mixed	37.3	54.6	50.1	33.5	43.2	77.5	49.4
FT-Full	37.7	59.2	54.5	34.0	46.8	90.8	53.8
DC-Tag	38.1	55.3	49.9	33.2	43.5	80.5	50.1
DC-Feat	37.7	54.9	49.5	32.9	43.6	79.9	49.9
LDR	37.0	54.7	49.9	33.9	43.6	79.9	49.8
TTM	37.3	54.9	49.5	32.9	43.6	79.9	49.7
DM	35.6	49.5	45.6	29.9	37.1	62.4	43.4
ADM	36.4	53.5	48.3	32.0	41.5	73.4	47.5
Res-Adap	37.3	57.9	53.9	33.8	46.7	90.2	53.3
Res-Adap-MT	37.9	56.0	51.2	33.5	44.4	88.3	51.9
Res-Adap-MT$^+$	37.5	57.1	52.4	33.7	46.2	89.5	52.7
Res-Adap-MT (gen)	37.7	51.0	34.0	30.4	34.2	15.2	36.4

Table 3: Translation performance of various multi-domain MT systems (En→Fr) compared to variants of the residual adapter models.

The two variants of the residual adapter model included in this first round of experiment have been presented in Section 2.1: **Res-Adap** is the multi-domain version of the approach of Bapna and Firat (2019) based on a two-step training procedure; while **Res-Adap-MT** is the "multi-task" version, where the parameters of the generic model and of the adapters are jointly learned from scratch. We also report results for the same system, using the the parameters of the **Mixed** model as initialization (**Res-Adap-MT**$^+$).[11]

Because of the limit of our computational resources, we restrict the experiments in this section to the En→Fr task. Results are in Table 3.

These results first show that full fine-tuning outperforms all other methods for the in-domain test sets. However, **Res-Adap** is able to reduce the gap with this approach for several domains, showing the effectiveness of residual adapters. The "multi-task" variant is slightly less effective in our experiments than the basic version, where optimization is performed in two steps. As it turns out, using residual adapters proves here better on average than the other reference multi-domain systems; it is also much better than the generic system for translating data from known domains, outperforming the **Mixed** system by more than 4 BLEU points in average. Gains are especially large for small domains such as LAW and REL.

Comparing training schemes (**Res-Adap** vs **Res-Adap-MT** vs **Res-Adap-MT**$^+$) suggests that the simultaneous learning of all parameters is detrimental to performance in our settings: we see that the 2-step procedure implemented in **Res-Adap** always yields the best scores, even when **Res-Adap-MT** is initialized with good parameter values . This may be because in this setting, the adapters have access to a stable version of the generic system. The last line (Res-Adap-MT (gen)) gives the results for a **Res-Adap-MT** trained system in which we cancel the adapter in inference - comparing this to **Mixed** shows how differently the generic parts of these two systems behave.

3.4 Varying the positions and number of residual adapters

Tables 4-5 report BLEU scores for 6 domains in each language pair: MED,LAW,BANK,TALK,IT and REL for En→Fr; GOV, ECO, TOUR, BANK, MED and NEWS for En→De. We first see that for the latter direction, the basic version **Res-Adap** also outperforms the **mixed** baseline on average, with large gains for the small domains TOUR, BANK and comparable results for the other domains.

By varying the number and position of residual adapters (see Section 2.1), we then contrast several implementations. Because the set of possible configurations is large, we only perform experiments for layers $i = 2, 4, 6$ (both for the encoder and decoder). Two settings are considered: keeping just one adapter or keeping the three. The trend is the same for the two language directions: suppressing adapters always hurts the overall performance, albeit by a small margin: having six adapters is better than three, which is better than keeping only one. With only one adapter active, we observe small,

[11] This system also includes a layer dropout policy that cancels adapter layers with probability 0.5

insignificant changes in performance when varying the adapter's depth.

3.5 Regularizing fine-tuning

The translation from English into German includes two domains (TOUR and BANK) that are extremely small and account only for a very small fraction of the training data (respectively for 0.039% and 0.022% of the total number of sentences). Fine-tuning on these domains can lead to serious overfitting. We assess two well-known regularization techniques for adapter modules, that could help mitigate this problem: weight decay and layer regularization.

For each method, the optimal hyper-parameter λ (weight decay or layer regularization coefficient, see Section 2.1.2) are chosen by grid search in a small set of values ($\{10^{-3}, 10^{-4}, 10^{-5}\}$).

