Factorized Transformer for Multi-Domain Neural Machine Translation

Yongchao Deng¹, Hongfei Yu², Heng Yu¹, Xiangyu Duan², Weihua Luo¹

¹Machine Intelligence Technology Lab, Alibaba DAMO Academy

²Soochow University, Suzhou, China

{yongchao.dyc, yuheng.yh, weihua.luowh}@alibaba-inc.com
 hfyutravel@gmail.com,xiangyuduan@suda.edu.cn

Abstract

Multi-Domain Neural Machine Translation (NMT) aims at building a single system that performs well on a range of target domains. However, along with the extreme diversity of cross-domain wording and phrasing style, the imperfections of training data distribution and the inherent defects of the current sequential learning process all contribute to making the task of multi-domain NMT very challenging.

To mitigate these problems, we propose the *Factorized Transformer*, which consists of an in-depth factorization of the parameters of an NMT model, namely Transformer in this paper, into two categories: domain-shared ones that encode common cross-domain knowledge and domain-specific ones that are private for each constituent domain. We experiment with various designs of our model and conduct extensive validations on English to French open multi-domain dataset. Our approach achieves state-of-the-art performance and opens up new perspectives for multi-domain and opendomain applications.

1 Introduction

Recent advances in Neural Machine Translation (NMT) (Bahdanau et al., 2015; Vaswani et al., 2017) have led to significant improvement in terms of translation quality (Wu et al., 2016; Hassan et al., 2018), opening new perspectives for Machine Translation in real-world scenarios. In order to deliver trust-worthy translations for end users, an NMT system is often required to meet expert-level translation quality in one or multiple related target domains, while performing well enough on a range of generic subjects, just like human experts do.

However, requiring a single NMT system to perform well on multiple distant domains simultaneously is a very challenging task. First, languages are highly polysemous: the same words or expressions may have different meanings in different contexts. Also wording and syntactic style may significantly vary depending on the domains. Second, a multi-domain NMT system in general suffers from two major issues: *Domain Bias* and *Catastrophic Forgetting* (Mccloskey and Cohen, 1989; Kirkpatrick et al., 2016; Thompson et al., 2019). While the former biases the model toward well-represented domains to the detriment of the low-resource ones, the latter makes the sequential learning process difficult as the model keeps forgetting previously learned knowledge when exposed to the new training examples.

Most of the existing NMT systems rely on the same network to model all domains, which means the same word embedding to represent all the meanings of a word and the same set of parameters to model its depending contexts. This type of configuration in general maximizes the knowledge transfer, but overlooks the specificity of each domain (Koehn and Knowles, 2017). An obvious solution for this problem is to dedicate an individual model to each constituent domain, which is unrealistic in practice as it dramatically increases the number of model parameters. Moreover, the recent success of multilingual applications (Johnson et al., 2017) show that a single NMT model where all parameters are shared can handle translation between hundred of language pairs, suggesting that model capacity may not be the key weakness of the current NMT models to deal with Multi-Domain problems. Thus, the need for a compact architecture with better parameter efficiency is appealing.

We propose the *Factorized Transformer* framework to deal with the multi-domain NMT problem. The Factorized Transformer consists in factorizing partially or fully basic components (embedding, attention and FFN layers) of a conventional Transformer architecture into domain-specific blocks and domain-shared blocks. This dual structure has several advantages: 1) It allows the model to leverage all available data, labeled or unlabeled, to build a generic model at an early stage of domain-agnostic training; 2) Domain singularities could be effectively learned by using domain-specific components and the respective in-domain training data during the stage of domain-aware training. The domain bias issue naturally disappears; 3) Domainspecific components are independently optimized, without any interference between target domains. The original performance of the generic model on un-adapted source domains is also preserved, overcoming the limit of catastrophic forgetting. 4) The design of Factorized Transformer is orthogonal to any data-driven approach, so that the benefit of both approaches can be combined.

Our contributions can be summarized as follows:

- We address the weaknesses of existing NMT systems in multi-domain scenarios by proposing the Factorized Transformer, which separately model domain-shared and domainspecific information via its dual structure.
- We validate our method over a large-scale English to French multi-domain setting. We study 3 variants of Factorized Transformer meeting different requirements of performance and parameter space limitation, our approach outperforms all previously state-ofthe-art multi-domain systems, reaching close to the combined performance of individual fine-tuned models.
- Our proposed architecture enables new perspectives for open domain applications.

