Multi-Domain Neural Machine Translation with Word-Level Adaptive Layer-wise Domain Mixing

Haoming Jiang
Georgia Tech
jianghm@gatech.edu

Chen Liang
Georgia Tech
cliang73@gatech.edu

Chong Wang
ByteDance
chong.wang@bytedance.com

Tuo Zhao
Georgia Tech
tourzhao@gatech.edu

Abstract

Many multi-domain neural machine translation (NMT) models achieve knowledge transfer by enforcing one encoder to learn shared embedding across domains. However, this design lacks adaptation to individual domains. To overcome this limitation, we propose a novel multi-domain NMT model using individual modules for each domain, on which we apply word-level, adaptive and layer-wise domain mixing. We first observe that words in a sentence are often related to multiple domains. Hence, we assume each word has a domain proportion, which indicates its domain preference. Then word representations are obtained by mixing their embedding in individual domains based on their domain proportions. We show this can be achieved by carefully designing multi-head dot-product attention modules for different domains, and eventually taking weighted averages of their parameters by word-level layer-wise domain proportions. Through this, we can achieve effective domain knowledge sharing, and capture fine-grained domain-specific knowledge as well. Our experiments show that our proposed model outperforms existing ones in several NMT tasks.

1 Introduction

Neural Machine Translation (NMT) has made significant progress in various machine translation tasks (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015; Wu et al., 2016). The success of NMT heavily relies on a huge amount of annotated parallel sentences as training data, which is often limited in certain domains, e.g., medical domain. One approach to address this is to explore unparalleled corpora, such as unsupervised machine translation (Lample et al., 2017, 2018). Another approach is to train a multi-domain NMT model and this is the focus of this paper. The simplest way is to build a unified model by directly pooling all training data from multiple domains together, as the languages from different domains often share some similar semantic traits, e.g., sentence structure, textual style and word usages. For domains with less training data, the unified model usually shows significant improvement.

Researchers have proposed many methods for improving multi-domain NMT. Though certain semantic traits are shared across domains, there still exists significant heterogeneity among languages from different domains. For example, Haddow and Koehn (2012) show that for a domain with sufficient training data, a unified model may lead to weaker performance than the one trained solely over the domain; Farajian et al. (2017); Luong et al. (2015); Sennrich et al. (2015a); Servan et al. (2016) also show that to improve the translation performance over certain domains, fine-tuning the unified model is often needed, but at the expense of sacrificing the performance over other domains. This indicates that a unified model might not well exploit the domain-specific knowledge for each individual domain.

To overcome this drawback, two lines of recent research focus on developing new methods by exploiting domain-shared and domain-specific knowledge to improve multi-domain NMT (Britz et al., 2017; Zeng et al., 2018; Tars and Fishel, 2018; Hashimoto et al., 2016; Wang et al., 2017; Chen et al., 2017; Wang et al., 2018; Gu et al., 2019; Chu and Wang, 2018; Dou et al., 2019; Pham et al., 2019; Chu and Dabre, 2019).

One line of research focuses on instance weighting, which assigns domain related weights to different samples during training. For example, Wang et al. (2017) consider sentence weighting and domain weighting for NMT. The sentence weight is determined by the bilingual cross-entropy of each

sentence pair based on the language model of each domain. The domain weight can be modified by changing the number of sentences from that domain in a mini-batch. Chen et al. (2017) propose a cost weighting method, where the weight of each pair of sentences is evaluated by the output probability of a domain classifier on the encoder embedding. Wang et al. (2018) propose a dynamic training method to adjust the sentence selection and weighting during training. We remark that many of these methods are complementary to our proposed model, and can be applied to improve the training of our model.

Another line of research attempts to design specific encoder-decoder architectures for NMT models. For example, Britz et al. (2017) consider domain-aware embedding given by the encoder, and then jointly train a domain classifier, taking the embedding as input to incorporate the domain information. Zeng et al. (2018); Su et al. (2019) further extend their approach by separating the domain-shared and domain-specific knowledge within the embedding. In addition, Zeng et al. (2018) and Shen et al. (2017) propose a maximum weighted likelihood estimation method, where the weight is obtained by word-level domain aware masking to encourage the model to pay more attention to the domain-specific words. The aforementioned methods, however, have a notable limitation: They enforce one single encoder to learn shared embedding across all domains, which often lacks adaptivity to each individual domain.

To better capture domain-shared knowledge beyond shared embedding from a single encoder, we propose a novel multi-domain NMT model using individual modules for each domain, on which we apply word-level, adaptive and layer-wise domain mixing. Our proposed model is motivated by the observation that although every sentence of the training data has a domain label, the words in the sentence are not necessarily only related to that domain. For instance, the word "article" appears in the domains of laws and business. Therefore, we expect the knowledge for translating the word "article" to be shared between these two domains. Our proposed model assigns a context-dependent domain proportion[1] to every word in the sentence. The domain proportions of the words can be naturally integrated into the Transformer model for capturing domain-shared/specific knowledge, as the multi-head dot-product attention mechanism is applied at the word-level. Specifically, we carefully design multi-head dot-product attention modules for different domains, and eventually mix these modules by taking weighted averages of their parameters by their layer-wise domain proportions.

