Go From the General to the Particular: Multi-Domain Translation with Domain Transformation Networks

Yong Wang[*]
The University of Hong Kong
wangyong@eee.hku.hk

Longyue Wang
Tencent AI Lab
vinnylywang@tencent.com

Shuming Shi
Tencent AI Lab
shumingshi@tencent.com

Victor O.K. Li
The University of Hong Kong
vli@eee.hku.hk

Zhaopeng Tu
Tencent AI Lab
zptu@tencent.com

Abstract

The key challenge of multi-domain translation lies in simultaneously encoding both the general knowledge shared across domains and the particular knowledge distinctive to each domain in a unified model. Previous work shows that the standard neural machine translation (NMT) model, trained on mixed-domain data, generally captures the general knowledge, but misses the domain-specific knowledge. In response to this problem, we augment NMT model with additional *domain transformation networks* to transform the general representations to domain-specific representations, which are subsequently fed to the NMT decoder. To guarantee the knowledge transformation, we also propose two complementary supervision signals by leveraging the power of knowledge distillation and adversarial learning. Experimental results on several language pairs, covering both balanced and unbalanced multi-domain translation, demonstrate the effectiveness and universality of the proposed approach. Encouragingly, the proposed unified model achieves comparable results with the fine-tuning approach that requires multiple models to preserve the particular knowledge. Further analyses reveal that the domain transformation networks successfully capture the domain-specific knowledge as expected.[1]

Introduction

In multi-domain translation, a unified neural machine translation (NMT) model is expected to provide high quality translations across a wide range of diverse domains. The main challenge of multi-domain translation lies in learning a unified model that simultaneously 1) exploits the *general knowledge* shared across domains, and 2) preserves the *particular knowledge* that represents distinctive characteristics of each domain. Unfortunately, standard NMT models trained on the mixed-domain data generally capture the general knowledge while ignoring the particular knowledge, rendering them sub-optimal for multi-domain translation (Koehn and Knowles 2017).

A natural approach to this problem is fine-tuning, which first trains a general model on all data and then separately fine-tunes it on each domain (Luong and Manning 2015). However, the fine-tuning approach requires maintaining a distinct NMT model for each domain, which makes it unwieldy in practice. Towards learning a unified multi-domain translation model, several researchers turn to augment the NMT model to learn domain-specific knowledge. For example, Kobus et al. (2016) introduced a special domain tag to the source sentence, and Britz et al. (2017) and Zeng et al. (2018) guide the encoder output to embed domain-specific knowledge via an auxiliary object. However, all the approaches require the encoder representations to embed both the general and the particular knowledge at the same time. Recent studies have shown that such overloaded usage of hidden representation makes training the model difficult, and such problem can be mitigated by separating these functions (Rocktäschel et al. 2017; Zheng et al. 2018).

In this work, we explicitly model the domain-specific functionality for multi-domain translation by introducing *domain transformation networks* (DTNs). More specifically, the DTNs transform the general knowledge learned by the encoder to the domain-specific knowledge, which is subsequently fed to the decoder. In this way, the encoder learns general knowledge in the standard fashion, and the newly added DTNs learn to preserve the particular knowledge. We employ a residual connection on DTNs to enable the decoder to exploit both the general and particular knowledge. To guarantee the knowledge transformation, we also propose two supervision strategies: 1) *domain distillation* that encourages the unified model to learn domain-specific knowledge in a teacher-student framework; and 2) *domain discrimination* that guides the encoder output and the transformed representation to embed the required knowledge with adversarial learning.

We conduct experiments on three language pairs: Chinese⇒English, German⇒English and English⇒French, covering balanced, unbalanced and large-scale multi-domain data. Experimental results show that our model significantly and consistently outperforms both the TRANSFORMER baseline by +3.35 BLEU points and previous multi-domain translation models (Kobus et al. 2016; Britz

[*] Work done when interning at Tencent AI Lab.

[1] The source code and experimental data are available at https://github.com/wangyong1122/dtn.

et al. 2017; Zeng et al. 2018) by +1.0∼2.0 BLEU points on different data, demonstrating the effectiveness and universality of the proposed approach. Encouragingly, our unified model is on par with the fine-tuning approach that requires multiple models to preserve the particular knowledge. Further analysis reveals that the domain transformation networks successfully capture the domain-specific knowledge while maintaining the specificity of each domain.

Contributions Our main contributions are:

1. Our study demonstrates the necessity of explicitly modeling the transformation from the general to the particular for multi-domain translation.
2. We exploit two supervision signals to simultaneously and incrementally encourage transformation of domain knowledge.
3. We construct several multi-domain data across languages, on which we empirically validate a variety of existing approaches.