2 Related Work

Multi-domain NMT has been an active research area. Prior work in this area can be divided into two main categories: data-driven and model-driven, although they are usually complementary.

Data-Driven Approaches Many researches focus on the exploration of data-driven approaches (van der Wees et al., 2017; Sajjad et al., 2017; Wang et al., 2018). Chu et al. (2017) finetuned the model using the mix of in-domain and out-ofdomain training corpora. Chen et al. (2017) scaled the top-level costs of NMT system according to each training sentence's similarity to the development set. Contrastingly, Farajian et al. (2017) utilized the similarity between each test sentence and the training instances to dynamically set the hyper-parameters of the learning algorithm and update the generic model on the fly. Li et al. (2018) went a step further by adapting a separate model for each sentence to boost the performance. While data-driven approaches are very effective in alleviating the domain bias issue in multi-domain scenarios, they in general require a very careful hyperparameters tuning and cannot reach optimums for all domains simultaneously, unless resort to scaling up brutally the number of models.

The model-driven approaches can be subdivided into two sub-categories:

Soft-Constraints-Based Approaches The subcategory consists in injecting domain information into the model parameters, by the means of sideconstraints, domain embeddings, so as to endow these parameters with domain knowledge, to make them domain-aware. Kobus et al. (2017) added an artificial token to the end of the input sequence to indicate the required target domain and exploited domain as a tag or a feature. Britz et al. (2017) employed discriminators, training objective or GANlike techniques to incorporate domain knowledge into the encoder or decoder. Chu and Dabre (2019) treated text domains as distinct languages in order to use multi-lingual approaches when implementing multi-domain NMT. Zeng et al. (2018) combine source-target domain classifiers and adversarial domain classifier during training. However, since the main model parameters (embeddings, encoder, decoder) remain shared across all domains, the capacity of these methods to deal with the interdomain conflicts might be limited.

Hard-Constraints-Based Approaches involve dedicating extra parameters to directly model domain-specific knowledge. Michel and Neubig (2018) introduces speaker-specific softmax bias to deal with adaptation for a large number of speakers, the idea of parameter factorization is also exploited. Adapter tuning is a recently arisen approach for transfer learning (Rebuffi et al., 2017, 2018; Houlsby et al., 2019; Stickland and Murray, 2019). Each task/domain is equipped with its own set of parameters in order to model and capture domain specificity, which is decoupled among different tasks. Bapna et al. (2019) successfully adapt this approach for domain adaptation and multilingual NMT models.

Our work falls into the second sub-category of the model-driven approaches and we hypothesize that the idea of introducing decoupled domainspecific parameters is crucial. We conduct experiments and analysis in the following sections to validate this hypothesis.

3 Approach

All basic components (embedding, attention and FFN matrices) of a conventional Transformer are factorized into multiple domain-specific blocks (Figure 1), one for each domain (colored ones) and a domain-shared block (white ones), common across all domains.

It's worth to notice that domain information is necessary for both training and inference, which could be obtained via external sources. Nevertheless, the domain prediction is not the main purpose of this work and we suppose in the whole paper, except otherwise mentioned, that domain information is known and passed as input to the model during training and inference.

3.1 Training Curriculum

We first briefly explain the training curriculum before moving to the detailed schemes of factorization, as the former is complementary to the latter and is designed to take advantage of the latter. The training curriculum can be theoretically divided into two stages: an early stage of *domain-agnostic training* and a later stage of *domain-specific training*, even though in practice, it could be achieved in an end-to-end curriculum.

Domain-Agnostic Training aims at building a generic model by sharing the model parameters across all available training domains. Using all available training data is beneficial for the model's overall performance as it allows the model to leverage knowledge from other domains that are related or close to the target domains. For example, the "JRC Acquis" domain (a collection of legislative texts of the European Union) would probably benefit from adding "europarl" domain (a collection of European Parliament texts) training data. Many data weighting schemes exist in the literature, however, this is beyond the scope of this paper and more importantly, the design of Factorized Transformer is orthogonal to any data-driven approach, so that the benefit of both approaches can be combined.

