

Acquiring Knowledge from Pre-Trained Model to Neural Machine Translation

Rongxiang Weng,^{1,2*} Heng Yu,² Shujian Huang,¹
Shanbo Cheng,² Weihua Luo²

¹National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

²Machine Intelligence Technology Lab, Alibaba Group, Hangzhou, China
{wengrx, yuheng.yh, weihua.luowh}@alibaba-inc.com, huangsj@nju.edu.cn,

Abstract

Pre-training and fine-tuning have achieved great success in natural language process field. The standard paradigm of exploiting them includes two steps: first, pre-training a model, e.g. BERT, with a large scale unlabeled monolingual data. Then, fine-tuning the pre-trained model with labeled data from downstream tasks. However, in neural machine translation (NMT), we address the problem that the training objective of the bilingual task is far different from the monolingual pre-trained model. This gap leads that only using fine-tuning in NMT can not fully utilize prior language knowledge. In this paper, we propose an APT framework for acquiring knowledge from pre-trained model to NMT. The proposed approach includes two modules: 1). a dynamic fusion mechanism to fuse task-specific features adapted from general knowledge into NMT network, 2). a knowledge distillation paradigm to learn language knowledge continuously during the NMT training process. The proposed approach could integrate suitable knowledge from pre-trained models to improve the NMT. Experimental results on WMT English to German, German to English and Chinese to English machine translation tasks show that our model outperforms strong baselines and the fine-tuning counterparts.

1 Introduction

Neural machine translation (NMT) based on the *encoder-decoder framework* (Sutskever, Vinyals, and Le 2014; Cho et al. 2014; Bahdanau, Cho, and Bengio 2014; Luong, Pham, and Manning 2015) has obtained state-of-the-art performance on many language pairs (Deng et al. 2018). Various advanced neural architectures have been explored for NMT under this framework, such as recurrent neural network (RNN) (Bahdanau, Cho, and Bengio 2014; Luong, Pham, and Manning 2015, RNNSearch), convolutional neural network (CNN) (Gehring et al. 2016, Conv-S2S) and self-attention network (Vaswani et al. 2017, Transformer).

Currently, most NMT systems only utilize the sentence-aligned parallel corpus for model training. Monolingual data, which is larger and easier to collect, is not fully utilized limiting the capacity of NMT models. Previously, sev-

eral successful attempts have been made to improve NMT by incorporating monolingual data (Gulcehre et al. 2015; Sennrich, Haddow, and Birch 2016; Zhang and Zong 2016; Poncelas et al. 2018). However, these studies only focus on the usage of word-level information, e.g. extracting information from word embedding. The rich contextual information from large scale monolingual data does not be fully utilized. Meanwhile, fine-tuning the parameters from unsupervised pre-trained models, like GPT (Radford et al. 2018) or BERT (Devlin et al. 2018), in which downstream tasks could exploit the contextual knowledge from large scale monolingual data, has gained tremendous success in a variety of natural language process tasks. Thus, the upcoming question is *whether the contextual knowledge from pre-trained models are useful in NMT*.

Due to the limited amount of high-quality parallel data, NMT as a complex text generation task, can not generate appropriate representation. The contextual knowledge from pre-trained models could naturally be a good complement for NMT. Nevertheless, how to integrate the knowledge from pre-trained models into NMT is another challenge: the improvement in NMT with the standard fine-tuning operation is relatively less (Sun et al. 2019). The main reason is that the training objective of the bilingual task is far different from the monolingual task. For example, even with a multi-lingual setting, the objective of BERT or GPT is also predicting the words from the same language, while translation requires the conversion of one language to another. Especially in languages having large differences in morphology and syntax, this gap will lead to that general knowledge from pre-trained models will be erased in the training process of NMT (Yang et al. 2019a). When using the fine-tuning method directly can not work well, *how to explore the potential abilities of pre-trained models in NMT* is an urgent problem to be solved.

In this paper, to address this appealing challenge, we design an APT framework for acquiring the knowledge from pre-trained models to NMT. Specifically, our APT framework has two modules. First, we propose a *dynamic fusion mechanism* which can learn a task-specific representation by adapting the general representation from pre-trained models, and adopt two controlling methods based on different granu-

*Code is available at: <https://github.com/wengrx/APT-NMT>
Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

larities to fuse the task-specific representation into NMT dynamically. This method could provide rich contextual information for NMT to model sentence better. Second, we introduce a *knowledge distillation paradigm* to distill the knowledge from pre-trained models to NMT continuously. With this method, NMT could learn the knowledge about how to translate source sentence to target sentences from parallel data and how to generate a better target sentence from monolingual data in the training process. Furthermore, according to our analysis and empirical results, we conclude that the best strategy for using the two methods in the encoder-decoder framework to improve translation quality.