Compared with existing models, ours has the following two advantages:

- Our proposed model is more powerful in capturing the domain-specific knowledge, as we design multiple dot-product attention modules for different domains. In contrast, existing models rely on one single shared encoder, and then one single unified translation model is applied, which often cannot adapt to each individual domain very well.

- Our proposed model is more adaptive in the process of domain knowledge sharing. For common words across domains, their domain proportions tend to be uniform, and therefore can significantly encourage knowledge sharing. For some words specific to certain domains, their domain proportions tend to be skewed, and accordingly, the knowledge sharing is encouraged only within the relevant domains. For example, the word "article" appears less in the medical domain than the domains of laws and business. Therefore, the corresponding domain proportion tends to favor the domains of laws and business more than the medical domain.

We evaluate our proposed model in several multi-domain machine translation tasks, and the empirical results show that our proposed model outperforms existing ones and improves the translation performance for all domains.

The rest of the paper is organized as follows: Section 2 introduces the background; Section 3 describes our proposed model in detail; Section 4 presents numerical experiments on EN-DE, EN-FR and ZH-EN datasets; Section 5 discusses the connection to word disambiguation.

2 Background

Neural Machine Translation (NMT) directly models the conditional distribution of the translated sentence $\mathbf{y} = (y_1, ..., y_\ell)$ given a source sentence $\mathbf{x} = (x_1, ..., x_\ell)$[2]. The conditional probability density function $p(\mathbf{y}|\mathbf{x})$ is parameterized by an encoder-decoder neural network: The encoder

[1] A word actually has multiple domain proportions at different layers of our model. See more details in Section 3

[2] Here we assume that we have applied padding to all sentences, and therefore, they are all of the same length.

encodes the source sentence into a sequence of hidden representations $\mathcal{H}(\mathbf{x}) = (h_1, ..., h_n)$, and the decoder generates target sentence one token at a time using these intermediate representations. More specifically, the decoder usually contains a recursive structure for computing $p(y_t|y_{<t}, \mathbf{x})$ by

$$p(y_t|y_{<t}, \mathbf{x}) = \mathcal{F}(\mathcal{G}_t, \mathcal{H}(\mathbf{x}), y_{t-1}),$$

where $\mathcal{G}_t$ denotes the hidden representation of the decoder for the t-th position of the sequence, and $\mathcal{F}$ denotes a multi-layered network that outputs the probability of y_t. Notice that $\mathcal{G}_t$ is generated by the $\mathcal{G}_{t-1}, \mathcal{H}(\mathbf{x})$, and the previous word y_{t-1}. Given N pairs of source/target sequences denoted by $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$, we train the NMT model by minimizing the cross-entropy loss as follows,

$$\min_{\mathcal{H},\mathcal{G},\mathcal{F}} L_{\text{gen}} = \frac{1}{n}\sum_{i=1}^{n} -\log p(\mathbf{y}_i|\mathbf{x}_i)$$

where $p(\mathbf{y}_i|\mathbf{x}_i) = \prod_{t=1}^{m} p(y_{i,t}|y_{i,<t}, \mathbf{x}_i)$.

Transformer is one of the most popular NMT models (Vaswani et al., 2017; Tubay and Costa-jussà, 2018; Devlin et al., 2018). The encoder and decoder in Transformer contain stacked self-attention and point-wise, fully connected layers without any explicit recurrent structure, which is different from existing RNN-based NMT models.

Specifically, Vaswani et al. (2017) propose a new attention function using the scaled dot-product as the alignment score, which takes the form,

$$\text{Attention}(Q, K, V) = \text{softmax}\Big(\frac{QK^\top}{\sqrt{d}}\Big)V, \quad (1)$$

where $Q, K, V \in \mathbb{R}^{\ell \times d}$ are the vector representations of all the words in the sequences of queries, keys and values accordingly. For the self-attention modules in the encoder and decoder, $Q = K = V$; For the attention module that takes into account the encoder and the decoder sequences, Q is different from the sequence represented by V and K.

Based on the above attention function in (1), Vaswani et al. (2017) further develop a multi-head attention module, which allows the NMT model to jointly attend to information from different representations at different positions. In particular, we consider a multi-head attention module with m heads. For the i-th head H_i, three point-wise linear transformations $W_{i,Q}$, $W_{i,K}$, $W_{i,V} \in \mathbb{R}^{d \times d/m}$ are first applied to the input Q, K and V, respectively, and then the scaled dot-product attention

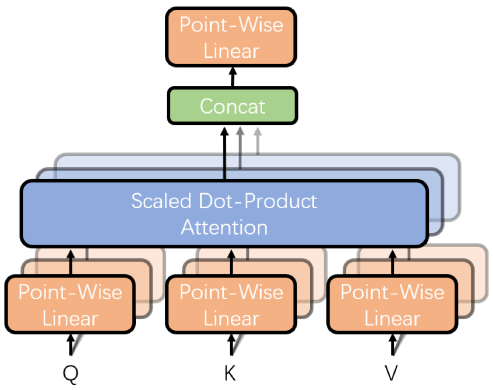

Figure 1: Multi-head Scaled Dot-Product Attention.

is applied: Let $\widetilde{Q}_i = QW_{i,Q}$, $\widetilde{K}_i = KW_{i,K}$ and $\widetilde{V} = VW_{i,V}$,

$$H_i = \text{Attention}(\widetilde{Q}_i, \widetilde{K}_i, \widetilde{V}_i). \quad (2)$$

Eventually, the final output applies a point-wise linear transformation $W_O \in \mathbb{R}^{d \times d}$ to the concatenation of the output from all heads:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(H_1, ..., H_m)W_O.$$

An illustrative example of the multihead attention architecture is provided in Figure 1.

