Encoder and Decoder, Not One Less for Pre-trained Language Model Sponsored NMT

Sufeng Duan[1,2] **and Hai Zhao**[1,2*]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
1140339019dsf@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

Abstract

Well pre-trained contextualized representations from pre-trained language models (PLM) have been shown helpful for enhancing various natural language processing tasks, surely including neural machine translation (NMT). However, existing methods either consider encoder-only enhancement or rely on specific multilingual PLMs, which leads to a much larger model or give up potentially helpful knowledge from target PLMs. In this paper, we propose a new monolingual PLM-sponsored NMT model to let both encoder and decoder enjoy PLM enhancement to alleviate such obvious inconvenience. Especially, incorporating a newly proposed frequency-weighted embedding transformation algorithm, PLM embeddings can be effectively exploited in terms of the representations of the NMT decoder. We evaluate our model on IWSLT14 En-De, De-En, WMT14 En-De, and En-Fr tasks, and the results show that our proposed PLM enhancement gives significant improvement and even helps achieve new state-of-the-art.

1 Introduction

Pre-trained language models (PLM) such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and XLNET (Yang et al., 2019) have significantly improved a wide range of natural language processing (NLP) tasks. However, neural machine translation (NMT) tasks often require big models and large scale of training data, which leads to fine-tuned PLMs easily forgetting supposed-not-to-be-forgotten knowledge. As a result, NMT model cannot yield promising results only by fine-tuning PLMs according to extensive studies (Zhu et al., 2020; Yang et al., 2020).

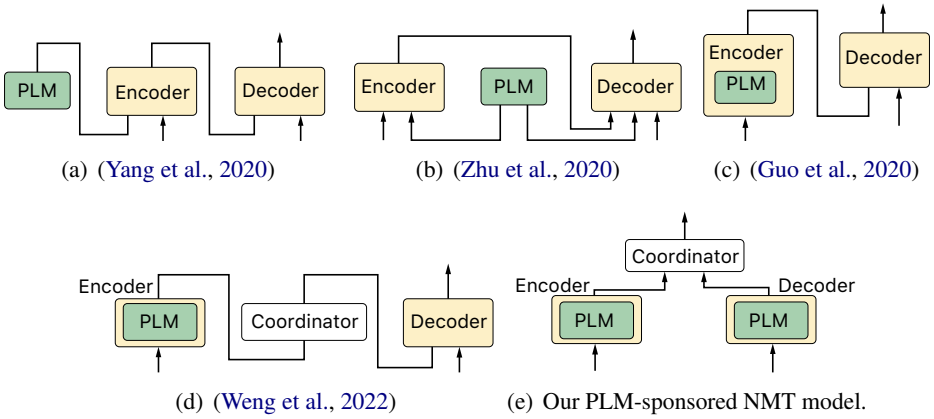
To make effective use of PLMs in NMT, there has been a series of explorations. Yang et al. (2020) proposed CTNMT based on asymptotic distillation to keep reminding the NMT model of BERT knowledge and dynamic switching gate to combine the encoded embedding from BERT and the encoder of NMT model. Zhu et al. (2020) proposed BERT-fused model using BERT as context-aware embedding for NMT, which exploits the representation from BERT and feeds it into every layer of the encoder. AB-Net (Guo et al., 2020) incorporates adapter layers into each BERT layer to conduct non-auto-regressive and auto-regressive decoding of sequence-to-sequence models for improvement. Weng et al. (2022) propose a framework to fuse BERT into NMT model using a layer-wise coordination structure with a partitioned multi-task learning method. In this paper, we also focus on incorporating BERT into NMT model.

These works do achieve improvement on NMT tasks with PLMs more or less while leaving huge challenges behind. For example, Yang et al. (2020); Zhu et al. (2020) simply put PLMs as a plugin for NMT model, which leads to oversized models. To avoid forgetting PLMs knowledge, Guo et al. (2020) completely abandon fine-tuning by simply freezing PLM parameters. Besides, Zhu et al. (2020); Weng et al. (2022) show that using monolingual PLMs for decoder initialization may hurt NMT performance so that NMT decoder cannot conveniently benefit from PLM enhancement.