To demonstrate the effectiveness of our APT approach, we implement the proposed approach based on the advanced pre-trained models and Transformer model (Vaswani et al. 2017). It is worth to mention that this framework could be applied to various neural structures based on the encoder-decoder framework (Bahdanau, Cho, and Bengio 2014; Gehring et al. 2016; Vaswani et al. 2017). Experimental results on WMT English to German, German to English and Chinese to English machine translation tasks show that our approach with BERT (Devlin et al. 2018) or GPT (Radford et al. 2018) outperforms the Transformer baseline and the fine-tuning counterparts.

2 Background

2.1 Neural Machine Translation

Here, we will introduce neural machine translation based on the Transformer network (Vaswani et al. 2017), which has achieved state-of-the-art performance in several language pairs (Deng et al. 2018).

Denoting a source-target parallel sentence pair as $\{\mathbf{x}, \mathbf{y}\}$ from the training set, where \mathbf{x} is the source sequence $(x_1, x_2, \dots, x_i, \dots, x_I)$ and \mathbf{y} is the target sequence $(y_1, y_2, \dots, y_j, \dots, y_J)$, I and J are the length of \mathbf{x} and \mathbf{y} , respectively.

In the encoding stage, a multiple layer encoder based on the self-attention architecture is used to encode \mathbf{x} into \mathbf{R}_N^E , which is composed by a sequence of vectors $(\mathbf{r}_{N,1}^E, \mathbf{r}_{N,2}^E, \dots, \mathbf{r}_{N,i}^E, \dots, \mathbf{r}_{N,I}^E)$, N is the depth of the encoder. The representation \mathbf{R}_N^E is calculated by:

$$\mathbf{R}_N^E = \text{LN}(\mathbf{H}_N^E + \text{FFN}(\mathbf{R}_{N-1}^E)), \quad (1)$$

where the $\text{LN}(\cdot)$ and $\text{FFN}(\cdot)$ are layer normalization (Ba, Kiros, and Hinton 2016) and feed forward network, respectively. The \mathbf{R}_{N-1}^E is from the $(N-1)$ th layer. The \mathbf{H}_N^E is computed by:

$$\mathbf{H}_N^E = \text{Att}(\mathbf{Q}_N^E, \mathbf{K}_{N-1}^E, \mathbf{V}_{N-1}^E), \quad (2)$$

where the $\text{Att}(\cdot)$ is a self-attention network and the \mathbf{Q}_N^E , \mathbf{K}_{N-1}^E , \mathbf{V}_{N-1}^E are query, key and value matrix, respectively. In this stage, they are equal to \mathbf{R}_{N-1}^E .

Typically, we define that \mathbf{R}_0^E is composed by $\text{emb}(x_i)$, which is the word embedding of x_i .

In the decoding stage, the decoder maximizes the conditional probability of generating the j th target word, which

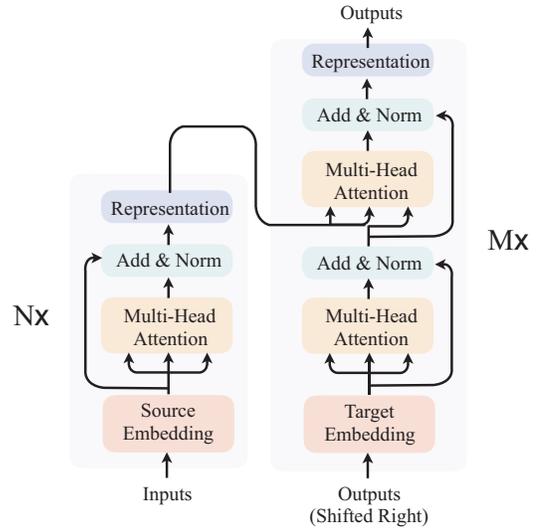


Figure 1: Overview of the structure of the Transformer network (Vaswani et al. 2017).

is defined as:

$$P(y_j|y_{<j}, \mathbf{x}) = \text{softmax}(\text{FFN}(\mathbf{r}_{M,j}^D)), \quad (3)$$

$\mathbf{r}_{M,j}^D$ is a vector from the target representation matrix \mathbf{R}_M^D , M is the depth of the decoder. The \mathbf{R}_M^D is

$$\mathbf{R}_M^D = \text{LN}(\text{FFN}(\mathbf{S}_M^D + \mathbf{C}_M^D)), \quad (4)$$

\mathbf{S}_M^D is computed by Eq. 1-2, where the query, key and value are equal to \mathbf{R}_{M-1}^D . \mathbf{C}_M^D is computed by:

$$\mathbf{C}_M^D = \text{Att}(\mathbf{Q}_M^D, \mathbf{K}_N^E, \mathbf{V}_N^E), \quad (5)$$

where \mathbf{Q}_M^D is equal to \mathbf{S}_M^D , \mathbf{K}_N^E and \mathbf{V}_N^E are equal to \mathbf{R}_N^E .

Finally, Transformer is optimized by maximizing the likelihood, denoted by:

$$\mathcal{L}_T = \frac{1}{J} \sum_{i=1}^J \log P(y_j|y_{<j}, \mathbf{x}; \theta_T), \quad (6)$$

where the θ_T is the parameters of NMT. The overview of the structure of Transformer is shown in Figure 1.