In addition to the above multi-head attention modules, each layer in the encoder and decoder in Transformer contains a point-wise two-layer fully connected feed-forward network.

3 Model

We present our Transformer-based multi-domain neural machine translation model with word-level layer-wise domain mixing.

3.1 Domain Proportion

Our proposed model is motivated by the observation that although every sentence in the training data has a domain label, a word in the sentence does not necessarily only belong to that single domain. Therefore, we assume that every word in the vocabulary has a domain proportion, which indicates its domain preference. Specifically, given the embedding $x \in \mathbb{R}^d$ of a word, k domains and $R \in \mathbb{R}^{k \times d}$, our model represents the domain proportion by a smoothed softmax layer as follows,

$$\mathcal{D}(x) = (1 - \epsilon) \cdot \text{softmax}(Rx) + \epsilon/k,$$

where $\epsilon \in (0, 1)$ is a smoothing parameter to prevent the output of $\mathcal{D}(x)$ from collapsing towards 0 or 1. Specifically, setting ϵ as a large value encourages the word to be shared across domains.

3.2 Word-Level Adaptive Domain Mixing

In our proposed model, each domain has its own multi-head attention modules. Recall that the point-wise linear transformations in the multi-head attention module $W_{i,Q}$'s, $W_{i,K}$'s, $W_{i,V}$'s and W_O are applied to each word separately and identically, as shown in Figure 2. Therefore, we can naturally

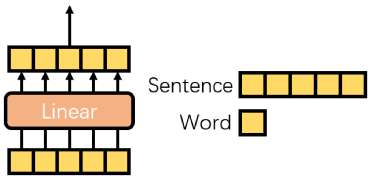

Figure 2: The Point-wise Linear Transformations are applied at the word-level.

integrate the domain proportions of the words with these multi-head attention modules. Specifically, we take the weighted averaging of the linear transformation based on the domain proportion $\mathcal{D}(x)$. For example, we consider the point-wise linear transformations $\{W_{i,Q,j}\}_{j=1}^{k}$ on the t-th word of the input, Q_t, of all domains. The mixed linear transformation can be written as

$$\overline{Q}_{i,t} = \sum_{j=1}^{k} Q_t^\top W_{i,Q,j} \mathcal{D}_{Q,j}(Q_t),$$

where $\mathcal{D}_{Q,j}(Q_t)$ denotes the j-th entry of $\mathcal{D}_Q(Q_t)$, and $\mathcal{D}_Q$ is the domain proportion layer related to Q. Then we only need to replace $\widetilde{Q}_i$ in (2) with

$$[\overline{Q}_{i,1}, ..., \overline{Q}_{i,n}].$$

An illustrative example is presented in Figure 3. For other linear transformations, we applied the domain mixing scheme in the same way. We re-

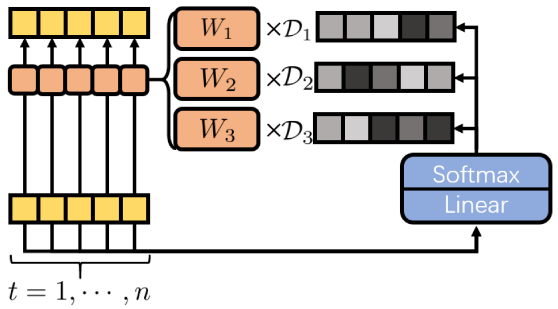

Figure 3: Word-level mixing with 3 domains. For simplicity, we omit the subscripts Q, i.

mark that the Transformer model, though does not have any explicit recurrent structure, handles the sequence through adding additional positional embedding for each word (in conjunction with sequential masking). Therefore, if a word appears in different positions of a sentence, its corresponding embedding is different. This indicates that the domain proportions of the same word can also be different across positions. This feature makes our model more flexible, as the same word in different positions can carry different domain information.

3.3 Layer-wise Domain Mixing

Recall that the Transformer model contains multiple multi-head attention modules/layers. Therefore, our proposed model inherits the same architecture and applies the word-level domain mixing to all these attention layers. Since the words have different representations at each layer, the corresponding domain proportions at each layer are also different, as shown in Figure 4. In addition to the multi-head attention layers, we also apply similar word-level domain mixing to the point-wise two-layer fully connected feed-forward network.

The layer-wise domain mixing allows the domain proportions to be context dependent. This is because the domain proportions are determined by the word embedding, and the word embedding at top layers is essentially learnt from the representations of all words at bottom layers. As a result, when the embedding of a word at some attention layer is already learned well through previous layers (in the sense that it contains sufficient contextual information and domain knowledge), we no longer need to borrow knowledge from other domains to learn the embedding of the word at the current layer. Accordingly, the associated domain proportion is expected to be skewed and discourages knowledge sharing across domains. This makes the process of knowledge sharing of our model more adaptive.