In existing NMT models using PLMs (Yang et al., 2020; Zhu et al., 2020; Guo et al., 2020; Weng et al., 2022), NMT is enhanced by PLMs in the same way as other NLP tasks: PLMs offer representations first and then the model makes a prediction based on such representations. However, the behavior of NMT differs from other NLP tasks in a small but vital detail. Most NLP tasks simply perform in a strict pipeline of first-encoding-then-decoding way, while NMT models cannot perform similarly as other tasks in which the NMT decoder has to encode predicted target words and

*Corresponding author. This work was supported by the Key Projects of National Natural Science Foundation of China (U1836222 and 61733011).

Findings of the Association for Computational Linguistics: ACL 2023, pages 3602–3613
July 9-14, 2023



(a) (Yang et al., 2020) (b) (Zhu et al., 2020) (c) (Guo et al., 2020)

(d) (Weng et al., 2022) (e) Our PLM-sponsored NMT model.

Figure 1: Structures of some existing autoregressive models and our model. All structures shown in figures are used for prediction. Note that the model in Figure 1(c) is an autoregressive model (Guo et al., 2020).

predict target sequence simultaneously. It means that the NMT decoder has to take the responsibility of encoding target words at the same time. Therefore, there is not so strict a first-encoding-then-decoding pipeline in NMT models like other NLP tasks, which makes fine-tuning PLM for NMT decoder hardly achieve success (Guo et al., 2020).

In light of such an observation on the difference between NMT and other NLP tasks, we propose a novel design to let both encoder and decoder of NMT model enjoy the empowerment of PLM for the first time. For the decoder part, we reorganize the decoder by replacing all self-attention sub-layers with PLMs to encode predicted target sentences and moving all cross-attention sub-layers to a novel **Coordinator**. Our decoder consisting of self-attention sub-layers has the same structure as the encoder which can be replaced with PLMs completely without any structure change and focuses on encoding the representations of predicted words. The novel coordinator views the encoder and decoder as equal and combines representations from the encoder and decoder for the prediction of unknown words. In our model, the encoder and decoder can be replaced by PLMs completely and the coordinator is randomly initialized, the same as other downstream tasks in which randomly initialized parameters are light and the PLMs are heavy. Our model design facilitates PLMs of source and target to focus on encoding source and target sentences respectively which are the original roles of PLMs.

We evaluate our model on four NMT tasks. Experimental results show that our model can achieve improvement of all tasks significantly using PLMs. Our model achieves 37.5 BLEU score on IWSLT14 German-English task outperforming other works, 31.0 and 44.5 BLEU score on WMT14 English-German and English-French tasks with fewer parameters compared to other works. We also give some ablations to evaluate the effectiveness of different methods in our models, and ablations show that PLMs indeed achieve the performance of the NMT model from both sides of the encoder and decoder.

2 Our Model

In this section, we introduce the architecture of our PLM-sponsored NMT model and the training objective.

2.1 Background

2.1.1 Transformer

As the state-of-the-art NMT model, Transformer(Vaswani et al., 2017) is the first self-attention-based model and is based on scaled dot-product attention:

$$Attention(Q, K, V) = \text{softmax}(\frac{Q^T K}{\sqrt{d_k}}) \cdot V, \quad (1)$$

where Q,K, V, and d_k are the query, key, value vectors, and dimension of the model respectively.

The encoder is composed of a stack of N identical layers. An encoder layer consists of a self-attention sub-layer and a feed-forward sub-layer. The model employs a residual connection (He et al., 2016) around each of the two sub-layers followed by layer normalization (Ba et al., 2016).

The decoder of Transformer is also composed of a stack of N identical layers while a decoder

layer has another cross-attention sub-layer following self-attention to incorporate source representations into representations of target sentence. The decoder uses masking in the self-attention to ensure that the predictions cannot use unknown outputs. Without recurrence for the order of sequence, the Transformer uses position encoding (Gehring et al., 2017) to mark the position.

2.1.2 Modes of Incorporating PLMs

Figure 1(a), 1(b), 1(c), and 1(d) show structures of some existing works. They show that existing works focus on incorporating PLMs into the encoder and randomly initialize the decoder which makes the model focus on training parameters in the decoder. Zhu et al. (2020) is different from other works in which representations from PLMs are incorporated into both encoder and decoder while it can only encode source language. Besides, these works show that the encoding of target words depends on the encoding of source words because decoders must use representations from the encoder to generate representations for prediction. It also makes a heavy randomly initialized part and a light PLM.