Background

Neural Machine Translation A standard NMT model directly optimizes the conditional probability of a target sentence $\mathbf{y} = y_1, \dots, y_J$ given its corresponding source sentence $\mathbf{x} = x_1, \dots, x_I$:

$$P(\mathbf{y}|\mathbf{x};\theta) = \prod_{j=1}^{J} P(y_j|\mathbf{y}_{<j}, \mathbf{x};\theta) \tag{1}$$

where θ is a set of model parameters and $\mathbf{y}_{<j}$ denotes the partial translation. The probability $P(\mathbf{y}|\mathbf{x};\theta)$ is defined on the neural network based encoder-decoder framework (Sutskever et al. 2014; Cho et al. 2014), where the encoder summarizes the source sentence into a sequence of representations $\mathbf{H} = \mathbf{H}_1, \dots, \mathbf{H}_I$ with $\mathbf{H} \in \mathbb{R}^{I\times d}$, and the decoder generates target words based on the representations. Typically, this framework can be implemented as recurrent neural network (RNN) (Bahdanau et al. 2015), convolutional neural network (CNN) (Gehring et al. 2017) and Transformer (Vaswani et al. 2017). The parameters of the NMT model are trained to maximize the likelihood of a set of training examples $D = \{[\mathbf{x}^m, \mathbf{y}^m]\}_{m=1}^{M}$:

$$\mathcal{L}(\theta) = \arg\max_{\theta} \sum_{m=1}^{M} \log P(\mathbf{y}^m|\mathbf{x}^m;\theta) \tag{2}$$

The training corpus generally consists of data from various domains, which are not distinguished by the NMT model. This may pose difficulties to multi-domain translation.

Multi-Domain Translation This task aims to build a unified model on the mixed-domain data by maximizing performances across all domains. Formally, there are N subsets $D_1, ..., D_N$ from different domains, where the n-th domain of subset $D_n = \{[\mathbf{x}_n^m, \mathbf{y}_n^m]\}_{m=1}^{M_n}$. Accordingly, the training objective is

$$\mathcal{J}(\theta) = \arg\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n(\theta) \tag{3}$$

which maximizes the likelihood over training examples in each domain (i.e., $\mathcal{L}_n(\theta)$). As seen, there is no explicit signals to guide the model to learn domain-aware information in the learning objective function. As a result, the parameters in a standard NMT model generally capture the general knowledge while ignoring the domain-specific knowledge.

Approach

Our goal is to build a unified model, which can achieve good performance on all domains. As shown in Figure 1, we augment the standard NMT model with the introduced *Domain Transformation* networks, which transform the general encoding representations to the domain-specific representations. To guarantee the knowledge transformation effectively, we also propose two complementary supervision signals: *Domain Distillation* and *Domain Discrimination*, leveraging the power of knowledge distillation and adversarial learning.

Domain Transformation

Residual Transformation Networks The basic idea of domain transformation is to separate the specific features of each domain from the general features across multiple domains. First, we learn a shared encoder that maps input sentences to general representations that preserve common knowledge regardless of domains. Simultaneously, we learn a transformation component that explicitly transforms the general representations to domain-specific representation spaces, each of which represents distinctive characteristics of one single domain. The residual connection on transformation networks implicitly serves as an interpolation of the general and domain specific representations.

Formally, the transformation module reads a sequence of hidden states and outputs transformed ones. The source sentence $\mathbf{x}$ is first summarized into general representations $\mathbf{H}$ by a shared encoder of the standard NMT model. Conditioned on the input latent representations $\mathbf{H}$, we then employ a residual model (He et al. 2016) to generate domain-specific representations $\mathbf{H}'$ by:

$$\mathbf{H}' = \mathcal{F}(\mathbf{H}, W_n) + \mathbf{H} \tag{4}$$

where W_n is the parameters related to the n-th domain and $\mathcal{F}(\cdot)$ is the functional mapping which can be implemented by different types of neural networks such as feed-forward network (FNN), CNN and self-attention network (SAN). Subsequently, the output representations $\mathbf{H}'$, which encode both the general knowledge $\mathbf{H}$ and the domain-specific knowledge $\mathcal{F}(\mathbf{H}, W_n)$, are fed to a shared decoder for generating the target sentence $\mathbf{y}$.

The differences between domains are usually uncertain and tiny, which leads to inefficiency of directly fitting a desired underlying mapping. Recently, "residual" – a concept in deep neural networks (He et al. 2016; Sohn et al. 2019) has been successfully applied to extract feature differences in fields of image classification (Sohn et al. 2019) and speech recognition (Van Den Oord et al. 2016) and achieves remarkable improvements of performance. In our preliminary