3.4 Training

Recall that $\mathcal{H}$ denotes the encoder, $\mathcal{F}$ denotes the decoder, and $\mathcal{D}$ denotes the domain proportion. Define $\Theta = \{\mathcal{F}, \mathcal{H}, \mathcal{D}\}$. The proposed model can be efficiently trained by minimizing a composite loss function defined as follows,

$$L^* = L_{\text{gen}}(\Theta) + L_{\text{mix}}(\Theta),$$

where $L_{\text{gen}}(\Theta)$ denotes the cross-entropy loss over the training data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n}$, and $L_{\text{mix}}(\Theta)$ denotes the cross entropy loss over the words/domain (hard) labels.

For $L_{\text{mix}}(\Theta)$, the domain labels are obtained from the training data. Specifically, for all words

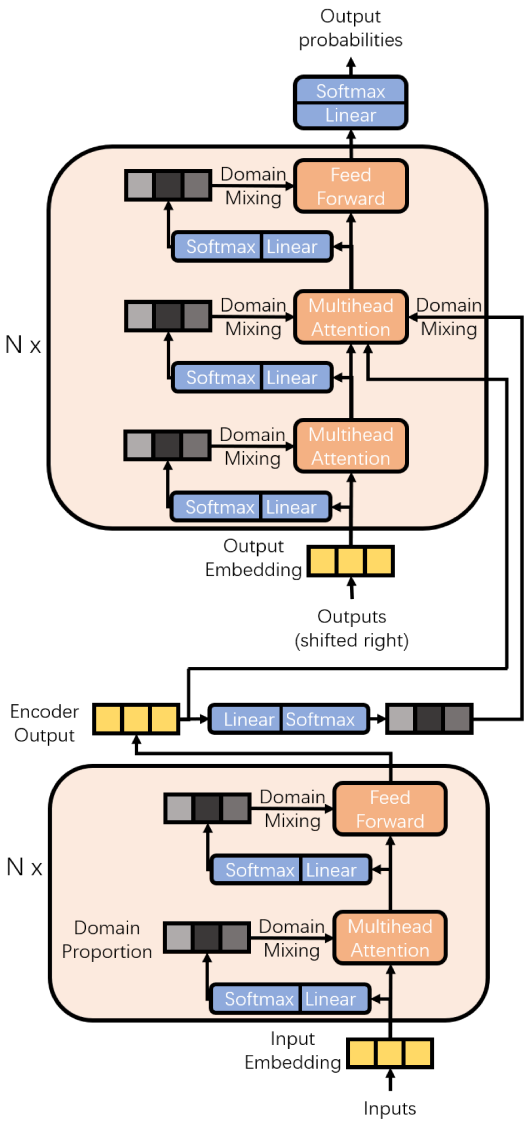

Figure 4: Illustration of Our Multi-domain NMT Model: Normalization and residual connection are omitted for simplicity. For all other detail, please refer to Vaswani et al. (2017).

in a sentence belonging to the J-th domain, we specify their domain hard labels as J. Then given the embedding x of a word, we compute the cross entropy loss of its domain proportion $\mathcal{D}(x)$ as $-\log(\mathcal{D}_J(x))$. Accordingly, $L_{\text{mix}}(\Theta)$ is the sum of the cross entropy loss over all such pairs of word/domain label of the training data.

4 Experiment

We conduct experiments on three different machine translation tasks:

• **English-to-German**. We use a dataset from two domains: News and TED. We collect the News domain data from Europarl (Koehn, 2005) and the TED domain data from IWLST (Cettolo et al., 2014).

• **English-to-French** We use a dataset containing two domains: TED and Medical domain. We collect TED domain data from IWLST (Cettolo et al., 2017) and medical domain data from Medline (Yepes et al., 2017).

• **Chinese-to-English** We use a dataset containing four domains: News, Speech, Thesis and Laws. We collect the Laws, Speech, and Thesis data from UM-Corpus (Tian et al.), and the News data from LDC (Consortium, 1992). The translation from Chinese-to-English is inherently difficult. The four-domains setting makes it even more challenging. This dataset is also used in Zeng et al. (2018).

The sizes of training, validation, and testing sets for different language pairs are summarized in Table 1. We tokenize English, German and French sentences using MOSES script (Koehn et al., 2007) and perform word segmentation on Chinese sentences using Stanford Segmenter (Tseng et al., 2005). All sentences are then encoded using byte-pair encoding (Sennrich et al., 2015b). We evaluate the performance using two metrics: BLEU (Papineni et al., 2002) and perplexity following the default setting in fairseq with beam search steps of 5.

Language	Domain	Train	Valid	Test
EN-DE	News	184K	18K	19K
	TED	160K	7K	7K
EN-FR	TED	226K	10K	10K
	MEDICAL	516K	25K	25K
ZH-EN	Laws	219K	600	456
	News	300K	800	650
	Speech	219K	600	455
	Thesis	299K	800	625

Table 1: The numbers of sentences in the datasets.

4.1 Baselines

Our baselines include the Transformer models trained using data from single and all domains. We also include several domain aware embedding based methods, which train the embedding of the encoder along with domain information.

• **Multitask Learning (MTL)** proposed in Britz et al. (2017) uses one sentence-level domain classifier to train the embedding. Note that their classifier is only used to predict the domain, while our model

uses multiple word-level domain classifiers to obtain the domain proportions for different layers (further used for domain mixing).