Domain-Specific Training Once the generic model comes to a convergence, the domain-shared parameters of the resulting generic model are then frozen. We unfold all domain-specific components to the number of target domains and initialize them

with the same corresponding matrix trained during the first stage. The specialization step is straightforward: the optimization of each set of domainspecific parameters can operate independently using the respective relevant in-domain data.

As each domain-specific matrix is initialized with the corresponding parameters from the underneath pre-trained network. Therefore, no transition performance degradation is observed along the extra module integration if any. In the case where an additional adaptation layer is involved (Fig 1 (F_6)), we initialize it to a block identity tensor to maintain the exact model performance coming off the domain-agnostic training. This property is of great practical value as it allows the network to adapt directly on top of a set of well-optimized parameters. Similar design can be found in adapter modules: (Rebuffi et al., 2018; Houlsby et al., 2019; Stickland and Murray, 2019), which relies on skip-connection or residual-connection in order to obtain a near identity initialization. Moreover, (Houlsby et al., 2019) observed that if the initialization deviates too far from the identity function, the model may fail to train with adapter modules for transferring BERT style parameters across NLP tasks. However, our proposed Factorized Transformer does not suffer from such problem as it has the exact identity initialization property.

3.2 Factorization Schemes of Basic Components

Throughout this section, we ignore all bias terms, as they may or may not exist depending on the variant/block of the Transformer architecture and also do not add significantly to the parameter count. We first go through some notations before getting into architecture description, d_m refers to the dimension of the model, which is equal to embedding size d_e and hidden size d_h in a conventional Transformer. V refers to the vocabulary size, without loss of generality, we suppose the source side and target side both share the same vocabulary size for the theoretical considerations. d_{filter} refers to the filter dimension used in the FFN layers. h denotes the number of heads used in multi-head attention. N_d represents the number of constituent domains. Finally, we introduce an extra dimension d_{inner} as the inner dimension used for linear factorization that we will explain in the following paragraphs.

Factorization of Embedding Blocks A conventional Transformer network has three wide embed-



Figure 1: Factorization schemes for basic blocks. (F_1) : output embedding, (F_2) : multi-query attention, (F_3) : FFN layer, (F_4) : input embeddings, (F_5) : multi-head attention, (F_6) : FFN layer v2. DF, SF and PA respectively stands for Deep Factorization, Shallow Factorization and Parallel Attention. Different matrix sizes involved in the factorization could be identified by segment length at the top left corner.

ding matrices W_e of dimensions $d_m * V$, which are often tied or partially tied (Press and Wolf, 2016) to reduce model size. NMT models usually require the vocabulary size V to be large, V is of the order of $100 * d_m$. This can easily result in an embedding matrix with millions of parameters, many of which are only updated sparsely during training. We follow the work of Lan et al. (2019) to factorize these blocks (Fig 1 (F_1) and (F_4)). More specifically, for each embedding matrix M_e , we decompose it along an inner dimension d_{inner} (Eq 1): $W_{emb} = W^C \times W_i^S$, where W^C is a shared matrix and W_i is a specific matrix for $i \in 1 \dots N_d$. The advantage of such decomposition is two-fold: First, instead of sharing the same word embedding for all domains, the domain-specific sub-matrices provide a capacity for the model to give a domain-specific meaning to each word embedding. Secondly, from a practical perspective, by using this decomposition, we reduce the embedding parameters from $O(V \times d_m)$ to $O(V \times d_{inner} + N_d \times d_{inner} \times d_m)$. If $d_{inner} \ll d_m$, the factorized form's parameter cost remains inferior to the original embedding block, resulting in better usage of model parameters.

$$\operatorname{EMB}(\boldsymbol{h}, D_d) = \sum_{j=1}^{N_d} \delta_{dj} \operatorname{EMB}_{D_d}(\boldsymbol{h})$$
(1)
where
$$\operatorname{EMB}_{D_d}(\boldsymbol{h}) = W_D^S W^C \boldsymbol{h}$$

Where the weight matrices are of dimension: $W^C \in R^{V \times d_{inner}}, W^S_{j \in 1...N_d} \in R^{d_{inner} \times d_m}$

Factorization of Attention Blocks The factorization of the attention blocks operates differently from the embedding blocks, as each attention block is composed of four relatively small weight matrices W^Q, W^K, W^V, W^O . Within the Multi-Head Attention (MHA) in a conventional Transformer, they are square matrices of the same dimensions d_m^2 . In the case of Multi-Query Attention (MQA) (Shazeer, 2019) instead of multi-head, we share the same key and value sub-matrices for all the heads, the dimensions of matrices W^K, W^V are reduced to $d_m * d_k = d_m^2/h$.