2.2 PLM-sponsored Architecture

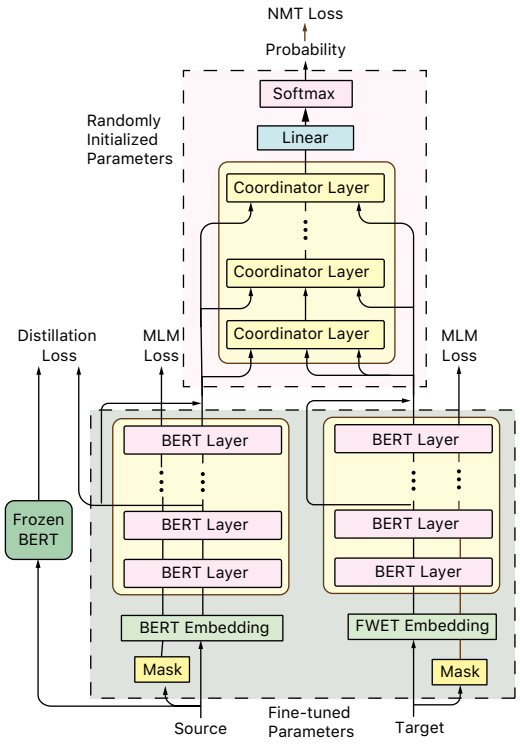


Figure 2: The proposed PLM-sponsored model.

Focus on the self-attention sub-layer in the top decoder layer and we can regard the previous decoder layers as a language model to generate the representation of the predicted target sentence. Furthermore, focus on the cross-attention sub-layer and we can regard the encoder and other decoder layers as two language models to generate representations of source and target language. With only one decoder layer, the model can be split into three parts: encoder for the source sentence, self-attention sub-layer for the predicted target sentence, and cross-attention sub-layer to combine representations of source and target sentences.

Based on the analysis above, we re-organize the decoder in which all self-attention sub-layers are split into our new decoder and all cross-attention sub-layers are split into our new coordination structure. Figure 2 shows the architecture of our model. Our decoder is a stack of self-attention sub-layers to encode predicted words with the same structure as the encoder, and the coordination structure called coordinator is a stack of cross-attention sub-layers. Different from existing works, our model can replace both encoder and decoder with PLMs and fine-tune all parameters.

Inspired by He et al. (2018), we adopt the coordinator which uses coordination as cross-attention to combine the source and target representations. Given source sentence $S = \{s_1, ..., s_m\}$ and target sentence $T = \{t_1, ..., t_n\}$, the representations of S and T generated by PLMs are r^S and r^T, respectively. r^S and r^T are fed to every coordinator layer as equal input. The j-th coordinator layer generates representation r_j by the coordination:

$$\begin{aligned} r_{input} &= \text{concat}(r^S, r^T), \\ r_{all} &= \text{concat}(r_{input}, r_{j-1}), \\ h_j &= \text{Attention}(r_{j-1}, r_{all}, r_{all}) + r_{j-1}, \\ r_j &= \text{LN}(\text{FFN}(h_j) + h_j). \end{aligned} \tag{2}$$

In the first layer, we directly use r^T as r_0. To mark the position, we add position embeddings to r_0.

Although we can use BERT to initialize parameters of the decoder, the prediction in our model is still uni-directional same as Transformer decoder. To ensure the word can only access known words, we use a special mask $M \in \mathbb{R}^{n \times (m+2n)}$ to process the attention-matrix generated by $\text{Attention}()$. The first part of the mask is $M_S \in \mathbb{R}^{n \times m}$ and used for R_S, and all values in M_S are 0. The second and third part $M_T, M_h \in \mathbb{R}^{n \times n}$ are upper triangular

matrices. The M is generated by

$$M = \text{concat}(M_S, M_T, M_h) \quad (3)$$

We use M to mask the attention matrix by setting values to $-\infty$ if the value in the corresponding position of M is 1.