• **Adversarial Learning (AdvL)** proposed in Britz et al. (2017) is a variant of MTL, which flips the gradient before it is back-propagated into the embedding. This encourages the embedding from different domains to be similar.

• **Partial Adversarial Learning (PAdvL)** To combine the advantages of the above two methods, we split the embedding into half of multitask part and half of adversarial part.

• **Word-Level Domain Context Discrimination (WDC)** Zeng et al. (2018) integrates MTL and AdvL with word-level domain contexts. This method requires the dimension of the embedding to be doubled and, thus, is not directly applicable in Transformer. We use a point-wise linear transformation to reduce the dimension.

Moreover, Zeng et al. (2018) consider the word-level domain aware weighted loss (**WL**). Specifically, they assign a domain-aware attention weight β_j to the j-th position in the output sentence, and the corresponding weighted loss is:

$$L_{\text{gen}} = -\frac{1}{n}\sum_{j=1}^{n}(1+\beta_j)\log p(y_j|\mathbf{x}, y_{<j}).$$

Here β_j is obtained by an attention based domain classifier built upon the last hidden layer.

4.2 Details of Our Implementation

All of our experiments are conducted under fairseq (Ott et al., 2019) environment. We follow the fairseq re-implementation of 12-layer Transformer designed for IWLST data. Specifically, the embedding dimension is 512 for both the encoder and decoder, the number of heads is 4, and the embedding dimension in the feed-forward layer is 1024. Such a model is actually larger than the base model in Vaswani et al. (2017) (76M vs. 65M parameters). Notice that, the number of parameters of the mixing model is k times larger (k is the number of domains). For a fair comparison, all baselines are tested using both the above model and an enlarged model, which has $\sqrt{k}$ times larger embedding dimension (so the weight matrices are k times larger). The enlarged model and the mixing model has the same number of parameters. The presented baseline results are the best of the two. In terms of the optimization, we follow the training recipe provided by fairseq. Specifically, we use Adam (Kingma and Ba, 2014) with $\beta_1 = 0.9, \beta_2 = 0.98$ with a weight decay parameter of 10^{-4}. The learning rate follows the inverse square root schedule (Vaswani et al., 2017) with warm-up steps of 4000, initial warm-up learning rate of 10^{-7}, and the highest learning rate of 5×10^{-4}. For effective training, L_{gen} is replaced by a label-smoothing cross-entropy loss with a smoothing parameter of 0.1 (Szegedy et al., 2016).

For our domain mixing methods, we set the smoothing parameter ϵ of the domain proportion as 0.05. Besides applying domain mixing to both the encoder and decoder (**E/DC**), we consider applying domain mixing to only the **Encoder**. The domain proportion layers $\mathcal{D}$ are only used for estimating the domain proportion and should not intervene in the training of the translation model. So the gradient propagation is cut off between the Transformer and the domain proportion as Figure 5 shows. More discussion about the training procedure can be found in Section 4.6.

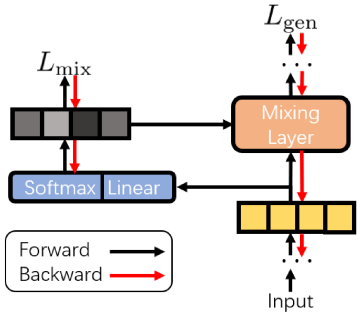

Figure 5: Computational graph for training the domain proportion layers.

4.3 Experimental Results

Table 2 shows the BLEU scores of the baselines and our domain mixing methods for English-to-German translation. As can be seen, our methods outperform the baselines on both domains. Notice that, our baseline method achieves 29.09 BLEU when training and testing on TED domain only, where Liu et al. (2019) only achieves 28.56 with the same training/testing data, the codebase (i.e., fairseq), and the network structure. This indicates that our reimplemented baseline is rather strong.

We also compare the perplexity on the validation set in Figure 6. As can be seen, our domain mixing methods converge faster than the baselines and all methods converge after 50 epochs. We also observe that the baselines get stuck at plateaus at the early

Method	News	TED
Direct Training		
News	26.09	6.15
TED	4.90	29.09
News + TED	26.06	28.11
Embedding based Methods		
MTL	26.90	29.27
AdvL	25.68	27.46
PAdvL	27.06	29.49
WDC + WL	27.25	29.43
Our Domain Mixing Methods		
Encoder	**27.78**	30.30
Encoder + WL	27.67	30.11
E/DC	27.58	**30.33**
E/DC + WL	27.55	30.22

Table 2: English-to-German.

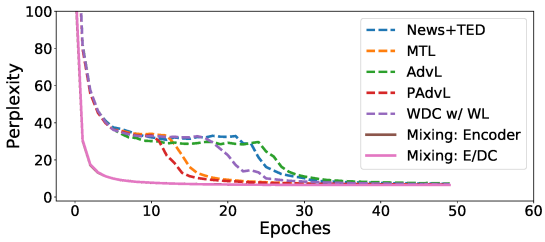

Figure 6: Perplexity v.s. Number of epochs for English-to-German.

stage of training. The possible reason is that their training enforces one unified model to fit data from two different domains simultaneously, which is computationally more difficult.