We consider two schemes of introducing domainspecific components. A "full" scheme (Fig 1 (F_5)) which consists in assigning different matrices for each domain for each transformation of W^Q, W^K, W^V, W^O in multi-head style attention, and a "light" scheme (Fig 1 (F_2)) which only parallelizes the relatively small matrices of W^K, W^V of the multi-query style attention. Concretely, if we denote the conventional attention mechanism as follows:

$$MHA(\boldsymbol{h}) = [(H_1(\boldsymbol{h}), \dots, H_h(\boldsymbol{h})]W^O$$
$$H_k(\cdot) = Softmax(\frac{\langle W_k^Q \cdot, W_k^K \cdot \rangle}{\sqrt{d_k}})W_k^V \cdot$$
(2)

where $[\cdot,\ldots,\cdot]$ stands for concatenation and $\langle\cdot,\cdot\rangle$ for dot product.

The factorization of the attention block in the full scheme with multi-head style attention can be written as:

$$MHA_{D_d}(\boldsymbol{h}) = [(H_1^{D_d}(\boldsymbol{h}), \dots, H_h^{D_d}(\boldsymbol{h})]W_{D_d}^O$$
$$H_k^{D_d}(\boldsymbol{\cdot}) = Softmax(\frac{\langle W_{D_d,k}^Q \cdot, W_{D_d,k}^K \cdot \rangle}{\sqrt{d_m/h}})W_{D_d,k}^V \cdot$$
(3)

And in the case of the light scheme with Multi-Query Attention:

$$MQA_{D_d}(\boldsymbol{h}) = [(H_1^{D_d}(\boldsymbol{h}), \dots, H_h^{D_d}(\boldsymbol{h})]W^O$$
$$H_k^{D_d}(\boldsymbol{\cdot}) = Softmax(\frac{\langle W_k^Q \boldsymbol{\cdot}, W_{D_d}^K \boldsymbol{\cdot} \rangle}{\sqrt{d_k}})W_{D_d}^V \boldsymbol{\cdot}$$
(4)

While the latter remains parameter efficient unless $N_d \gg h$, the former significantly increases the model parameters.

Factorization of FFN Blocks FFN blocks are composed of coupled linear matrices joined via a ReLU activation on their amplifying inner dimension d_{filter} . We could perform twice the linear factorization as for case of embedding matrices (Fig 1 (F_3)), or introduce an extra layer of square matrices, one for each domain (Fig 1 (F_6)). In general, few additional parameters are needed for the factorization of the FFN blocks unless $N_d \gg d_{filter}/d_m$

$$FFN(\boldsymbol{h}) = W_2(\max(0, W_1\boldsymbol{h}))$$
(5)

where the weight matrices are of dimension: $W_1 \in R^{d_m \times d_{filter}}, W_2 \in R^{d_{filter} \times d_m}$

The first factorization scheme (Fig 1 (F_3)) for the FFN block can be written as:

$$FFN(\boldsymbol{h}, D_d) = \sum_{j=1}^{N_d} \delta_{dj} FFN_{D_d}(\boldsymbol{h})$$

where $FFN_{D_d}(\boldsymbol{h}) = f_{2,D_d}(\max(0, f_{1,D_d}(\boldsymbol{h})))$
 $f_{i,D_d}(\boldsymbol{h}) = W_{i,D_d}^S W_i^C \boldsymbol{h}$
(6)

where the weight matrices are of dimension: $W_1^C \in R^{d_m \times d_{inner}}, W_{1,j \in 1...N_d}^S \in R^{d_{inner} \times d_{filter}},$ $W_2^C \in R^{d_{filter} \times d_{inner}}, W_{2,j \in 1...N_d}^S \in R^{d_{inner} \times d_m}$ The second factorization scheme (Fig 1 (F₆))