2.2 Pre-training Model

Recently, a variety of pre-training models (PT), like ELMo (Peters et al. 2018), GPT (Radford et al. 2018), BERT (Devlin et al. 2018), etc, are proposed to obtain language knowledge from large scale monolingual data.

Formally, given a sentence $\mathbf{z} = (z_1, z_2, \dots, z_k, \dots, z_K)$, K is the length of \mathbf{z} , the pre-trained model is adopted to get the contextual representation: $\mathbf{R}_L^P = \text{PT}(\mathbf{z}; \theta_P)$, where L is the depth of the pre-trained model and θ_P is the parameters of the pre-training model. The $\text{PT}(\cdot)$ could be implemented by a variety of structures like Bi-LSTM (Peters et al. 2018) or self-attention network (Radford et al. 2018; Devlin et al. 2018).

There are two main objectives to train the pre-trained model (Yang et al. 2019b). The first kind is using an auto-regressive language model objective, which predicts the next word $P(z_k|z_{<k}; \theta_p)$ by the k th representation $\mathbf{r}_{L,k}^p$ from \mathbf{R}_L^p . Another popular method is similar to the auto-encoder, which needs to pre-process the sentence \mathbf{z} to get a processed one $\hat{\mathbf{z}}$ by masking several words \mathbf{z}^m . Then, the pre-trained model predicts the masked words to re-construct the \mathbf{z} by $P(\mathbf{z}^m|\hat{\mathbf{z}}; \theta_p)$ in the training process.

3 Approach

Owing to the limited amount of parallel data, it is hard for NMT to generate appropriate contextual representation. The pre-trained models are an useful complement to provide NMT models with proper language knowledge. However, previous integration methods like fine-tuning: initializing parameters from pre-trained models, may not suit for machine translation which is a bilingual generation task. The general contextual information from pre-trained models is quite different from the task-specific representation of NMT model.

Thus, we propose a novel APT framework including a *dynamic fusion mechanism* and a *knowledge distillation paradigm*, to fully utilize pre-trained contextual knowledge in NMT models. We will introduce the two methods in details and discuss the different integration strategies in the encoder and decoder of NMT models. For convenience, we will present the dynamic fusion mechanism on the encoder and the knowledge distillation paradigm on the decoder, respectively.

3.1 Dynamic Fusion Mechanism

We propose a *dynamic fusion mechanism* to obtain the *task-specific representation* by transforming general pre-trained representations in pre-trained models. Specifically, we use an adapter for transforming general knowledge to more appropriate features of NMT during the training process. Furthermore, previous work (Peters et al. 2018; Dou et al. 2018; Wang et al. 2018) shows that representations from each layer in a deep model have different aspect of meaning. Following this intuition, we expand our idea by employing the adapter on all layers' representation from pre-trained models to get different kinds of knowledge, from concrete to abstract.

Formally, the general representations from pre-trained models are $\mathbf{R}^P = (\mathbf{R}_1^P, \dots, \mathbf{R}_l^P, \dots, \mathbf{R}_L^P)$. For the l th layer's representation \mathbf{R}_l^P , the task-specific representation is computed by:

$$\mathbf{R}_l^T = G_l(\mathbf{R}_l^P), \quad (7)$$

where the proposed adapter $G_l(\cdot)$ is a simple MLP. Mikolov, Le, and Sutskever (2013) and Wu et al. (2019) pointed out the representation space of similar languages can be transferred by a linear mapping. In our scenario, which is in same language, the mapping function can transfer the general representation to task-specific representation effectively.

Subsequently, we propose two methods based on different granularity to control how much the task-specific representation should be fused into Transformer dynamically. First,

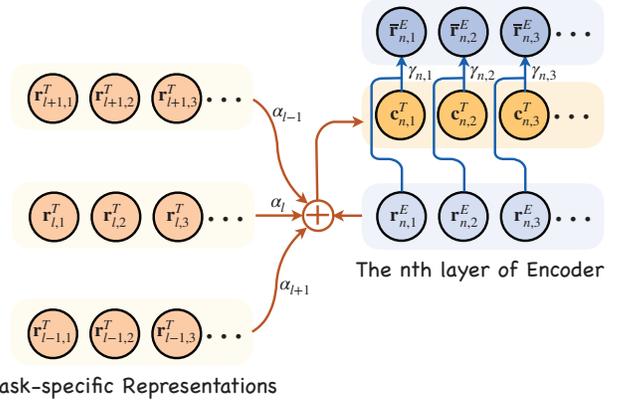


Figure 2: Overview of the dynamic fusion mechanism employed on the encoder of Transformer.

the demand of external information from each layer is different. Thus, compared with using layer coordination (He et al. 2018) directly, we further propose a *layer-aware attention mechanism* to capture compound contextual information. Formally, given the n th layer's vanilla representation \mathbf{R}_n^E computed by Equation 1-2, the corresponding external representation is computed by:

$$\mathbf{C}_n^T = \sum_{l=1}^L \alpha_l \mathbf{R}_l^T, \alpha_l = \frac{\exp(e_l)}{\sum_{t=1}^L \exp(e_t)}, \quad (8)$$

$$e_l = \text{FFN}\left(\frac{1}{I} \sum_{i=1}^I \mathbf{r}_{l,i}^T \cdot \frac{1}{I} \sum_{i=1}^I \mathbf{r}_{n,i}^E\right). \quad (9)$$

The layer-aware attention mechanism can determine which representation from pre-trained model is more important for current layer. The composite representation \mathbf{C}_n^T can capture more suitable information by considering a larger context.