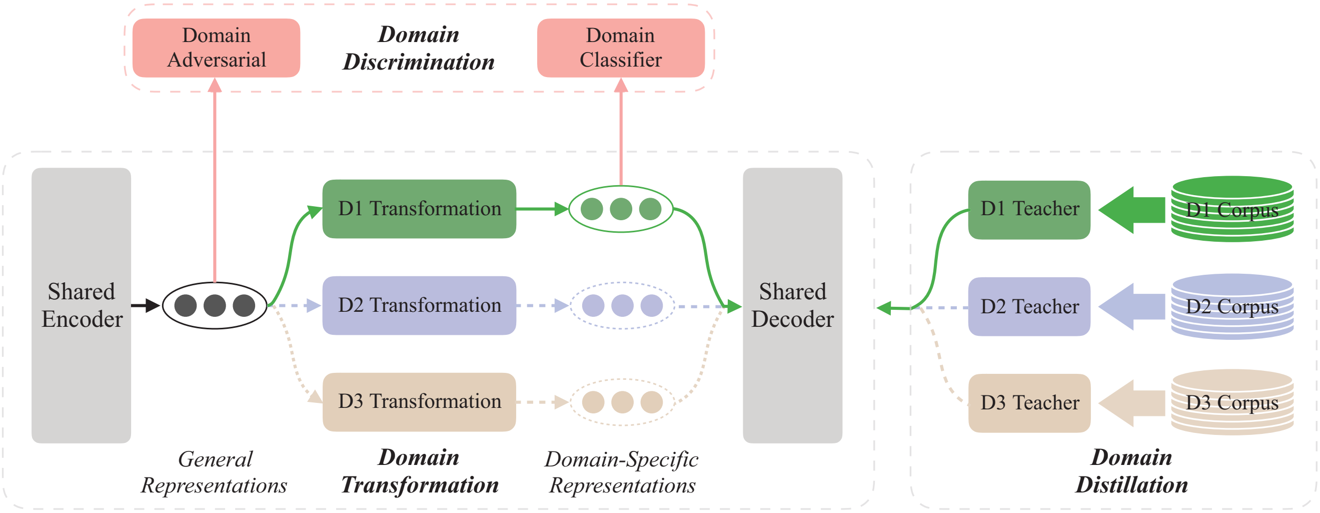

Figure 1: Architecture of the proposed multi-domain translation model, which consists of two key components: 1) *domain transformation* that transforms from the general representations to domain-specific representations, and we maintain a distinct transformation network for each domain; 2) *domain supervision* that contains two sub-components: *domain distillation* and *domain discrimination*. Domain distillation learns domain-specific model guided by domain teachers, which are fine-tuned on corresponding training corpora. Domain discrimination guides the two types of representations to embed the required content. In this example, the data of Domain 1 ("D1") are used to train the model, and solid line denotes the information flow.

experiments, we have investigated two different implementations of transformation networks including stacked feed-forward networks and multi-head attention networks. We found that the multi-head attention mechanism performs better in respect of capturing such domain-aware characteristics for the multi-domain translation task. In this study, we also parameterized $\mathcal{F}(\cdot)$ based on domain symbols, where each transformation module is able to maintain its own domain-aware parameters.

Domain-Aware Batch Learning To train distinct parameters of each domain, we propose a domain-aware learning strategy, in which one batch only contains training examples in a certain domain. One straightforward implementation is to alternately or randomly feed domain-aware batches into our proposed model. However, in the preliminary experiments, we found severe overfitting problems when using unbalanced multi-domain data. To overcome this, we propose a more balanced method, which heuristically selects a certain domain-batch by considering its distribution over the entire training corpus. Formally, domain-batches are sampled according to a multinomial distribution with probabilities $\{q_i\}_{i=1,\ldots,N}$:

$$q_i = \frac{p_i^\alpha}{\sum_{j=1}^N p_j^\alpha} \qquad p_i = \frac{n_i}{\sum_{k=1}^N n_k} \tag{5}$$

where n_i is the number of batches of the i-th domain and $\alpha = 0.7$ is the balance factor, which aims to increase the number of tokens associated with low-resource domains and alleviates the bias towards high-resource domains.

Domain Supervision

Domain Distillation The generalization ability of the teacher model can be transferred to the student by using the class probabilities produced by the cumbersome model for training the small model (Hinton et al. 2014). Recent studies on speech recognition show that training student networks with multiple teachers achieves promising empirical results (You et al. 2019).

Inspired by these observations, we propose to teach a unified model with multiple teachers trained on different domains. Specifically, we employ the soft targets produced by fine-tuned models as the supervision signal to train our unified model with the benefits of exploiting more data information and simultaneously reducing the interference across domains.

For the learning objective, we linearly interpolate soft target distribution produced by the corresponding domain teacher with hard labels:

$$\begin{aligned}\mathcal{L}(\theta) = \arg\max_{\theta} \sum_{(\mathbf{x},\mathbf{y})\in D} \Big\{ & (1-\lambda)\log P(\mathbf{y}|\mathbf{x};\theta) \\ & +\lambda \sum_{j=1}^{J}\sum_{k=1}^{|V|} \hat{P}(y_j = k|\mathbf{y}_{<j},\mathbf{x};\hat{\theta}) \\ & \times \log P(y_j = k|\mathbf{y}_{<j},\mathbf{x};\theta) \Big\}\end{aligned} \tag{6}$$

where λ is a hyper-parameter that is shared across multiple domains, $|V|$ is the vocabulary size of the target language, and $\hat{P}(\cdot)$ is the soft target.

Domain Discrimination Adversarial and discriminative learning can effectively distinguish between different types of features (Ganin and Lempitsky 2015; Chen et al. 2017b; Sun et al. 2018; Adams et al. 2019). In this work, we augment the transformation networks with the ability of domain discrimination. Specifically, the adversarial domain classifier is deployed at the input of DTNs, namely:

$$P(d|\mathbf{x};\psi) = \mathrm{softmax}(W_D^\top \widetilde{\mathbf{H}}) \tag{7}$$

where d is the domain symbol, W_D is the weights of softmax classifier and $\widetilde{\mathbf{H}}$ is the weighted representations of the encoding representations ($\mathbf{H}$), which is calculated as follows:

$$\widetilde{\mathbf{H}} = \sum_{i=1}^{I} \alpha_i \mathbf{H}_i \tag{8}$$

where the computation of α_i is similar to self-attention (Lin et al. 2017), in which the query is a trainable vector. Furthermore, we conduct a domain classifier on the output of DTNs $\widetilde{\mathbf{H}}'$ to guide it to embed domain-specific knowledge:

$$P(d|\mathbf{x};\gamma) = \mathrm{softmax}(W_D'^\top \widetilde{\mathbf{H}}') \tag{9}$$

where γ is a set of parameters of the domain classifier and $\widetilde{\mathbf{H}}'$ can be obtained according to Equation (8).