Results in Tables 4 and 5 show that regularizing the adapter model can positively impact the test performance for the smallest domains (this is especially clear for weight-decay (**Res-Adap-WD**) in En→De), at the cost however of a small drop in performance for the other domains. Using layer regularization proves here to be comparatively less effective. Finding better ways to set the regularization parameters, for instance by varying λ for each domain based on the available supervision data, is left for future work.

3.6 Highway and Gated Residual Adaptaters

We now turn to the evaluation of our new architectural variants: Highway residual adapters **Res-Adap-HW** on the one hand, and Gated residual adapters **Res-Adap-Gated** on the other hand. We use the same domains and settings as before, focusing here exclusively on the language direction En→Fr.

To also evaluate the robustness with respect to out-of-domain examples, we perform two additional experiments. We first generate translations with erroneous (more precisely: randomly assigned) domain information: the corresponding results appear in Table 6 under column RND. We also compute translation for a domain unseen in training (NEWS) as follows. For each sentence of this test set, we automatically evaluate the closest domain,[12] then use the predicted domain label to compute the translation. This is an error-prone process, which also challenges the robustness of our multi-domain systems. Results are in Table 6.

A first observation is that for domains seen in training, our variants **Res-Adap-HW** and **Res-Adap-Gated** achieve BLEU scores that are on a par to those of the original version (**Res-Adap**), with insignificant variations across test sets.

The two other settings are instructive in several ways: they first clearly illustrate the brittleness of domain-adapted systems, for which large drops in performance (more than 15 BLEU points on average) are observed when the domain label is randomly chosen. Our gated variant however proves much more robust than the other adaptation strategy and performs almost on par to the generic system for that test condition. The same trend holds for the unseen NEWS domain, with **Res-Adap-Gated** being the best domain adapted system in our set, outperforming the other variants by about 2 BLEU points.

4 Related Work

Training with data from multiple, heterogeneous sources is a common scenario in natural language processing (Dredze and Crammer, 2008; Finkel and Manning, 2009). It is thus no wonder that the design of multi-domain systems has been proposed for many tasks. In this short survey, we exclusively focus on machine translation; it is likely that similar methods (parameter sharing, instance selection/weighting, adversarial training, etc) have also been proposed for other tasks.

Early approaches to multi-domain MT were proposed for statistical MT, either considering multiple data sources (eg. Banerjee et al. (2010); Clark et al. (2012); Sennrich et al. (2013); Huck et al. (2015)) or domains containing several topics (Eidelman et al., 2012; Hasler et al., 2014). Two main strategies emerge: feature-based methods, where domain labels are integrated through supplementary features; and instance-based methods, involving a measure of similarity between train and test domains.

The former approach has also been adapted to NMT: Kobus et al. (2017); Tars and Fishel (2018) use an additional domain feature in an RNN model, in the form of an extra domain-token or of additional domain-features associated with each word. Chen et al. (2016) apply domain control on the *target* side, using a topic vector to describe the

[12] As measured by the perplexity of a language model trained with only in-domain data..

Model / Domain	MED	LAW	BANK	TALK	IT	REL	AVG	PARAMS
Mixed	37.3	54.6	50.1	33.5	43.2	77.5	49.4	65M/0
Res-Adap	37.3	57.9	53.9	33.8	46.7	90.2	53.3	65M/12M
Res-Adap$_{(2,4,6)}$	37.7	57	53	33.3	45	90	52.7	65M/6M
Res-Adap$_{(6)}$	37.7	55.8	51.5	33.9	43.6	89.2	51.9	65M/2M
Res-Adap$_{(4)}$	37.9	55.6	51.7	33.7	44.4	88.7	52	65M/2M
Res-Adap$_{(2)}$	37.8	55.5	51.4	34	43.8	86.7	51.5	65M/2M
Res-Adap-WD	37.2	56.0	52.9	33.4	46.0	90.6	52.7	65M/12M
Res-Adap-LR	37.4	56.1	51.8	33.3	45.0	89.7	52.2	65M/12M

Table 4: Translation performance of various fine-tuned systems (En→Fr). We report BLEU scores for each domain, as well as averages across domains. Column PARAMS reports the number of domain-agnostic/domain-specific parameters.