Following above intuition, the demand of each hidden state from same layer is also different. A fine-grained method is necessary to control the fusion ratio of each hidden state. We adopt a simple *contextual gating mechanism* (Kuang et al. 2018) to implement it.

Formally, the representation $\mathbf{c}_{n,i}^T$ from \mathbf{C}_n^T is fused into the corresponding state $\mathbf{r}_{n,i}^E$ from \mathbf{R}_n^E by:

$$\bar{\mathbf{r}}_{n,i}^E = \mathbf{r}_{n,i}^E + \gamma_{n,i} * \mathbf{c}_{n,i}^T, \quad (10)$$

where the gate $\gamma_{n,i}$ is computed by:

$$\gamma_{n,i} = \text{sigmoid}(\text{FFN}(\mathbf{r}_{n,i}^E \cdot \mathbf{c}_{n,i}^T)) \quad (11)$$

The overview is illustrated in Figure 2. Different from previous works (Ramachandran, Liu, and Le 2017; Peters et al. 2018; Radford et al. 2018), the proposed feature-based method can make a *deep fusion* which could incorporate appropriate information into each layer, that is, Transformer can access specific surface information in lower layers and the latent one in higher layers.

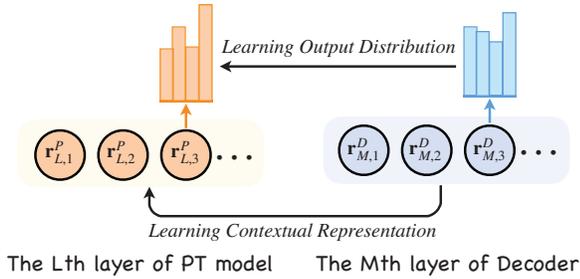


Figure 3: Overview of the knowledge distillation paradigm employed on the decoder of Transformer.

3.2 Knowledge Distillation Paradigm

Besides the dynamic fusion mechanism, we also propose a *knowledge distillation paradigm* to learn pre-trained representation in the training process. We introduce two auxiliary learning objectives distilling the knowledge from pre-trained models to NMT in word and sentence levels, respectively.

Firstly, the word level knowledge distillation objective is defined as:

$$\mathcal{L}_W = \frac{1}{J} \sum_{j=1}^J \sum_{k=1}^V P(y_j = k | y; \theta_P) \cdot \log(P(y_j = k | \mathbf{x}, y_{<j}; \theta_T)) \quad (12)$$

where the J is the length of the given target sentence \mathbf{y} , the V is vocabulary size. The $P(y_j | y_{<j}, \mathbf{x}; \theta_T)$ is computed by Equation 3. Compared with only minimizing the one-hot label from reference, this word level training function can learn the output distribution from pre-trained models, which is more diverse.

Then, different from previous sentence level knowledge distillation methods (Chen et al. 2017), our objective learns the sentence level information by fitting contextual representation directly:

$$\begin{aligned} \mathcal{L}_S &= \frac{1}{J} \|\mathbf{R}_M^D - \mathbf{R}_L^P\|_2^2 \\ &= \frac{1}{J} \sum_{j=1}^J \|\mathbf{r}_{M,j}^D - \mathbf{r}_{L,j}^P\|_2^2, \end{aligned} \quad (13)$$

where the M is output layer of the decoder. The $\mathbf{r}_{M,j}^D$ and $\mathbf{r}_{L,j}^P$ are from the decoder and pre-trained model, respectively. The vanilla sentence level training objective need to sample output sentence in the training process, which may cause a bias and decrease efficiency, while our method could learn contextual information from the hidden state directly. The overview of the proposed knowledge distillation paradigm is shown in Figure 3.

Finally, the loss function of our APT is:

$$\mathcal{L} = \mathcal{L}_T + \eta \cdot \mathcal{L}_S + \beta \cdot \mathcal{L}_W, \quad (14)$$

where η and β are used to balance the preference among the two losses, which we are set to 0.5 individually.

3.3 Integration Strategy

The duties of encoder and decoder in NMT are different. And the decoder has two states in the training and inference stages. Thus, different integration strategies are needed for our proposed approach.

The encoder needs to capture contextual information by modeling input sentence. The goal of exploiting external contextual information in the encoder is for modeling input sentence better. Thus, even the knowledge distillation method with sentence level objective could be used in the encoder, the dynamic fusion mechanism is more suitable.