Overall, the training objective is a linear interpolation of the likelihood and the domain discrimination:

$$\mathcal{L}(\theta,\gamma,\psi) = \arg\max_{\theta,\gamma,\psi} \sum_{(\mathbf{x},\mathbf{y},d)\in D} \Big\{ \underbrace{\log P(\mathbf{y}|\mathbf{x};\theta)}_{\textit{likelihood}} + \underbrace{\log P(d|\mathbf{x};\gamma)}_{\textit{domain classifier}} + \underbrace{\log P(d|\mathbf{x};\psi) + \delta \times H(P(d|\mathbf{x};\psi))}_{\textit{domain adversarial}} \Big\} \tag{10}$$

where δ is the balance factor, and $H(P(\cdot))$ is the entropy of the probability distribution of the adversarial domain classifier with N domain labels. Following Zeng et al. (2018) and Chen et al. (2017c), we also employed the two-phase training strategy, where we alternatively optimized $\mathcal{L}(\theta,\gamma,\psi)$ with $\{\theta,\gamma\}$ and ψ. Besides, we discarded the component $\log P(d|\mathbf{x};\psi)$ when training the $\{\theta,\gamma\}$ parameter set.

Discussion While these two supervisions have their own characteristics, domain distillation and discrimination are complementary to each other. Domain distillation exploits more information of data, including shared and domain-aware knowledge across domains. As a strong supervision signal, domain discrimination is used to guide the transformation model to learn the distinct information between general and domain-specific representations.

Experiments

Setup

Data We conducted experiments on four different corpora, as listed in Table 1. For Chinese⇒English (Zh⇒En) translation, we used both a small-scale and a large-scale corpus. The small one is the same as that used by Zeng et al. (2018), and consists of four evenly distributed domains: *law*, *oral*, *thesis* and *news*. The large corpus is collected from CWMT2017 Lingosail, TVSub (Wang et al. 2018) and LDC, which consists of four balanced domains: *law*, *news*, *patent* and *subtitle*. For German⇒English (De⇒En) and English⇒French (En⇒Fr) translation tasks, we used a large amount of training data extracted from OPUS. They respectively contain four and two unevenly-distributed (unbalanced) domains including *law*, *medical*, *information technology* and *Koran* and *European Parliament*. The validation and test sets are officially-provided, otherwise randomly selected from the corresponding training corpora.

Corpus	D	$\|S\|$	Corpus	D	$\|S\|$
Zh⇒En (small)	Law	0.22	De⇒En	Law	0.59
	Oral	0.22		Med.	0.87
	Thesis	0.30		IT	0.31
	News	0.30		Koran	0.53
Zh⇒En (large)	Law	1.46	En⇒Fr		
	News	1.54		Med.	0.89
	Patent	2.90		Parl.	2.04
	Sub.	1.77			

Table 1: Statistics of training corpora: "D" and "$|S|$" indicate the domain and the number of sentences (in millions). As seen, Zh⇒En can be regarded as "balanced data" as the number of training samples is similar across domains while De⇒En and En⇒Fr are "unbalanced data" as the numbers of sentence pairs are very different.

All the data were tokenized and then segmented into subword symbols using byte-pair encoding (Sennrich et al. 2016b) with 30K merge operations to alleviate the out-of-vocabulary problem. We used 4-gram BLEU score (Papineni et al. 2002) as the evaluation metric, and bootstrap resampling (Koehn 2004) for statistical significance.

Model For fair comparison, we implemented all proposed and other approaches on the advanced *Transformer* model (Vaswani et al. 2017) using the open-source toolkit Fairseq (Ott et al. 2019). We followed Vaswani et al. (2017) to set the configurations of the NMT model, which consists of 6 stacked encoder/decoder layers with the layer size being 512. All the models were trained on 8 NVIDIA P40 GPUs where each was allocated with a batch size of 4,096 tokens. We trained the baseline model for 100K updates using Adam optimizer (Kingma and Ba 2015), and the proposed models were further trained with corresponding parameters initialized by the pre-trained baseline model. We fixed the hyper-parameters λ and δ as 0.1.

Baseline Comparisons To make the evaluation convincing, we re-implemented and compared with five previous models on multi-domain adaptation, which can be divided into two categories with respect to the number of models. The multiple-model approaches require to maintain a dedicated NMT model for each domain:

#	Architecture	#M	#Para.	BLEU					
				Law	Oral	Thesis	News	Avg.	△
	RNN-based NMT (Zeng et al. 2018)								
1	RNNSearch	1	–	45.82	9.15	13.93	19.73	22.16	–
2	+ Domain Context		–	55.03	10.20	18.04	22.29	26.39	–
	Transformer-based NMT (*this work*)								
3	Transformer	1	95.2M	65.72	10.28	20.38	27.22	30.90	–
4	+ Fine-tune (Luong and Manning 2015)	4	95.2M	$70.34^{\uparrow}$	8.15	$25.03^{\uparrow}$	$36.17^{\uparrow}$	34.92	+4.02
5	+ Mixed Fine-tune (Chu et al. 2017)		95.2M	66.81	9.42	18.28	$34.53^{\uparrow}$	32.26	+1.36
6	+ Domain Control (Kobus et al. 2016)	1	95.2M	66.18	10.06	20.45	28.10	31.20	+0.30
7	+ Domain Discriminate (Britz et al. 2017)		95.2M	65.81	8.99	21.20	$28.54^{\uparrow}$	31.15	+0.25
8	+ Domain Context (Zeng et al. 2018)		97.3M	66.81	9.75	$22.74^{\uparrow}$	$28.90^{\uparrow}$	32.05	+1.15
9	+ Domain Transformation	1	107.8M	$67.70^{\uparrow}$	8.88	$21.72^{\uparrow}$	$31.07^{\uparrow}$	32.34	+1.44
10	+ Domain Supervision		108.4M	66.63	8.23	$28.43^{\uparrow}$	$33.72^{\uparrow}$	34.25	+3.35