The second factorization scheme (Fig 1 (F_6)) for the FFN block can be formulated as:

$$FFN^{V2}(\boldsymbol{h}, D_d) = \sum_{j=1}^{N_d} \delta_{dj} W^A_{D_d} FFN(\boldsymbol{h}) \quad (7)$$

where $W^A_{j \in 1...N_d} \in R^{d_m \times d_m}$

3.3 Overall Architecture Designs of Factorized Transformer

We consider three architecture designs of Factorized Transformer for multi-domain NMT in this paper, namely Deep Factorization (DF), Shallow Factorization (SF) and Parallel Attention (PA). These designs have been deliberately chosen as extreme cases to provide insights on the limits of the Factorized Transformer, regarding different requirements of performance and parameter space limitation. Other more progressive combination schemes could be also interesting to be investigated depending on the final goal and constraints of applications.

Deep Factorization (DF) We combine the factorization schemes (F_1) , (F_2) , (F_3) , (F_4) , and it's called deep factorization, since factorization is applied to all the main blocks and the combination of domain-shared parameters and domain-specific parameters occur through the whole model. We set the d_{inner} to 280 to obtain the same model capacity as the Transformer base setting for fair comparison.

Shallow Factorization (SF) We rely on the entire original architecture of Transformer to encode domain-shared knowledge as a conventional Transformer, so that we will not suffer from the loss of knowledge transfer capacity compared to the original Transformer. The domain-specific components are plugged into the main architecture as light weight add-on modules. We also duplicate the

Domain	Train	Scale	Dev	Test
OOD	70M	x1	-	-
Emea	338K	x50	1K	1K
Subtitles	34M	x1	1K	1K
News	197K	x100	1K	1K
Iwslt	223K	x100	1K	1K
Jrc	483K	x50	1K	1K
Total	105M	x2	-	-

Table 1: Statistics of training corpora: "OOD" stands for Out-Of-Domain, "Scale" indicates the scale factor for oversampling.

key, value matrices as domain-specific components. It corresponds to the combination of factorization schemes (F_2) and (F_6) in Figure 1.

Parallel Attention (PA) We parallelize all the attention matrices of the original multi-head attention (Vaswani et al., 2017) to boost the model capacity reserved to each domain. This configuration (Fig 1 (F_5)) can be seen as a factorization of the entire network into domain-shared non-attention blocks and domain-specific blocks.

4 **Experiments**

4.1 Experiment settings

Datasets In this paper, we evaluate our proposed method on a 100 million English-French open multi-domain dataset from OPUS corpus¹. It contains sentences from twelve domains including News, Spoken, Laws and Medical etc. We divided the corpora into training, development and test sets. We select five domains of News, Iwslt, Jrc, Emea and Subtitles as evaluation criteria, all data from other domains are considered as outof-domain data and used for training only. Following (Sajjad et al., 2017), we oversampled the low-resource domains to match the same order of size for high resource domains, out-of-domain sentences are not concerned by the oversampling. All sentence pairs are then concatenated and shuffled into a final training data. We tokenize English and French sentences using MOSES script². Byte-pair encoding (Sennrich et al., 2016) is employed in the experiment 50,000 joint pairs, the source and target vocabulary is set to the 50,000 most frequent tokens. Table 1 provides the corpora statistics used in our experiments.

Systems Settings We employ Transformer (Vaswani et al., 2017) as our basis architecture. Six layers are stacked in both the encoder and decoder, and the dimensions of the embedding vectors and all hidden vectors are set to 512. The inner layer of the feed-forward sublayer has the dimension of 2048. We use 8 heads in the multi-head or multi-query attention. The target embedding and the output embedding are shared in our experiments. We use the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.997, \varepsilon = 10^{-9}$ during training. The initial learning rate is 0.0003. The learning rate decay schedule is applied for initial warm up and annealing (Vaswani et al., 2017). During training, each mini-batch contains 4096 tokens and we use a dropout rate of 0.1 on all datasets including attention dropout. During evaluation, we employ lowercase token BLEU (Papineni et al., 2002) as our evaluation metric and use mteval-13a script. In addition, during decoding, we use the beam search algorithm and the beam size is set to 4.