Existing works only fine-tune parameters of the encoder while have to train the parameters of the decoder from the randomly initialized, which makes a light PLM and heavy randomly initialized parameters. In our model, both encoder and decoder are initialized by PLMs and fine-tuned, which makes the training more balance compared to other works. Our coordinator views source and target PLMs as equal, which learns the cross-lingual knowledge and fine-tunes the source and target PLMs to reduce the negative effect of fine-tuning source PLMs. Besides, because we can incorporate PLMs into the encoder and decoder without structure change, we can train the decoder with MLM loss to learn knowledge from monolingual training data similar to the encoder.

2.3 Half-layers Knowledge Distillation

The *catastrophic forgetting* problem, which is caused by fine-tuning PLMs with a large scale of training data, makes the model forget linguistic knowledge in PLMs and hurts the performance. To alleviate such a drawback, we adopt knowledge distillation training by following Yang et al. (2020) for the encoder.

Different from Yang et al. (2020) which trains all layers with distillation, we only train the first half of the encoder layers to learn knowledge of BERT which can also make the other layers learn knowledge from training data. Same as Yang et al. (2020), the training objective of the distillation of our model is to penalize the mean-squared-error loss between the hidden states of the first half of the layers and the representations from BERT:

$$\mathcal{L}_{kd} = -||r_{plm} - h_{M/2}||^2, \quad (4)$$

where M is the number of layers, r_{plm} is the representation generated by BERT, and $h_{M/2}$ is the hidden state of the first half of encoder layers. To balance different training objectives during training, we use the same strategy as Yang et al. (2020) to train the model:

$$\mathcal{L}'_{NMT} = \alpha \cdot \mathcal{L}_{\mathcal{NMT}} + (1-\alpha) \cdot \mathcal{L}_{kd}, \quad (5)$$

where $\mathcal{L}_{\mathcal{NMT}}$ is the sum of other losses, α is a hyper-parameter setting to 0.9 in this paper. BERT to generate r_{plm} is frozen as the teacher model and will not be used for evaluation.

Different from Yang et al. (2020); Zhu et al. (2020) which incorporates representations from a parallel BERT, we incorporate hidden states of the first half of layers into the final representations using gate and FFN:

$$\begin{aligned} g^S &= \text{Sigmoid}(\text{concat}([h^S_M, h^S_{M/2}])W), \\ h^S &= g^S \cdot h^M + (1-g^S) \cdot h^{M/2}, \\ r^S &= \text{LN}(\text{FFN}(h^S) + h^S), \end{aligned} \quad (6)$$

where $W \in \mathbb{R}^{(2d_m) \times d_m}$ is parameter matrices and d_m is the dimension of the representation. With our method, the model can extract representations of BERT from the encoder directly instead of BERT. The frozen BERT can be removed from the model during evaluation to decrease the number of parameters.

2.4 Frequency-weighted Embedding Transformation

Both current PLMs and NMT models usually adopt subword tokenization for word segmentation, for which BPE (Sennrich et al., 2016) style segmentation has been popularly adopted so far. Although previous works use embeddings of PLMs in NMT models directly, we argue that it is necessary to pay attention to the differences between embeddings of PLMs and NMT models. To use PLMs in the NMT model without considering the effect of different segmentation of sentences, we design a frequency-weighted embedding transformation (FWET) algorithm to represent NMT embeddings from PLM embeddings according to contextualized subword frequency by comparing sequence segmentation. Note that BPE segmentation indeed performs over words, however, our proposed algorithm has to exploit contextualized statistics over sentences to build the transformation vector. Algorithm 1 shows the algorithm of our method. This method is a pre-processing method which means that it will not influence the training.

2.5 Training Objective

Given a source sentence $S = \{s_1, ..., s_m\}$ as input, the NMT model needs to predict the target sentence $T = \{t_1, ..., t_n\}$ from left to right by using representations of S. The training objective of NMT is