Table 2: Translation results on small-scale *balanced* Zh⇒En multi-domain data used by Zeng et al. (2018). We also list the results of Zeng et al. (2018) on RNN-based NMT. "#M" denotes the number of required models and "#Para." denotes the number of parameters. "+" denotes appending new features to the above row. "↑" indicates statistically significant difference ($p < 0.01$) from "Transformer" in the corresponding domains.

- *Fine-tune* (Luong and Manning 2015) that first trained a model on the entire data, and then fine-tuned multiple models separately using in-domain datasets.
- *Mixed Fine-tune* (Chu et al. 2017) that extended the fine-tune approach by training on out-of-domain data, then fine-tuning on in-domain and out-of-domain data.

The unified model methods handle adaptation to multiple domains within a unified NMT model:

- *Domain Control* (Kobus et al. 2016) that introduced domain tag to the source sentence.
- *Domain Discrimination* (Britz et al. 2017) that adopted domain classification via multitask learning.
- *Domain Context* (Zeng et al. 2018) that incorporated the word-level context for domain discrimination.

Our work falls into the unified model, where the above three related approaches are comparable to ours. Our work is not directly comparable to the fine-tuning approaches due to the different numbers of required models.

Results

Table 2 and Table 3 respectively show results on the small-scale balanced Zh⇒En data used by Zeng et al. (2018) and our newly-built large-scale corpus. Besides, Table 4 shows results on Zh⇒En and Zh⇒En multi-domain data. As seen, the proposed models significantly and incrementally improve the translation quality in all cases, although there are considerable differences among different scenarios.

Baselines In Table 2, the Transformer model (Row 3) greatly outperforms the results of RNN-based models reported by Zeng et al. (2018) on the same data (Rows 1-2), which makes the evaluation convincing in this work. The fine-tuning approaches (Rows 4-5) achieve significant improvements over the Transformer baseline. We attribute this to the facts that 1) the fine-tuning maintains a distinct model for each domain; and 2) there are sufficient data in each target domain. The unified models (Rows 6-8 in Table 2) consistently improve translation performance, and the "+Domain Context" method achieves the best performance at the cost of introducing additional parameters. The unified models are directly comparable to our approach.

Our Models As shown in Table 2, the proposed models (Row 9-10) outperform not only the Transformer baseline (Row 3) but also comparable approaches (Rows 6-8). Introducing transformation networks (Row 9) improves translation performance over Transformer baseline by +1.44 BLEU point, indicating that DTNs can effectively capture domain-aware knowledge. Besides, adding two supervision signals (Row 10) can outperform the baseline by +3.35 BLEU. Surprisingly, the performance of our unified model is on par with fine-tuning which requires four separate models (34.25 vs. 34.92 BLEU). This is encouraging, since the fine-tune approach catastrophically increases the overhead of deployment in practice, while our approach avoids this problem without a significant decrease of translation performance.

Translation Quality on Other Scenarios To validate the robustness of our approach, we also conducted experiments on a large-scale Zh⇒En corpus (as shown in Table 3) and other language pairs (as shown in Table 4). As seen, the superiority of our approach holds across different data sizes and language pairs, demonstrating the effectiveness and universality of the proposed approach. Furthermore, our unified model surprisingly outperforms the fine-tuning (multiple-model) on the unbalanced De⇒En corpus.

Analysis

We conducted extensive analyses on the small-scale Zh⇒En data to better understand our model in terms of effectiveness of domain transformation and supervision.

#	Architecture	Zh⇒En					
		Law	News	Patent	Tvsub	Avg.	△
1	Transformer	38.77	49.05	47.68	30.30	41.45	–
2	+ Fine-tune (Luong and Manning 2015)	42.32$^{\uparrow}$	50.34$^{\uparrow}$	49.16$^{\uparrow}$	30.54	43.09	+1.64
3	+ Mixed Fine-tune (Chu et al. 2017)	40.84$^{\uparrow}$	49.53	46.45	30.95	41.91	+0.49
4	+ Domain Control (Kobus et al. 2016)	39.27	49.30	48.02	30.55	41.79	+0.34
5	+ Domain Discriminate (Britz et al. 2017)	39.21	49.07	47.76	30.16	41.55	+0.10
6	+ Domain Context (Zeng et al. 2018)	39.35	49.77	47.71	30.31	41.79	+0.34
7	+ Domain Transformation	40.04$^{\uparrow}$	50.35$^{\uparrow}$	48.35$^{\uparrow}$	30.96	42.43	+0.98
8	+ Domain Supervision	41.01$^{\uparrow}$	50.55$^{\uparrow}$	48.61$^{\uparrow}$	31.55$^{\uparrow}$	42.93	+1.48

Table 3: Translation results on *large-scale* balanced Zh⇒En multi-domain data built in this work. "+" denotes appending new features to the above row. "↑" indicates statistically significant difference ($p < 0.01$) from "Transformer" on different domains.