Benchmark Systems We compare our system with multi-domain systems previously reported in the literature, a system is considered as multidomain system if all its parameters can be contained within a unified and deployment-friendly framework. Such candidates are Domain Control (DC) (Kobus et al., 2017) and Target Token Mixing (TTM) (Britz et al., 2017), which are sideconstraint based pioneer works of using domain information for multi-domain training; Multitask Learning (ML) (Britz et al., 2017) method and the Word-level Domain Context (WDC) (Zeng et al., 2018) method both add classifiers to the training so that the network can distinguish mulit-domain contexts; As mentioned in the introduction, adapterbased method is also considered. We use the "bottleneck" Residual Adapters (RA) reported in Bapna et al. (2019) with an inner dimension set to 2048. We re-implement all previously reported RNNbased approach with the Transformer architecture for fair comparison.

We omit any data-driven approach, as it is orthogonal to our approach and can be naturally combined together. We choose a balanced scheme described above as a pretty strong data-mixing baseline, the best system after several preliminary experiments.

4.2 Experimental Results and Analysis

The results of our system are shown at the bottom of Table 2. The performances of benchmark multi-

¹http://opus.nlpl.eu/

²http://www.statmt.org/moses/

SYSTEMS	#P	NEWS	IWSLT	JRC	EMEA	SUB	AVG-5	Δ
Tranformer-base	1x	35.33	41.49	64.20	56.58	30.66	45.65	-
Tranformer-base+finetuning (FT)	5x	35.46	41.63	69.00	61.96	33.42	48.30	+2.65
Domain Control (DC)	1x	36.12	41.47	63.97	56.15	30.97	45.73	+0.08
Target Token Mixing (TTM)	1x	35.97	41.81	64.05	56.04	30.74	45.72	+0.07
Multitask Learning (ML)	1x	34.87	41.72	64.04	56.57	30.35	45.51	-0.14
Word-level Domain Context (WDC)	1x	36.26	41.73	64.54	56.49	30.78	45.96	+0.31
Residual Adapters (RA)	2.3x	35.33	41.49	65.90	59.72	32.31	46.95	+1.30
Factorized Transformer (ours)								
Deep Factorization (DF)	1x	35.92	41.39	66.03 [†]	59.25	32.89 [†]	46.99 [†]	+1.34 [†]
Shallow Factorization (SF)	1.1x	36.38 [†]	42.46 [†]	65.47	58.63	32.34^{\dagger}	47.05 [†]	$+1.40^{\dagger}$
Parallel Attention (PA)	1.8x	35.39	41.69	67.21 [†]	61.70^{\dagger}	33.14 [†]	47.78 [†]	+2.13 [†]

Table 2: Benchmark results on 105 million English to French multi-domain open data. "#P" denotes the scale factor of parameter compared to the baseline. "AVG-5" refers to the average score across the 5 domains"†" indicates the scores of our systems that outperform all other benchmark systems except the combined performance of 5 individual finetuned models.

domain systems are reported at the upper part of Table 2. A standard Transformer base setting is used as baseline for our experiments. It worth to notice that the extensive use of extra out-of-domain general data contributes for the strong performance of the baseline model for general domains, no significant improvement is observed even after finetuning (Luong and Manning, 2015) with in-domain data for News and Iwslt domains. We refer to the average score over the 5 target domains (AVG-5) as multi-domain performance. We also report the combined performance of 5 fully fine-tuned models as the upper bound performance (+2.65 BLEU in average) for Multi-Domain approaches.

Our proposed Factorized Transformer systems clearly outperform the baseline and other multidomain systems in terms of multi-domain performance (AVG-5) as well as individual performance for most settings: our Deep Factorization, Shallow Factorization, Parallel Attention systems respectively yield +1.34, +1.40 and +2.13 BLEU gain over the baseline system. Substantial gains are observed for the domains of JRC (law text), EMEA (medical text) and SUB (subtitles) which have every specific terminologies and syntactic style. No significant improvement is observed for the domains of NEWS and IWSLT, which are still kinds of general domains.