However, compared with the encoder, the decoder is difficult to exploit pre-trained knowledge for two reasons. First, the main role of the decoder is generating a target sentence by feeding source representation, which involves the transformation of semantic space. So, the representation from the decoder is far different from the pre-trained model. Then, the *exposure bias* (Lee, Mansimov, and Cho 2018; Wu et al. 2018) leads to that the ground-truth representation which is generated by reference is not available in the inference stage. So, we think using the knowledge distillation to learn language knowledge is a better solution, which will not influence the original goal of translation, and help to generate a better sentence.

In general, we integrate the APT framework by employing the dynamic fusion mechanism on the encoder and the knowledge distillation paradigm on the decoder. We also report the comparison of other strategies in the experiment.

4 Experiment

4.1 Implementation Detail

Data-sets We conduct experiments on the WMT datasets¹, including WMT17 Chinese to English (ZH→EN), WMT 14 English to German (EN→DE) and German to English (DE→EN) and the corresponding monolingual data.

On the ZH→EN, we use WMT17 as training set which consists of about 7.5 million sentence pairs (only CWMT part). We use `newsdev2017` as validation set which has 2002 sentence pairs, and `newstest2017` as test set which have 2001 sentence pairs. On the EN→DE and DE→EN, we use WMT14 as training set which consists of about 4.5 million sentence pairs. We use `newstest2013` as validation set which has 3000 sentence pairs, and `newstest2014` as test set which have 3003 sentence pairs.

Following Song et al. (2019), on the English and German, we use the monolingual data from WMT News Crawl. We select 50M sentence from year 2007 to 2017 for English and German respectively. Then, we choose 50M sentence from Common Crawl for Chinese.

Settings We apply byte pair encoding (BPE) (Sennrich, Haddow, and Birch 2015) to all language pairs and limit the vocabulary size to 32K.

For Transformer, we set the dimension of the input and output of all layers as 512, and that of the feed-forward layer to 2048. We employ 8 parallel attention heads. The number

¹<http://www.statmt.org/wmt17/translation-task.html>

Model	Pre-trained Model		EN→DE		DE→EN		ZH→EN	
	Encoder	Decoder	BLEU	Δ	BLEU	Δ	BLEU	Δ
Transformer (Vaswani et al. 2017)	N/A	N/A	27.3	–	N/A	–	N/A	–
Transformer (Zheng et al. 2019)	N/A	N/A	27.14	–	N/A	–	N/A	–
Transformer (Dou et al. 2018)	N/A	N/A	27.31	–	N/A	–	24.13	–
Transformer	N/A	N/A	27.31	–	32.51	–	24.47	–
w/ Fine-tuning	GPT	N/A	27.82	+0.51	33.17	+0.66	25.11	+0.64
	N/A	GPT	27.45	+0.14	32.87	+0.36	24.59	+0.12
	GPT	GPT	27.85	+0.54	32.79	+0.28	25.21	+0.74
	BERT	N/A	28.22	+0.91	33.64	+1.13	25.33	+0.86
	N/A	BERT	27.42	+0.11	33.13	+0.62	24.78	+0.31
	BERT	BERT	28.32	+1.01	33.57	+1.06	25.45	+0.98
	GPT	BERT	28.29	+0.98	33.33	+0.82	25.42	+0.95
	BERT	GPT	28.32	+1.01	33.57	+1.05	25.46	+0.99
	MASS		28.07	+0.76	33.29	+0.78	25.11	+0.64
	DAE		27.63	+0.33	33.03	+0.52	24.67	+0.20
w/ APT Framework	GPT	BERT	28.89	+1.58	34.32	+1.81	25.98	+1.51
	BERT	GPT	29.23	+1.92	34.84	+2.33	26.21	+1.74
	GPT	GPT	28.97	+1.66	34.26	+1.75	26.01	+1.54
	BERT	BERT	29.02	+1.71	34.67	+2.16	26.46	+1.99

Table 1: Translation qualities on the EN→DE, DE→EN and ZH→EN experiments.

of layers for the encoder and decoder are 6. Sentence pairs are batched together by approximate sentence length. Each batch has 50 sentence and the maximum length of a sentence is limited to 100. We use label smoothing with value 0.1 and dropout with a rate of 0.1. We use the Adam (Kingma and Ba 2014) to update the parameters, and the learning rate was varied under a warm-up strategy with 4000 steps. Other settings of Transformer follow Vaswani et al. (2017).

we also implement GPT (Radford et al. 2018), BERT (Devlin et al. 2018) and MASS (Song et al. 2019) in our Transformer system. The implementation details are as follows:

- GPT: Radford et al. (2018) proposed a pre-trained self-attention language model. We implement it on both source and target languages based on the aforementioned Transformer decoder.
- BERT: Devlin et al. (2018) proposed a pre-trained bi-directional encoder optimized by the masked token and next sentence objectives. Following Lample and Conneau (2019), we implement it only using the masked token objective, which doesn't require monolingual data has document boundary.
- MASS: Song et al. (2019) proposed a masked sequence to sequence pre-training model for text generation tasks. It masks a continuous segment from a sentence as the label, and the rest of the sentence as the input of encoder. We implement it in our Transformer system without any modification.