#	Architecture	De⇒En						En⇒Fr			
		Law	Med.	IT	Koran	Avg.	△	Med.	Par.	Avg.	△
1	Transformer	62.72	66.26	40.95	24.82	48.68	–	65.91	35.58	50.75	–
2	+ Fine-tune	65.10$^{\uparrow}$	68.03$^{\uparrow}$	42.76$^{\uparrow}$	21.49	49.35	+0.67	68.56$^{\uparrow}$	35.98	52.27	+1.52
3	+ Mixed Fine-tune	63.48$^{\uparrow}$	66.08	41.88	26.95$^{\uparrow}$	49.60	+0.92	67.87$^{\uparrow}$	35.29	51.58	+0.83
4	+ Domain Contr.	63.04	66.69	41.13	24.27	48.78	+0.10	67.01$^{\uparrow}$	35.70	51.36	+0.61
5	+ Domain Discr.	62.74	66.31	41.04	23.93	48.51	-0.17	65.98	35.74	50.86	+0.11
6	+ Domain Conte.	63.29	66.95	41.66	23.12	48.76	+0.08	66.57	35.67	51.12	+0.37
7	+ Domain Trans.	63.33	66.95	42.32$^{\uparrow}$	24.00	49.15	+0.47	67.23$^{\uparrow}$	35.80	51.52	+0.77
8	+ Domain Super.	64.59$^{\uparrow}$	67.95$^{\uparrow}$	42.16$^{\uparrow}$	24.09	49.70	+1.02	67.85$^{\uparrow}$	35.80	51.83	+1.08

Table 4: Translation results on *unbalanced* De⇒En and En⇒Fr multi-domain data. "+" denotes appending new features to the above row. "↑" indicates statistically significant difference ($p < 0.01$) from "Transformer" on different domains.

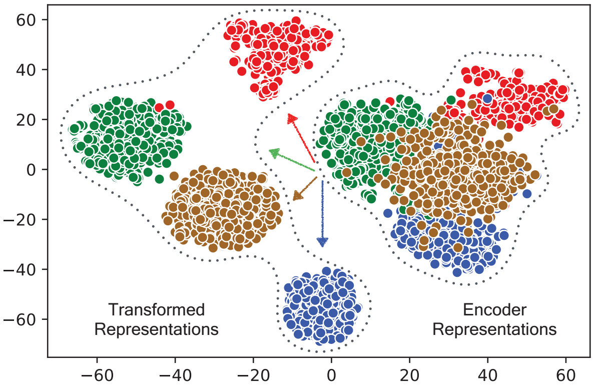

Figure 2: Visualization of encoder (left) and transformed (right) representations. Dots in different colors denote sentences in different domains.

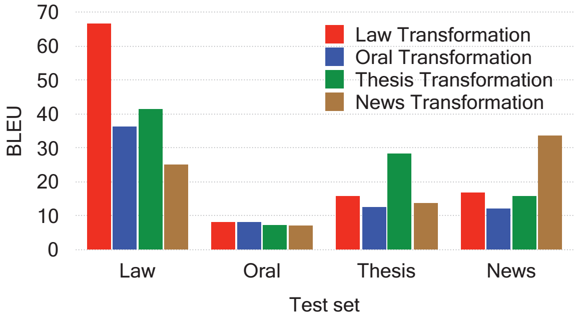

Figure 3: Translation results of test set in each domain decoded by four domain-specific transformation modules. As seen, each specialized transformation model performs best on its corresponding domain.

Effects of Domain Transformation

Domain Transformation With the dimension reduction technique of t-SNE (Maaten and Hinton 2008), we visualized the general and domain-specific representations of source sentences in test set. As shown in Figure 2, the representation vectors in different domains are centered in different regions. Furthermore, the distribution of encoder representations is concentrated to preserve shared knowledge, while the transformed representations are diverse to keep domain-specific characteristics. This confirms that our approach is able to distinctively transform the source-side domain knowledge from the general to the particular.

Domain-Specific Translation We further examined whether each specialized transformation module acquires its specific domain knowledge. Figure 3 shows the translation results of test set in each domain decoded by four different domain-specific transformation modules. As seen, the each transformation module performs best on its corresponding domain. Some domains with more distinctive characteristics (e.g., *Law*) can achieve more significant

#	Model	BLEU
1	Transformer	30.90
2	+ Distillation (sequence)	31.45
3	+ Distillation (word)	31.51
4	+ Domain Transformation	32.34
5	+ Domain Distillation (sequence)	32.70
6	+ Domain Distillation (word)	33.05
7	+ Domain Discrimination	33.18
8	+ Both	34.25

Table 5: Translation results when different supervision signals are used for training our multi-domain model. "Distillation (sequence)" and "Distillation (word)" denote applying distillation at sequence and word level, respectively. "Both" denotes applying "Discrimination" and "Distillation (word)".

performances. In contrast, in less-distinctive domains (e.g., *Oral*), different transformation modules have similar performances. This is consistent with our expectation that each transformation component is specialized to maintain particular knowledge in one domain.