Surprisingly, most of the previous multi-domain techniques, except adapter-based approach, yield very marginal gain over the Transformer baseline in our experiment setting. As all these techniques are re-implemented under the Transformer architecture, we assert that Transformer may have a stronger outof-the-box expressive ability compared to its RNNbased counterparts. Also, all soft-constraint-based



Figure 2: Parameter Efficiency (= Δ / #P from Table 2) for multi-domain benchmark systems.

systems perform better for domains that are closed to general domains (News, Iwslt) with big amount of out-of-domain data than the low-resource and over-sampled ones, which validate the assumption that models with a single shared set of parameters are more likely to be biased toward high resource domains to the detriment of the low-resource ones. Adapter-based system has the closest overall performance, demonstrating the benefit of separating the training process into domain-shared and domainspecific stages with the corresponding shared or domain-specific parameters.

Parameter Efficiency All of our systems demonstrate better parameter efficiency, measured by the ratio between the performance gain and the parameter scale factor (Fig 2).

Impact of Catastrophic Forgetting Our Factorized Transformer can also be used for domain adaptation tasks. One of the main concerns of domain

SYSTEMS	In	Out-Of-	AVG-5	
515112115	SUB	IWSLT	EMEA	
Transformer	30.66	41.49	56.58	45.65
+ Finetune	33.42	38.22	29.49	33.70
+ L2 reg	31.81	39.47	47.82	42.22
+ EWC	31.96	39.25	50.41	43.10
+ Mix-Finetune	31.98	41.23	58.01	46.32
FactorTrans-PA	33.42	41.52	58.59	46.85

Table 3: Benchmark for Domain Adaptation Techniques. The domain SUB is fine-tuned using in-domain data, the results of JRC and NEWS domains are omitted for space reason, which are taken into account in the average score (AVG-5). FactorTrans-PA refers to the Parallel Attention design of our approach using the fine-tuned model as pre-trained model.

Open **Tag-Free** SYSTEMS IT **EMEA** SUB Transformer (no IT) 32.33 56.58 30.66 FactorTrans-PA 32.33 58.59 33.42 + use tag (oracle) + ens-uniform 29.47 53.12 30.76 + ens-soft 31.25 58.21 33.26 + ens-learnable 31.10 58.38 33.41

Table 4: Experimental results for Open-Domain setting. **ens-uniform** refers to the ensemble system with fixed equal weights; **ens-soft**: weight as classifier's output distribution, over the known domains only; **enslearnable**: weight vector tuned over balanced train/dev data from known domains all combined.

adaptation is how to limit the degradation caused by the catastrophic forgetting problem. Table 3 shows the benchmark results between one of our Factorized Transformer system (PA) and some popular domain adaptation techniques. The fine-tuned system achieves the best in-domain performance (Subtitle), however, it suffers from severe catastrophic forgetting problem as its performance in the domain of EMEA is nearly halved. Our Factorized Transformer can operate on top of the fine-tuned system to recover most of the performance drop while preserving the optimal in-domain performance, The resulting system outperforms the fine-tuned system by +13.12 BLEU and the baseline system by +1.20 BLEU in overall performance. Introducing regularization techniques such as L2 (Barone et al., 2017), EWC (Kirkpatrick et al., 2016; Thompson et al., 2019) and mix-finetuning (Chu et al., 2017) can alleviate the drop in the domains of IWSLT, however it limits the performance of in-domain.

5 Towards Open-Domain NMT

In many real-world scenarios, the domain information is unknown at inference time, and even worse, the test inputs may also be out-of-domain, which means the model has never seen data from the same domains during training. For such unknown domains, NMT systems are known to have poor performance, especially adapted ones (Freitag and Al-Onaizan, 2016; Koehn and Knowles, 2017).