After the training stage, we use beam search for heuristic decoding, and the beam size is set to 4. We measure the translation quality with the NIST-BLEU (Papineni et al. 2002). We implement our approach with the in-house implementation of Transformer derived from the *tensor2tensor*².

²<https://github.com/tensorflow/tensor2tensor>

4.2 Main Results

Translation Quality The results on the EN→DE, DE→EN and ZH→EN are shown in Table 1. For a fair comparison, we also report several Transformer baseline from previous work (Vaswani et al. 2017; Zheng et al. 2019; Dou et al. 2018). Our Transformer baseline achieves similar or better results comparing with them. Compared with our baseline, Transformer with the APT framework based on different pre-trained models improves 1.92, 2.33 and 1.99 BLEU scores on the EN→DE, DE→EN and ZH→EN, respectively (**bold font**). It's worth to mention that the percentage improvement on the ZH→EN, whose difference of syntax and morphology is bigger than German and English, is more than other language pairs.

Compared with Fine-tuning We also implement the fine-tuning method with different pre-trained models. When the encoder is initialized by BERT and the decoder is initialized by BERT or GPT, the BLEU score improves about 1 point on three translation tasks. Our APT framework outperforms the fine-tuning method on all tasks whenever using BERT or GPT. This results demonstrate that the proposed approach is more effective for obtaining the knowledge from pre-trained model than fine-tuning in neural machine translation.

GPT Vs. BERT Although our work combining with GPT or BERT achieves remarkable improvements, there are several differences when employing them on encoder or decoder. First, BERT is better than GPT on the encoder when using the proposed APT framework (+0.13 to +0.48). We think the reason is that compared with the uni-directional language model of GPT, the masked language model could obtain more contextual information. While on the decoder side, GPT gets better performance than BERT due to it can

Model	BLEU
Transformer-Big (Vaswani et al. 2017)	28.46
w/ Fine-tuning (Lample and Conneau 2019)	27.70
w/ Feature (Lample and Conneau 2019)	28.70
w/ CTNMT (Yang et al. 2019a)	30.10
Transformer-Base	27.31
w/ APT framework	29.23

Table 2: The comparison of the proposed method and previous work on the EN→DE task.

Model	BLEU	Δ
Transformer	27.31	–
w/o Knowledge Distillation	28.77	+1.46
w/o Contextual Gating	28.44	+1.13
w/o Layer-aware Attention	28.39	+1.08
w/o Contextual Gating	28.03	+0.72
w/o Layer-aware Attention	28.03	+0.72
w/ Knowledge Distillation	29.23	+1.92
w/o Contextual Gating	28.91	+1.60
w/o Layer-aware Attention	28.68	+1.37
w/o Contextual Gating	28.43	+1.12
w/o Layer-aware Attention	28.43	+1.12
w/o Dynamic Fusion	28.68	+1.37
w/o Word Distillation	28.31	+1.02
w/o Sent Distillation	28.56	+1.15
w Dynamic Fusion	29.23	+1.92
w/o Word Distillation	28.76	+1.45
w/o Sent Distillation	28.87	+1.56

Table 3: Ablation study on the EN→DE task.

model sequential information which is an important factor for the decoding process.

Compared with Previous Work We also report several recent work related to use pre-trained model in NMT. The results are summarized in Table 2. When using the public BERT³ to fine-tune the Transformer-big (Vaswani et al. 2017), BLEU score decreases 0.76. However, in our implementation, the fine-tuning method improves 1.01 BLEU based on the Transformer-base. Furthermore, the feature-based approach is better than fine-tuning which contrasts other tasks (Devlin et al. 2018). This result also verified the fine-tuning doesn’t fit NMT. Our approach with *base* setting could outperform their whose parameter size is far larger than us⁴.

4.3 Ablation Study

To show the effectiveness of each module from the proposed framework, we do a detailed ablation study here. On the one hand, we show the effectiveness of the context gating and

³<https://github.com/google-research/bert>

⁴CTNMT uses the *big* setting and beam size is 8. Furthermore, the size of monolingual data they used is far larger than us. So, it’s unfair to compare us with them directly.

Model	Method	BLEU
Transformer	N/A	27.31
Encoder	w/ Dynamic Fusion	28.77
	w/ Knowledge Distillation	28.21
	w/ Dynamic Fusion Knowledge Distillation	28.69
Decoder	w/ Dynamic Fusion	27.41
	w/ Knowledge Distillation	28.68
	w/ Dynamic Fusion Knowledge Distillation	27.71

Table 4: The comparison of translation qualities for using the dynamic fusion mechanism and knowledge distillation paradigm with different strategies on the EN→DE task.

layer-aware attention from the dynamic fusion in the first two parts. Whether using knowledge distillation or not, the layer-aware attention is a bit more important than context gating. The fine-grained method of context gating could provide further improvement based on the layer-aware attention. When ablating both of them, which like a layer coordination method, the BLEU score drops about 0.7 point.

On the other hand, the word level and sentence level distillation objectives could be used individually. We also evaluate them with or without dynamic fusion mechanism. Compared with word level distillation, without sentence level distillation will have more negative influence, which reveal that learning contextual knowledge is important than only learning the output distribution.