Table 3 shows the BLEU scores of the baselines and our domain mixing methods for English-to-French translation. Note that though the data from the Medical and TED domains are slightly imbalanced (about 1:2.5), our methods can still outperform the baselines on both domains.

Method	TED	Medical
Direct Training		
TED	28.22	7.32
Medical	7.03	53.73
Medical + TED	39.21	53.40
Embedding based Methods		
MTL	39.14	53.37
AdvL	39.54	53.46
PAdvL	39.56	53.23
WDC + WL	39.79	53.85
Our Domain Mixing Methods		
Encoder	40.30	54.05
Encoder + WL	40.43	54.14
E/DC	40.52	54.28
E/DC + WL	**40.60**	**54.39**

Table 3: English-to-French.

Table 4 shows the BLEU scores of the baselines and our domain mixing methods for Chinese-to-English translation. As can be seen, our methods outperform the baselines on all domains except Thesis. We remark that the translation for the Thesis domain is actually very difficult, and all methods obtain poor performance.

Method	Laws	News	Speech	Thesis
Direct Training				
Laws	51.98	3.80	2.38	2.64
News	6.88	31.99	8.12	4.17
Speech	3.33	4.90	18.63	3.08
Thesis	5.90	5.55	4.77	11.06
Mixed	48.87	26.92	16.38	12.09
Embedding based Methods				
MTL	49.14	27.15	16.34	11.80
AdvL	48.93	26.51	16.18	12.08
PAdvL	48.72	27.07	15.93	**12.23**
WDC + WL	42.16	25.81	15.29	10.14
Our Domain Mixing Methods				
Encoder	50.21	27.94	16.85	12.03
Encoder + WL	50.11	27.48	16.79	11.93
E/DC	**50.64**	**28.48**	17.41	11.71
E/DC + WL	50.04	28.17	**17.60**	11.59

Table 4: Chinese-to-English.

Moreover, we find that for Chinese-to-English task, all our baselines are sensitive to the architecture of the Transformer. Their training will fail, if we place the layer normalization at the end of each encoder and decoder layer (as Vaswani et al. (2017) suggest). Therefore, we move the layer normalization to their beginnings. Surprisingly, our domain mixing methods are very stable regardless of the position of the layer normalization. More details can be found in Table 8 of Appendix A.

4.4 Ablation Study

We further shows that the performance gains are from the domain mixing methods, instead of from the new model architecture design. Table 5 shows the BLEU scores with and without using domain labels under the same network structure and the same number of parameters as in the domain mixing methods. The only difference is that we remove domain label to guide the training of domain proportion, i.e., only L_{gen} is used in the training loss, and L_{mix} is removed. Training without domain labels shows a slight improvement over baseline, but is still significantly worse than our proposed method for most of the tasks. Therefore, we can conclude that our proposed domain mixing approach indeed improves performance.

4.5 Visualizing Domain Proportions

To further investigate our domain mixing methods, we plot the domain proportions of the word em-

Method	Direct Training	w/o DL	with DL (Ours)
English-to-Germany			
News	26.06	26.25	**27.78**
TED	28.11	28.27	**30.30**
English-to-French			
TED	39.21	39.39	**40.30**
Medical	53.40	53.33	**54.05**
Chinese-to-English			
Laws	48.87	48.96	**50.21**
News	26.92	27.02	**27.94**
Speech	16.38	16.15	**16.85**
Thesis	**12.09**	12.03	12.03

Table 5: BLEU Scores with and without domain labels (DL) under equal model capacity.

bedding at different layers. A uniform proportion, e.g., $(0.5, 0.5)$, is encouraging knowledge sharing across domains, while a skewed proportion, e.g., $(0.1, 0.9)$, means there is little knowledge to share across domains. Figure 7 illustrates how the knowledge sharing is controlled via the domain proportion. The selected sentence is from the English-to-French task, containing TED and Medical domains. Specifically, we observe :

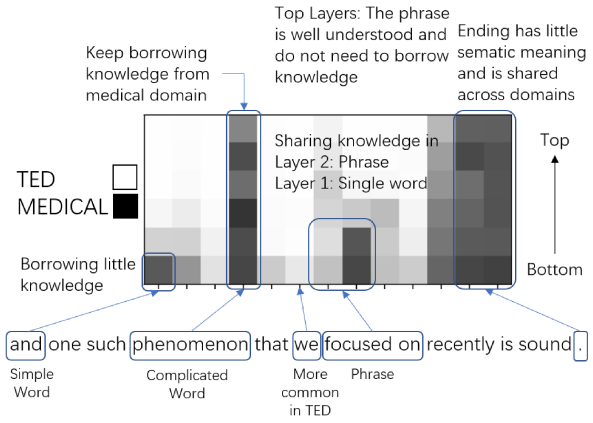

Figure 7: Domain proportion of a sentence from the TED domain for English-to-French task. The domain proportion is extracted from all layers of the encoder.

- The domain proportions of different words at different layers have various patterns.

- At the bottom layers, the domain proportion of a word is closely related to its frequency of occurrence.

- Some words with simple semantic meanings do not need to borrow much knowledge from other domains, e.g., *and*; Some other words need to borrow knowledge from other domains to better understand their own semantic meaning. For example, the word *phenomenon* keeps borrowing/sharing knowledge from/to the medical domain at every layer.