Model / Domain	GOV	ECO	TOUR	BANK	MED	NEWS	AVG	PARAMS
Mixed	29.3	30.5	17.6	38.1	47.9	20.9	30.6	213M/0M
Res-Adap	29.6	30.4	19.2	49.0	47.2	20.6	33.1	213M/48M
Res-Adap$_{(2,4,6)}$	29.7	30.5	18.8	49.6	47.1	20.6	32.7	213M/24M
Res-Adap$_{(6)}$	29.5	30.4	18.1	49.1	46.9	20.4	32.4	213M/8M
Res-Adap$_{(4)}$	29.7	30.4	18.1	49.6	47.0	20.6	32.6	213M/8M
Res-Adap$_{(2)}$	29.6	30.4	18.3	49.4	46.7	20.6	32.5	213M/8M
Res-Adap-WD	29.7	30.8	20.4	50.2	47.7	20.6	33.2	213M/48M
Res-Adap-LR	29.6	30.4	19.2	49.0	47.2	20.6	33.1	213M/48M

Table 5: Translation performance of various fine-tuned systems (En→De). We report BLEU scores for each domain, as well as averages across domains. Column PARAMS reports the number of domain-agnostic/domain-specific parameters.

Model / Domain	MED	LAW	BANK	TALK	IT	REL	AVG	RND	NEWS
Mixed	37.3	54.6	50.1	33.5	43.2	77.5	49.4	49.4	23.5
FT-Full	37.7	59.2	54.5	34.0	46.8	90.8	53.8	32.5	20.2
Res-Adap	37.3	57.9	53.9	33.8	46.7	90.2	53.3	38.4	20.5
Res-Adap-HW	37.5	57.2	53.4	33.1	46.3	91.0	53.1	36.6	20.2
Res-Adap-HW-MT	37.4	56.4	52.1	33.7	44.8	89.8	52.4	27.1	20.4
Res-Adap-HW-MT^{+}	37.7	57.0	52.5	33.5	46.1	89.0	52.6	46.5	21.4
Res-Adap-Gate	38.0	57.5	53.0	33.5	46.0	90.1	53.0	49.0	22.5

Table 6: Translation performance of highway and gated variants for En→Fr. NEWS is excluded from the training data and considered as an out-of-domain test.

whole document context. Similar ideas are developed in Chu and Dabre (2018); Pham et al. (2019), where domain differences and similarities are enforced through parameter sharing schemes. Parameter-sharing also lies at the core of the work by Jiang et al. (2019), who consider a Transformer model containing both domain-specific and domain-agnostic heads.

Britz et al. (2017) study three general techniques to take domain information into account in training: they rely on either domain classification or domain normalization on the source or target side. A contribution of this study is an adversarial training scheme to normalize representations across domains and make the combination of multiple data sources more effective. Similar techniques (parameter sharing, automatic domain classification/normalization) are at play in Zeng et al. (2018): in this work, the lower layers of the MT use auxiliary classification tasks to disentangle domain-specific from domain-agnostic representations. These representations are first processed separately, then merged to compute the final translation.

Farajian et al. (2017); Li et al. (2018) are two recent representatives of the instance-based approach: for each test sentence, a small adaptation corpus is collected based on similarity measures and used to fine-tune a mix-domain model. As shown in the former work, also adapting the training regime on a per sentence basis is crucial to make these techniques really effective.

Finally, note that a distinct evolution of the residual adapter model of Bapna and Firat (2019) is presented in Sharaf et al. (2020), where meta-learning techniques are used to make fine-tuning more effective in a standard domain-adaptation setting.

5 Conclusion and outlook

In this paper, we have performed an experimental study of the residual adapter architecture in the context of multi-domain adaptation, where the goal is to build one single system that (a) performs well for domain seen in training, ideally as well as full fine-tuning; (b) is also able to robustly handle translations for new, unseen domains. We have shown that this architecture allowed us to quickly adapt a model to a specific domain, delivering BLEU performance than are much better than the generic, mixed domain baseline, and close the gap with the full-finetuning approach, at a modest computational cost. Several new variants have been introduced and evaluated for two language directions: if none that able to clearly surpass the baseline, residual adapter models, they provide directions for improving this model in practical settings: unbalanced data condition, noise in label domains, etc. In our future work, we would like to continue the development of the gated variant, which, it seems to us, provides a flexible and robust tool to address the various challenges of multi-domain machine translation.