4.4 Impact of Different Integration Strategies

In this section, we analyze the different integration strategies for our APT framework. Specifically, we employ three integration settings on the encoder and decoder, respectively. The results are summarized in Table 4.

Encoder side The different strategies employed on the encoder are shown in the first part of Table 4. Here, we only use sentence level knowledge distillation. The knowledge distillation can help the encoder to model input sentence better, while the effect is not as good as using the dynamic fusion mechanism. Moreover, the performance doesn’t improve when adopting both the knowledge distillation and the dynamic fusion. It shows that the effectiveness of knowledge distillation is covered by dynamic fusion in this scenario.

Decoder side We make a comparison on the decoder side under the same settings above mentioned. The dynamic fusion doesn’t work on the decoder side, in which pre-trained models can’t get the ground truth as a input in the decoding stage, so the task-specific representation generated by the dynamic fusion is incomplete and contains noisy. According to this experiment, the knowledge distillation is better than feature-based or fine-tuning methods in the decoder.

Model	Layers	BLEU
Transformer	N/A	27.31
w/ Dynamic Fusion	Embedding	27.94
	1st-5th	28.42
	Output	28.34
	All	28.77
w/ Knowledge Distillation	Embedding	28.21
	1st-5th	28.01
	Output	28.68
	All	28.22

Table 5: The comparison of employing dynamic fusion mechanism and knowledge distillation paradigm on different layers on the EN→DE task.

4.5 Effectiveness on Different Layers

We compare the effectiveness of employing the APT on different layers. The results are shown in Table 5. The phenomena from the encoder and decoder sides are different. On the encoder side, more layers fuse external knowledge, better performance the model achieve. Moreover, high layers can get more gain comparing with low layers. These results indicate that the dynamic fusion can improve the ability of modeling input sentence at all layers. High layers of the encoder need external contextual knowledge more than low layers to get the semantic from the input sentence.

On the decoder side, adopting knowledge distillation on the output layer obtains the best performance. And only adopting it on the embedding is better than others. We think the middle layers focus on transforming the source representation from the encoder to the target, so the external contextual knowledge doesn't help them much. On the embedding layer, the pre-trained embedding is better than the embedding of NMT. So, only fitting embedding can get a considerable improvement. The representation from output layer is used to generate the target sentence, the contextual representation learned from the pre-trained model could give it more language information to generate a better sentence.

5 Related Work

Pre-trained Model In NLP field, there are many pre-trained models have been proposed to learn the contextual information from large scale monolingual data. Peters et al. (2018) introduced Embedding learned from Bi-LSTM based Language Models (ELMo) and successfully apply it in question answering, textual entailment, etc. Inspired by them, Radford et al. (2018) proposed to use self-attention network based language model (GPT) to replace the Bi-LSTM structure, which further improves the performance of the pre-trained model. Then, Devlin et al. (2018) proposed to use the bi-directional encoder representation from Transformer (BERT) with a masked token objective and a next sentence objective to capture global contextual information. After that, several varieties are proposed, like MASS (Song et al. 2019), XLNet (Yang et al. 2019b), etc.

These pre-training methods attended the state-of-the-art in several tasks by fine-tuning the pre-trained parameters

with labeled data from downstream tasks. However, fine-tuning these advanced model by parallel data can not work well in NMT, because of the gap between the bilingual and monolingual tasks. Our APT framework can erase this gap and fully release the potential of these model in NMT filed.

Exploiting Monolingual Data for NMT Several successful attempts have been made to utilize monolingual data in NMT directly. Sennrich, Haddow, and Birch (2016) proposed to use back-translation to generate synthetic parallel data from monolingual data. Currey, Barone, and Heafield (2017) proposed a copy mechanism to copy fragments of sentences from monolingual data to translated outputs directly. Zhang et al. (2018) proposed to jointly train the source-to-target and target-to-source NMT models with the pseudo parallel data from monolingual data. However, these studies only focus on the usage of word or surface information, the rich contextual information from large scale monolingual data does not be fully utilized.

Then, some researchers also pay attention to apply the pre-trained model for NMT. Di Gangi and Federico (2017) use source side pre-trained embedding and integrate it into NMT with a mix-sum/gating mechanism. They only focus on utilizing the pre-trained embedding, leaving the underlying linguistic information ignored. Ramachandran, Liu, and Le (2017) firstly proposed the fine-tuning method in NMT. However, the general information can not be used in NMT directly leading to the information from pre-trained models is less exploited. Our approach can fuse the adapted task-specific representation into NMT and distill knowledge from pre-trained model to NMT in the training process, exploiting the pre-trained knowledge effectively.