Algorithm 1 Frequency-weighted Embedding Transformation

Input: Corpus $\mathbf{S} = \{S_1, ..., S_k\}$, PLM Embedding E_P

Output: NMT Embedding E

Create real value vector dic^N and let all values in dic^N to 0.
for Sentence S_j in **S do**
  Tokenizing S_j for NMT model, $S_j^N = \{s_{j,1}^N, ..., s_{j,x}^N\}$
  Tokenizing S_j for PLMs, $S_j^P = \{s_{j,1}^P, ..., s_{j,y}^P\}$
  while $len(S_j^N) > 0$ and $len(S_j^P) > 0$ **do**
    if $s_{j,1}^N$ == $s_{j,1}^P$ **then**
      $dic^N(s_{j,1}^N, s_{j,1}^P) += 1$
      Remove $s_{j,1}^N$ and $s_{j,1}^P$ from S_j^N and S_j^P respectively.
    else if $s_{j,last}^N$ == $s_{j,last}^P$ **then**
      $dic^N(s_{j,last}^N, s_{j,last}^P) += 1$
      Remove $s_{j,last}^N$ and $s_{j,last}^P$ from S_j^N and S_j^P respectively.
    else
      Break
    end if
  end while
  if $len(S_j^P) == 0$ or $len(S_j^N) == 0$ **then**
    Continue
  end if
  Let weight $w = len(S_j^P)/len(S_j^N)$
  for $S_{j,l}^N$ in S_j^N **do**
    for $S_{j,q}^P$ in S_j^P **do**
      $dic^N(s_{j,l}^N, s_{j,q}^P) = dic^N(s_{j,l}^N, s_{j,q}^P) + w$
    end for
  end for
end for
for k in set of dic^N **do**
  $E_N(k) = Softmax(dic^N(k))E_P(k)$
end for
RETURN Embedding $E = E_N$

to minimize the negative log-likelihood as:

$$\mathcal{L}_{\mathcal{NMT}} = -\sum_{j}^{n} \log(p(t_j|t_1, ..., t_{j-1}, s_1, ..., s_m)) \tag{7}$$

Existing NMT models often use the loss in Eq 7 to train the model to predict unknown words. However, using PLMs in the NMT model requires other losses to fine-tune the model. The encoders of the NMT model can be viewed as language models which can generate representations with linguistic information of the corresponding language. Therefore, we can also train encoders using the same way as PLMs to learn corpora. We use two different training objectives for the encoder and decoder:

Masked language model (MLM) (Devlin et al., 2019) for the encoder. The encoder has the same structure as BERT which makes the encoder learn lingual information from corpora using MLM. To avoid hurting the performance of NMT tasks, we use the masked sentences for MLM and the original sentence for NMT tasks.

Unknown word prediction MLM PLM for the decoder is supposed to predict the unknown word using the known contextual representation without source representation. Given a target sentence $T = \{t_1, ..., t_n\}$, the loss for the unidirectional encoder is to minimize the negative log-likelihoods as

$$\mathcal{L}_T = -\sum_{j}^{n} \log(p(t_j|t_1, ..., t_{j-1})) \tag{8}$$

To avoid overfitting of PLM in the decoder, we choose only 10% words to calculate $\mathcal{L}_T$.

The training function of the model is

$$\mathcal{L} = (\alpha \cdot \mathcal{L}_{\mathcal{NMT}} + (1-\alpha) \cdot \mathcal{L}_{kd}) + \mathcal{L}_{MLM} + \mathcal{L}_T \tag{9}$$

3 Experiment

3.1 Datasets

Our model is evaluated on four tasks, IWSLT14 English-German (IWSLT En-De), German-English (De-En), WMT14 English-German (WMT En-De), and English-French (En-Fr).

IWSLT14 dataset contains 153K training sentence pairs. We use script[1] to preprocess the dataset, and use 7K data from the training set as validation set and the combination of dev2010, dev2012, tst2010, tst2011 and tst2012 as test set with 7K sentences.

WMT14 En-De dataset contains 4.5M sentence pairs for training. We use 7K from training set as validation set, and newstest2014 as test set. We use script[2] to preprocess En-De dataset. The sentences longer than 250 are removed from the training dataset.