Acknowledgements

This work was granted access to the HPC resources of [TGCC/CINES/IDRIS] under the allocation 2020- [AD011011270] made by GENCI (Grand Equipement National de Calcul Intensif)

References

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710, Online. Association for Computational Linguistics.

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

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, ICLR, San Diego, CA.

Pratyush Banerjee, Jinhua Du, Baoli Li, Sudip Kumar Naskar, Andy Way, and Josef van Genabith. 2010. Combining multi-domain statistical machine translation models using automatic classifiers. In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas*, AMTA 2010, Denver, CO, USA.

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.

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.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. In *Proceedings of the Twelth Biennial Conference of the Association for Machine Translation in the Americas*, AMTA 2012, Austin, Texas.

Chenhui Chu and Raj Dabre. 2018. Multilingual and multi-domain adaptation for neural machine translation. In *Proceedings of the 24st Annual Meeting of the Association for Natural Language Processing*, NLP 2018, pages 909–912, Okayama, Japan.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, COLING 2018, pages 1304–1319, Santa Fe, New Mexico, USA.

Jonathan H. Clark, Alon Lavie, and Chris Dyer. 2012. One system, many domains: Open-domain statistical machine translation via feature augmentation. In *Proceedings of the Tenth Biennial Conference of the Association for Machine Translation in the Americas*, (AMTA 2012), San Diego, CA.

Mark Dredze and Koby Crammer. 2008. Online methods for multi-domain learning and adaptation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 689–697, Honolulu, Hawaii.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 678–683, Sofia, Bulgaria. Association for Computational Linguistics.

Vladimir Eidelman, Jordan Boyd-Graber, and Philip Resnik. 2012. Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 115–119, Jeju Island, Korea. Association for Computational Linguistics.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark.

Jenny Rose Finkel and Christopher D. Manning. 2009. Hierarchical Bayesian domain adaptation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 602–610, Boulder, Colorado.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.

Eva Hasler, Phil Blunsom, Philipp Koehn, and Barry Haddow. 2014. Dynamic topic adaptation for phrase-based MT. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 328–337, Gothenburg, Sweden. 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. volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799, Long Beach, California, USA. PMLR.

Matthias Huck, Alexandra Birch, and Barry Haddow. 2015. Mixed domain vs. multi-domain statistical machine translation. In *Procedings of the Machine Translation Summit*, MT Summit XV, pages 240–255, Miami Florida.

Haoming Jiang, Chen Liang, Chong Wang, and Tuo Zhao. 2019. Multi-domain neural machine translation with word-level adaptive layer-wise domain mixing. *CoRR*, abs/1911.02692.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

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.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2018. One sentence one model for neural machine translation. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *Proceedings of the International Workshop on Spoken Language Translation*, IWSLT, Da Nang, Vietnam.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation - Advances in Research and Theory*, 24(C):109–165.

Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 312–318, Melbourne, Australia. Association for Computational Linguistics.

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA.

Minh Quang Pham, Josep-Maria Crego, Jean Senellart, and François Yvon. 2019. Generic and Specialized Word Embeddings for Multi-Domain Machine Translation. In *Proceedings of the 16th International Workshop on Spoken Language Translation*, IWSLT, page 9p, Hong-Kong, CN.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Holger Schwenk, and Walid Aransa. 2013. A multi-domain translation model framework for statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 832–840, Sofia, Bulgaria. Association for Computational Linguistics.

Amr Sharaf, Hany Hassan, and Hal Daumé III. 2020. Meta-learning for few-shot NMT adaptation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 43–53, Online. Association for Computational Linguistics.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Toma Erjavec, Dan Tufis, and Dniel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, LREC'06, Genoa, Italy. European Language Resources Association (ELRA).

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Montreal, Canada. MIT Press.

Sander Tars and Mark Fishel. 2018. Multi-domain neural machine translation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, EAMT, pages 259–269, Alicante, Spain. EAMT.

Jörg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

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, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

David Vilar. 2018. Learning hidden unit contribution for adapting neural machine translation models. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 500–505, New Orleans, Louisiana.

Joern Wuebker, Patrick Simianer, and John DeNero. 2018. Compact personalized models for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language*

Processing, pages 881–886, Brussels, Belgium. Association for Computational Linguistics.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Brussels, Belgium. Association for Computational Linguistics.

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