Effects of Domain Supervision

Contribution Analysis Table 5 lists translation results when baseline or our model uses either *domain distillation* or *domain discrimination*, or both signals. As seen, adding supervision signal consistently improves the performance over the "Domain Transformation" model (Rows 6-7), and combining both signals accumulatively achieves the best performance (+1.9 BLEU, Row 8). This confirms the hypothesis in the section of domain supervision that the effects are reflected in three aspects: 1) weak supervision encourages model to exploit both shared and domain-aware knowledge across domains; 2) strong supervision guides model to learn distinct features; 3) combination makes them complementary to each other. It is also interesting to investigate the effect of domain supervision without transformation networks (Rows 1-3), which still improves performance (31.45 vs. 30.90), demonstrating the effectiveness and universality of domain supervision.

Concerning the distillation approach (Rows 2-3 and 5-6), we revisited word-level and sequence-level distillation methods for Transformer-based NMT. Different from the results reported by Kim and Rush (2016) on RNN-based models, we found that word-level distillation marginally outperformed its sequence-level counterpart (31.51 vs. 31.45 on top of "Transformer", and 33.05 vs. 32.70 on top of "+ Domain Transformation"). Through case studies, we found that word-level distillation produced more fluent outputs, possibly due to providing smoother target labels. This explains why word-level distillation is a widely-used implementation in multi-lingual and multi-domain tasks on top of Transformer-based models (Tan et al. 2019; You et al. 2019). Therefore, we applied word-level distillation in our work.

Case Study Table 6 shows a translation example randomly selected from the test set in *Thesis* domain. As seen, augmenting transformation module into NMT can generate more domain-specific words but with relatively lower fluency. Adding supervision signals can incrementally generate more fluent domain-specific phrases and patterns. For instance, the Chinese word "yuan wai" is ignored by baseline and mis-translated by "+Trans." model, while the "+Supervison" models correctly translate it into "outside hospital". This demonstrates that our model can comprehensively capture domain-specific knowledge in terms of words, phrases and patterns.

Input	143 li yuan wai xin bo zhou ting huan zhe jing song lai yi yuan, fu su cun huo jin 2 li (1.4%).
Reference	In the other 143 patients occurring *sudden arrest of heart beat* **outside hospital**, only 2 survived (1.4%).
Baseline	In the other 143 patients who received *cardiac arrest*, only 2 survived (1.4%).
+Trans.	In the 143 patients admitted to **hospital**, only 2 (1.4%) survived for *resuscitation*.
+Distill.	In the other 143 patients who suffered *a sudden arrest of heart beat* **outside hospital**, only 2 (1.4%) survived.
+Discri.	In the other 143 patients who suffered *sudden arrest of heart beat* **outside hospital**, only 2 survived (1.4%).

Table 6: An example of Zh⇒En translation sampled from *Thesis* test set. Domain-specific words, phrases and patterns are highlighted with text formatting (i.e. italic, bold or underline). Our "+Trans.", "+Distill." and "+Discri" models are consistent with Table 5. As seen, augmenting transformation networks into NMT can generate more domain-specific words but with low fluency. Adding supervision signals can incrementally generate more fluent domain-specific phrases and patterns.

Related Work

Domain Adaptation From conventional statistical machine translation (SMT) to state-of-the-art NMT, domain adaptation techniques have been widely investigated to adapt models trained on one or more source domains to outside target domain (Chu and Wang 2018; Wang et al. 2017a; van der Wees et al. 2017; Chen et al. 2017a; Wang et al. 2017b). Although domain adaptation techniques boost translation quality on in-domain data, translation quality for out-of-domain data tends to degrade.

Fine-tune is the conventional way for domain adaptation (Luong and Manning 2015; Sennrich et al. 2016a; Freitag and Al-Onaizan 2016). Chu et al. (2017) extended the fine-tune strategy by training the model on out-of-domain data, which is then fine-tuned on a mix of in-domain and out-of-domain data. The two approaches can be easily applied to multi-domain translation by separately maintaining a fine-tuned model for each domain. In this study, we empirically compare with the fine-tune strategies, and find that our unified model achieves comparable results with the fine-tuning approaches.

Multi-Domain Translation Multi-domain machine translation aims to construct the NMT model with the ability of translating sentences across different domains. Kobus et al. (2016) introduced embeddings of source domain tag to the encoder, which can perform domain-adapted translations in multiple domains. Britz et al. (2017) presented various mixing paradigms for multi-domain settings, and demonstrated their efficacy across multiple language pairs. Zeng et al. (2018) explored utilizing word-level domain contexts and jointly modeled multi-domain NMT and domain classification tasks. Our work is different in that 1) we learn the domain-specific knowledge by transforming from the general knowledge, while Zeng et al. (2018) split the encoder representation into general and domain-specific representations with two separate gates; and 2) we maintain a distinct transformation network with its own parameters for each domain, while Zeng et al. (2018) used a shared set of parameters across domains. In addition, we exploit more domain supervision techniques (e.g., domain distillation) to further improve multi-domain translation performance.

Furthermore, Gu et al. (2019) maintained a distinct set of encoder-decoder for each domain. This is analogous to the fine-tuning strategy, which maintains multiple models rather than a unified model for multi-domain translation. In addition, our approach also benefits from capturing the correlations between the general and domain-specific knowledge with the introduced transformation networks.