- The ending of the sentence only conveys a stopping signal, and thus is shared across all domains.

- The domain proportions at the bottom layers tend to be more diverse, while those at the top layers tend to be more skewed, as shown in Figure 8 for English-to-German task.

- The domain proportions of the decoder tend to be more skewed than those of the encoder, which demonstrates little knowledge sharing. Figure 9 shows the histograms of word-level domain proportions at different layers in both the encoder and decoder. This might explain why the mixing decoder only contributes limited performance gain for the English-to-German task.

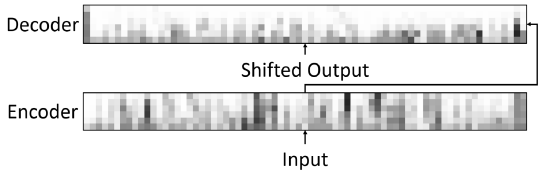

Figure 8: Domain proportions of a sentence pair for English-to-German task. White represents the News domain and black represents the TED domain. The domain proportions of both the encoder (bottom) and the decoder (top) are presented.

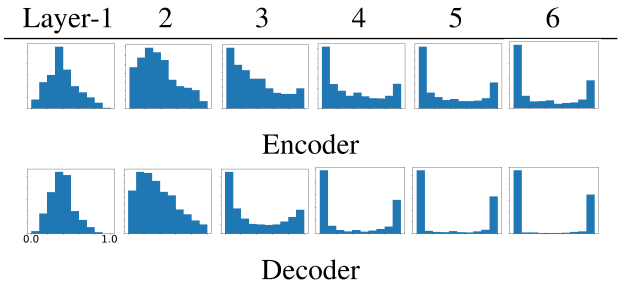

Figure 9: Histograms of the domain proportions of each layer in our domain mixing model for English-to-German Task. Within each histogram, 0 means pure News domain, and 1 means pure TED domain.

4.6 Combining Domain Mixing with Domain Aware Embedding

The embedding based methods can be naturally combined with our domain mixing methods. As we mentioned in 4.2, the domain proportion is trained solely, meaning gradient does not propagate between the domain proportion layers $\mathcal{D}$ and

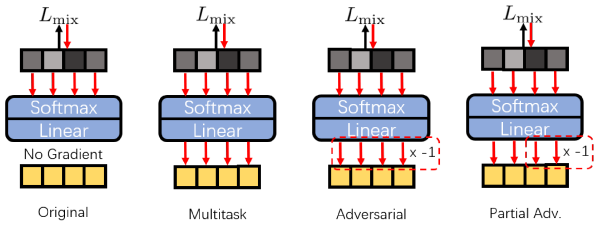

Figure 10: Back-propagation for different embedding based methods.

the Transformer. The computation of the gradient, on the other hand, is the key to combining two methods. Specifically, we encourage the embedding to be domain aware via MTL, AdvL and PAdvL, where we use the domain proportion layers to guide the training of the embedding. Figure 10 illustrates the back-propagation under different methods. Table 6 shows the performance for Chinese-to-English task under this setting. Here we consider applying domain mixing only to the encoder as the baseline. As can be seen, by applying appropriate domain aware embedding, the performance can be further improved.

Method	Laws	News	Speech	Thesis
Encoder	50.21	27.94	16.85	12.03
+MTL	49.15	26.82	15.72	11.93
+Adv	50.18	27.72	**16.99**	**12.16**
+PAdvL	49.01	26.63	16.06	12.15
+Multitask + WL	48.75	26.78	16.53	12.11
+Adv + WL	**50.24**	**28.21**	16.98	12.00
+PAdv + WL	48.87	26.86	16.14	11.89

Table 6: BLEU Scores of Domain Mixing + Domain Aware Embedding for Chinese-to-English Task

5 Discussions

One major challenge in multi-domain machine translation is the word ambiguity in different domains. For example, the word "article" has different meanings in the domains of laws and media. When translating "article" into Chinese, the translated words are "条款" and "文章", meaning a separate clause of a legal document and a piece of writing. Our proposed word-level layer-wise domain mixing approach tends to reduce the word ambiguity. As mentioned in Section 3.3, our model extracts different representations of each word from contexts at different layers. Accordingly, the domain proportion of each word evolves from bottom to top layers, and can eventually help identify the corresponding domains.

Laws	"**Article** 37 The freedom of marriage ..." "第三十七条:婚姻的自由..."
Media	"... working on an **article** about the poems ..." "... 正在写一篇诗的文章 ..."

Table 7: The ambiguity of "articles".

Moreover, as mentioned in Section 3.2, the positional embedding also contributes to the word disambiguation in multi-domain translation. For example, in the law domain, we find that "article" often appears at the beginning of a sentence, while in the media domain, the word "article" may appear in other positions. Therefore, varying domain proportions for different positions can help with word disambiguation.

We remark that word disambiguation across domains actually requires $\mathcal{D}(x)$ to be powerful for predicting the domain of the word. However, a powerful $\mathcal{D}(x)$ tends to yield skewed domain proportions and is not flexible enough for domain knowledge sharing. To trade off between strength and flexibility of $\mathcal{D}(x)$, the smoothing parameter ϵ of $\mathcal{D}(x)$ (see Section 3.1) needs to be properly set.