6 Conclusion

In this paper, we first address the problem that the contextual knowledge from pre-trained models can not be used well with the fine-tuning method in NMT, due to the large gap between the bilingual machine translation task and monolingual pre-trained models. Then, we propose an APT framework, which could fuse the task-specific representation adapted from general representation by a dynamic fusion mechanism and learn the contextual knowledge from pre-trained models by a knowledge distillation paradigm. Moreover, we conclude the best strategy for using the two methods in the encoder-decoder framework. Experiments on three machine translation tasks show that the proposed APT framework achieves prominent improvements by fully acquiring the knowledge from pre-trained models to NMT.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments. Thanks to Shaohui Kuang and Wei Zou for their insightful comments. Shujian Huang is the corresponding author. This work is supported by the National Science Foundation of China (No. U1836221, 61672277) and the Jiangsu Provincial Research Foundation for Basic Research (No. BK20170074).

References

- Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.
- Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*.
- Chen, Y.; Liu, Y.; Cheng, Y.; and Li, V. O. 2017. A teacher-student framework for zero-resource neural machine translation. In *ACL*.
- Cho, K.; van Merriënboer, 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*.
- Currey, A.; Barone, A. V. M.; and Heafield, K. 2017. Copied monolingual data improves low-resource neural machine translation. In *MT*.
- Deng, Y.; Cheng, S.; Lu, J.; Song, K.; Wang, J.; Wu, S.; Yao, L.; Zhang, G.; Zhang, H.; Zhang, P.; et al. 2018. Alibaba’s neural machine translation systems for wmt18. In *Conference on Machine Translation: Shared Task Papers*.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv*.
- Di Gangi, M. A., and Federico, M. 2017. Can monolingual embeddings improve neural machine translation? *Proc. of CLiC-it*.
- Dou, Z.-Y.; Tu, Z.; Wang, X.; Shi, S.; and Zhang, T. 2018. Exploiting deep representations for neural machine translation. *arXiv*.
- Gehring, J.; Auli, M.; Grangier, D.; and Dauphin, Y. N. 2016. A convolutional encoder model for neural machine translation. *arXiv*.
- Gulcehre, C.; Firat, O.; Xu, K.; Cho, K.; Barrault, L.; Lin, H.-C.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2015. On using monolingual corpora in neural machine translation. *arXiv*.
- He, T.; Tan, X.; Xia, Y.; He, D.; Qin, T.; Chen, Z.; and Liu, T.-Y. 2018. Layer-wise coordination between encoder and decoder for neural machine translation. In *NIPS*.
- Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization.
- Kuang, S.; Xiong, D.; Luo, W.; and Zhou, G. 2018. Modeling coherence for neural machine translation with dynamic and topic caches. In *Coling*.
- Lample, G., and Conneau, A. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.
- Lee, J.; Mansimov, E.; and Cho, K. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. *arXiv*.
- Luong, M.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*.
- Mikolov, T.; Le, Q. V.; and Sutskever, I. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.
- Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: A method for automatic evaluation of machine translation. In *ACL*.
- Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. *arXiv*.
- Poncelas, A.; Shterionov, D.; Way, A.; Wenniger, G. M. d. B.; and Passban, P. 2018. Investigating backtranslation in neural machine translation. *arXiv*.
- Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. *CoRR*.
- Ramachandran, P.; Liu, P.; and Le, Q. 2017. Unsupervised pretraining for sequence to sequence learning. In *EMNLP*.
- Sennrich, R.; Haddow, B.; and Birch, A. 2015. Neural machine translation of rare words with subword units. *arXiv*.
- Sennrich, R.; Haddow, B.; and Birch, A. 2016. Improving neural machine translation models with monolingual data. In *ACL*.
- Song, K.; Tan, X.; Qin, T.; Lu, J.; and Liu, T.-Y. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *ICML*.
- Sun, M.; Jiang, B.; Xiong, H.; He, Z.; Wu, H.; and Wang, H. 2019. Baidu neural machine translation systems for wmt19. In *MT*.
- Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*.
- 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, Q.; Li, F.; Xiao, T.; Li, Y.; Li, Y.; and Zhu, J. 2018. Multi-layer representation fusion for neural machine translation. In *COLING*.
- Wu, L.; Tan, X.; He, D.; Tian, F.; Qin, T.; Lai, J.; and Liu, T.-Y. 2018. Beyond error propagation in neural machine translation: Characteristics of language also matter. *arXiv*.
- Wu, P.; Huang, S.; Weng, R.; Zheng, Z.; Zhang, J.; Yan, X.; and Chen, J. 2019. Learning representation mapping for relation detection in knowledge base question answering. In *ACL*.
- Yang, J.; Wang, M.; Zhou, H.; Zhao, C.; Yu, Y.; Zhang, W.; and Li, L. 2019a. Towards making the most of bert in neural machine translation. In *EMNLP*.
- Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; and Le, Q. V. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.
- Zhang, J., and Zong, C. 2016. Exploiting source-side monolingual data in neural machine translation. In *EMNLP*.
- Zhang, Z.; Liu, S.; Li, M.; Zhou, M.; and Chen, E. 2018. Joint training for neural machine translation models with monolingual data. *arXiv*.
- Zheng, Z.; Huang, S.; Tu, Z.; Dai, X.; and Chen, J. 2019. Dynamic past and future for neural machine translation. In *EMNLP*.