Model ensembling is a reasonable approach to deal with unknown domains (Freitag and Al-Onaizan, 2016; Saunders et al., 2019). The compact and unified architecture of Factorized Transformer makes it ideal for this purpose as at each step all domain-specific representations can be computed in parallel and feed-forward to obtain multiple domain-specific word prediction probabilities (logits). We consider in this section the unseen IT domain as a new unknown test domain. The test set is drawn from the GNOME corpus from the OPUS website. Under the open domain paradigm, we do not use any development or training data. We experiment with 3 simple variants of model ensembling based on the Parallel Attention design of our approach (See Table 4 for details). We ensemble all of the 5 adapted domains' output and that of the "general" domain, which corresponds to the base model before any domain-aware training and is more likely to have good performance for unknown domains than its adapted counterparts (Freitag and Al-Onaizan, 2016; Saunders et al., 2019).

The results (Table 4) demonstrate the potential of our Factorized Transformer for open-domain applications: not surprisingly, a naive combination of adapted systems (ens-uniform) result in degradation in all domains. The ens-soft and ens-learnable systems both manage to preserve the in-domain performance for known domains while still performing reasonably well for the unknown IT domain.

6 Conclusion

In this paper, we propose the *Factorized Transformer* framework to overcome the limits of traditional multi-domain NMT approaches in modeling all domain knowledge within a single shared set of parameters. By factorizing wisely the parameters of the Transformer model into domain-shared and domain-specific parts, we significantly improve the model's parameter efficiency and provide new perspectives for open domain applications.

Acknowledgments

We thank the anonymous reviewers for their detailed and constructed comments. This work is supported by National Key R&D Program of China (2018YFB1403202)

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *proceedings of ICLR 2015*.
- Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. 2019. Simple, Scalable Adaptation for Neural Machine Translation. *arXiv e-prints*, page arXiv:1909.08478.
- Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1489–1494.
- Denny Britz, Quoc V. Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017, pages 118–126.
- Boxing Chen, Colin Cherry, George Foster, and Samuel Larkin. 2017. Cost weighting for neural machine translation domain adaptation. In *Proceedings* of the First Workshop on Neural Machine Translation, pages 40–46, Vancouver. Association for Computational Linguistics.
- Chenhui Chu and Raj Dabre. 2019. Multilingual multidomain adaptation approaches for neural machine translation. *CoRR*, abs/1906.07978.
- Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers, pages 385–391.
- 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, WMT 2017, Copenhagen, Denmark, September 7-8, 2017, pages 127–137.
- Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.

- Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. arXiv preprint arXiv:1803.05567.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzkebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, pages 2790–2799.
- Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2016. Overcoming catastrophic forgetting in neural networks. Cite arxiv:1612.00796.
- 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. IN-COMA Ltd.
- Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *CoRR*, abs/1706.03872.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. arXiv eprints, page arXiv:1909.11942.
- Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2018. One sentence one model for neural machine translation. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018), Miyazaki, Japan. European Languages Resources Association (ELRA).
- Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.
- Michael Mccloskey and Neil J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *The Psychology of Learning and Motivation*, 24:104–169.

- Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. *CoRR*, abs/1805.01817.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *ACL*.
- Ofir Press and Lior Wolf. 2016. Using the output embedding to improve language models. *ArXiv*, abs/1608.05859.
- Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems 30*, pages 506–516. Curran Associates, Inc.
- Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2018. Efficient parametrization of multidomain deep neural networks. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 8119–8127.
- Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Yonatan Belinkov, and Stephan Vogel. 2017. Neural machine translation training in a multi-domain scenario. *arXiv preprint arXiv:1708.08712*.
- Danielle Saunders, Felix Stahlberg, Adrià de Gispert, and Bill Byrne. 2019. Domain adaptive inference for neural machine translation. *CoRR*, abs/1906.00408.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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.
- Noam Shazeer. 2019. Fast Transformer Decoding: One Write-Head is All You Need. *arXiv e-prints*, page arXiv:1911.02150.
- Asa Cooper Stickland and Iain Murray. 2019. BERT and pals: Projected attention layers for efficient adaptation in multi-task learning. In *Proceedings* of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, pages 5986–5995.
- Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 2062–2068.

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Rui Wang, Masao Utiyama, Andrew Finch, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2018. Sentence selection and weighting for neural machine translation domain adaptation. *IEEE/ACM Trans. Audio*, *Speech and Lang. Proc.*, 26(10):1727–1741.
- Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark. Association for Computational Linguistics.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.
- Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multidomain neural machine translation with word-level domain context discrimination. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, pages 447–457.