6 Conclusions

We present a novel multi-domain NMT with word-level layer-wise domain mixing, which can adaptively exploit the domain knowledge. Unlike the existing work, we construct multi-head dot-product modules for each domain and then combine them by the layer-wise domain proportion of every word. The proposed method outperforms the existing embedding based methods. We also show mixing method can be combined with embedding based methods to make further improvement.

Moreover, we remark that our approach can be extended to other multi-domain or multi-task NLP problems.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Niehues Jan, Stüker Sebastian, Sudoh Katsuitho, Yoshino Koichiro, and Federmann Christian. 2017.

Overview of the iwslt 2017 evaluation campaign. In *International Workshop on Spoken Language Translation*, pages 2–14.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th iwslt evaluation campaign, iwslt 2014. In *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*, page 57.

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.

Chenhui Chu and Raj Dabre. 2019. Multilingual multi-domain adaptation approaches for neural machine translation. *arXiv preprint arXiv:1906.07978*.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. *arXiv preprint arXiv:1806.00258*.

The Linguistic Data Consortium. 1992. The linguistic data consortium description.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Zi-Yi Dou, Junjie Hu, Antonios Anastasopoulos, and Graham Neubig. 2019. Unsupervised domain adaptation for neural machine translation with domain-aware feature embeddings. *arXiv preprint arXiv:1908.10430*.

M Amin Farajian, Marco Turchi, Matteo Negri, Nicola Bertoldi, and Marcello Federico. 2017. Neural vs. phrase-based machine translation in a multi-domain scenario. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 280–284.

Shuhao Gu, Yang Feng, and Qun Liu. 2019. Improving domain adaptation translation with domain invariant and specific information. *arXiv preprint arXiv:1904.03879*.

Barry Haddow and Philipp Koehn. 2012. Analysing the effect of out-of-domain data on smt systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 422–432. Association for Computational Linguistics.

Kazuma Hashimoto, Akiko Eriguchi, and Yoshimasa Tsuruoka. 2016. Domain adaptation and attention-based unknown word replacement in chinese-to-japanese neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 75–83.

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.

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

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2019. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

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 NAACL-HLT 2019: Demonstrations*.

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*, pages 311–318. Association for Computational Linguistics.

Minh Quang Pham, Josep-Maria Crego, Jean Senellart, and François Yvon. 2019. Generic and specialized word embeddings for multi-domain machine translation.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Christophe Servan, Josep Crego, and Jean Senellart. 2016. Domain specialization: a post-training domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06141*.

Yan Shen, Dahlmann Leonard, Petrushkov Pavel, Hewavitharana Sanjika, and Khadivi Shahram. 2017. Word-based domain adaptation for neural machine translation. In *Proceedings of the 15th International Workshop on Spoken Language Translation*, pages 31–38.

Jinsong Su, Jiali Zeng, Jun Xie, Huating Wen, Yongjing Yin, and Yang Liu. 2019. Exploring discriminative word-level domain contexts for multi-domain neural machine translation. *IEEE transactions on pattern analysis and machine intelligence*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Sander Tars and Mark Fishel. 2018. Multi-domain neural machine translation. *arXiv preprint arXiv:1805.02282*.

Liang Tian, Derek F Wong, Lidia S Chao, Paulo Quaresma, and Francisco Oliveira. Um-corpus: A large english-chinese parallel corpus for statistical machine translation.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter for sighan bakeoff 2005. In *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*.

Brian Tubay and Marta R Costa-jussà. 2018. Neural machine translation with the transformer and multi-source romance languages for the biomedical wmt 2018 task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 667–670.

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 Transactions on Audio, Speech, and Language Processing*, 26(10):1727–1741.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488.

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*.

Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Karin Verspoor, Ondrej Bojar, Arthur Boyer, Cristian Grozea, Barry Haddow, Madeleine Kittner, Yvonne Lichtblau, et al. 2017. Findings of the wmt 2017 biomedical translation shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 234–247.

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 *Conference on Empirical Methods in Natural Language Processing*, pages 447–457.

A Complementary Experiments – Chinese to English

Experiment results of the original Transformer, where layer normalization is at the end each layer.

Method	Laws	News	Spoken	Thesis
Laws	10.37	0.45	0.27	0.27
News	0.39	5.12	0.91	0.57
Spoken	0.70	1.11	6.19	0.83
Thesis	0.63	0.25	0.16	1.24
Mixed	5.45	4.09	2.67	1.85
Multitask	6.16	3.83	1.91	1.53
Adversarial	5.93	3.38	1.85	1.37
PAdv	6.58	3.90	2.32	1.80
WDC. w/ WL	7.13	3.87	2.45	1.88
Our Proposed Mixing Method				
Encoder	50.16	27.61	16.92	11.85
+ Decoder	50.45	28.15	17.45	11.62

Table 8: Chinese to English

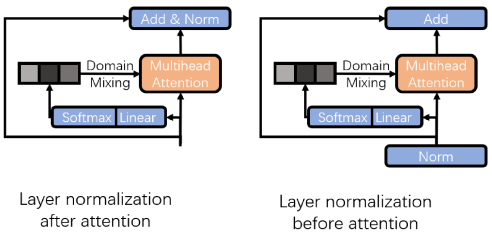

Figure 11: Two variants of layer normalization