Conclusion and Future Work

In this paper, we propose to explicitly transform domain knowledge from the general to the particular for a multi-domain NMT model. In order to guarantee knowledge transformation, we also exploit two kinds of supervision signal to further improve the translation quality. Empirical results on a variety of language pairs demonstrate the effectiveness and universality of the proposed approach. We also conducted extensive analyses to demonstrate the necessity of explicitly modelling the transformation of domain knowledge for multi-domain translation.

The proposed approach significantly improves translation performance at the cost of increased computational complexity. Network compression would be a promising direction to alleviate this problem. In future work we plan to exploit different model compacting techniques such as knowledge distillation (Hinton et al. 2014) and network pruning (Han et al. 2016), to make deployment of our approach more practical.

References

Adams, O.; Wiesner, M.; Watanabe, S.; and Yarowsky, D. 2019. Massively multilingual adversarial speech recognition. In *NAACL*.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Britz, D.; Le, Q.; and Pryzant, R. 2017. Effective domain mixing for neural machine translation. In *CMT*.

Chen, B.; Cherry, C.; Foster, G.; and Larkin, S. 2017a. Cost weighting for neural machine translation domain adaptation. In *WNMT*.

Chen, X.; Shi, Z.; Qiu, X.; and Huang, X. 2017b. Adversarial multi-criteria learning for chinese word segmentation. In *ACL*.

Chen, Y.; Liu, Y.; Cheng, Y.; and Li, V. O. 2017c. A teacher-student framework for zero-resource neural machine translation. In *ACL*.

Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMNLP*.

Chu, C., and Wang, R. 2018. A survey of domain adaptation for neural machine translation. In *COLING*.

Chu, C.; Dabre, R.; and Kurohashi, S. 2017. An empirical comparison of simple domain adaptation methods for neural machine translation. In *ACL*.

Freitag, M., and Al-Onaizan, Y. 2016. Fast domain adaptation for neural machine translation. In *arXiv preprint arXiv:1612.06897*.

Ganin, Y., and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional sequence to sequence learning. In *ICML*.

Gu, S.; Feng, Y.; and Liu, Q. 2019. Improving domain adaptation translation with domain invariant and specific information. In *NAACL*.

Han, S.; Mao, H.; and Dally, W. J. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hinton, G.; Vinyals, O.; and Dean, J. 2014. Distilling the knowledge in a neural network. In *NIPS Deep Learning Workshop*.

Kim, Y., and Rush, A. M. 2016. Sequence-level knowledge distillation. In *EMNLP*.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kobus, C.; Crego, J.; and Senellart, J. 2016. Domain control for neural machine translation. In *RANLP*.

Koehn, P., and Knowles, R. 2017. Six challenges for neural machine translation. In *WNMT*.

Koehn, P. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*.

Lin, Z.; Feng, M.; Santos, C. N. d.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. In *ICLR*.

Luong, M.-T., and Manning, C. D. 2015. Stanford neural machine translation systems for spoken language domains. In *IWSLT*.

Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. In *JMLR*.

Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; and Auli, M. 2019. Fairseq: A fast, extensible toolkit for sequence modeling. In *NAACL*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Rocktäschel, T.; Welbl, J.; and Riedel, S. 2017. Frustratingly short attention spans in neural language modeling. In *ICLR*.

Sennrich, R.; Haddow, B.; and Birch, A. 2016a. Improving neural machine translation models with monolingual data. In *ACL*.

Sennrich, R.; Haddow, B.; and Birch, A. 2016b. Neural machine translation of rare words with subword units. In *ACL*.

Sohn, K.; Shang, W.; Yu, X.; and Chandraker, M. 2019. Unsupervised domain adaptation for distance metric learning. In *ICLR*.

Sun, S.; Yeh, C.-F.; Hwang, M.-Y.; Ostendorf, M.; and Xie, L. 2018. Domain adversarial training for accented speech recognition. In *ICASSP*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Tan, X.; Ren, Y.; He, D.; Qin, T.; Zhao, Z.; and Liu, T.-Y. 2019. Multilingual neural machine translation with knowledge distillation. In *ICLR*.

Van Den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A. W.; and Kavukcuoglu, K. 2016. Wavenet: A generative model for raw audio. In *SSW*.

van der Wees, M.; Bisazza, A.; and Monz, C. 2017. Dynamic data selection for neural machine translation. In *EMNLP*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*.

Wang, R.; Finch, A.; Utiyama, M.; and Sumita, E. 2017a. Sentence embedding for neural machine translation domain adaptation. In *ACL*.

Wang, R.; Utiyama, M.; Liu, L.; Chen, K.; and Sumita, E. 2017b. Instance weighting for neural machine translation domain adaptation. In *EMNLP*.

Wang, L.; Tu, Z.; Shi, S.; Zhang, T.; Graham, Y.; and Liu, Q. 2018. Translating pro-drop languages with reconstruction models. In *AAAI*.

You, Z.; Su, D.; and Yu, D. 2019. Teach an all-rounder with experts in different domains. In *ICASSP*.

Zeng, J.; Su, J.; Wen, H.; Liu, Y.; Xie, J.; Yin, Y.; and Zhao, J. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In *EMNLP*.

Zheng, Z.; Zhou, H.; Huang, S.; Mou, L.; Xinyu, D.; Chen, J.; and Tu, Z. 2018. Modeling past and future for neural machine translation. In *TACL*.