[1] https://github.com/pytorch/fairseq/blob/master/examples/translation/prepare-iwslt14.sh

[2] https://github.com/pytorch/fairseq/blob/master/examples/translation/prepare-wmt14en2de.sh

Models	WMT14				IWSLT14			
	En-De	En-Fr	#Param	%	En-De	De-En	#Param	%
TF-base (Vaswani et al.)	27.3	38.9	66M	22%	28.5	34.1	39M	17%
TF-big (Vaswani et al.)	28.3	41.1	220M	75%	-	-	-	-
BERT-fused (Zhu et al.)	30.7	43.8	500M	172%	30.5	36.1	500M	208%
CTNMT (Yang et al.)	30.1	42.3	330M	113%	-	-	-	-
AB-Net (Guo et al.)	30.6	43.6	420M	144%	30.0$^\flat$	36.7$^\flat$	266M	111%
Weng et al.	30.2$^\flat$	43.3$^\flat$	370M	127%	29.0$^\flat$	35.0$^\flat$	266M	111%
PLM-sponsored Model	31.0$^{*\flat}$	44.5$^{*\flat}$	291M	100%	30.2$^{*\flat}$	37.5$^{*\flat}$	240M	100%

Table 1: Comparison of our PLM-sponsored NMT model and other existing works on IWSLT14 and WMT14 tasks. AB-Net is the model with an auto-regressive decoder same as Guo et al. (2020). Results with $^\flat$ indicate the results obtained by our implementation. Results with * present statistically significant differences ($p < 0.05$). Note that TF is short for Transformer.

WMT14 En-Fr dataset contains 36M sentence pairs for training. We use 26K data from training set as validation set, and newstest2014 as test set. We use script [3] for En-Fr. The sentences longer than 250 are removed from the training dataset.

For languages using embeddings of PLMs, we tokenize words into wordpiece tokens using the same vocabulary as BERT, and the vocabularies sizes for English and German are 30K. For languages using embeddings and FWET, we use BPE algorithm (Sennrich et al., 2016) to process words into subwords, and the number of subword tokens is 10K, 30K and 40K for IWSLT14, WMT14 En-De and WMT14 En-Fr.

3.2 Model Configurations

The PLMs used for our two encoders are `bert-base` models with 12 layers, 12 self-attention heads, 768 dimensions of embedding size, and 3072 dimensions of feed-forward layers. On IWSLT14 tasks, the PLM for English and German are `bert-base-uncased` and `dbmdz/bert-base-german-uncased` respectively. On WMT14 En-De task, we use `bert-base-cased` for English and `bert-base-german-cased` for German. On WMT14 En-Fr task, we use `bert-base-cased` for English and `bert-base-multilingual-cased` for French. All PLMs are from huggingface[4].

The coordinator layer of our model always has the same model configuration as Transformer-base on three tasks. Especially, the coordinator has 12 layers in the model with PLMs. The dropout for PLMs is 0.5 and 0.3 on IWSLT14 and WMT tasks.

3.3 Training Settings

Our model[5] is implemented based on fairseq-0.9.0[6] and transformers[7] (Wolf et al., 2020). Our models are trained on one CPU (Intel i7-5960X) and two nVidia RTX 3090 GPUs.

We follow the Transformer and use Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$ for our model. For all tasks, we use the learning rate setting strategy implemented by fairseq. The initial learning rate and label smoothing rate are 10^{-7} and 0.1. For IWSLT14 and WMT14 tasks, the learning rates are 0.0005 and 0.0007, the update steps are 100,000 and 2,000,000, and the warmup steps are 4000 and 8000 respectively. The batch size is 4096 for all tasks.

To evaluate our model, we use the beam search decoder for IWSLT14 tasks with beam width 5. For WMT14 task, we follow the Transformer and use the beam search decoder with beam width 4 and the length penality $\alpha = 0.6$. We use the tokenzied case-insensitive BLEU scores for IWSLT14 tasks and tokenized case-sensitive BLEU scores for WMT14 task. BLEU scores is tested by *bootstrap-hypothesis-difference-significance.pl*[8]. We use Transformer as the baseline model. For fair comparison, we re-implement AB-Net (Guo et al., 2020) on IWSLT14 tasks and Weng et al. (2022) on all tasks, and compare our model with other existing

[3] https://github.com/pytorch/fairseq/blob/master/examples/translation/prepare-wmt14en2fr.sh

[4] https://huggingface.co/

[5] Paper code: https://github.com/akibcmi/PLM-NMT

[6] https://github.com/facebookresearch/fairseq

[7] https://github.com/huggingface/transformers

[8] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/analysis/bootstrap-hypothesis-difference-significance.pl

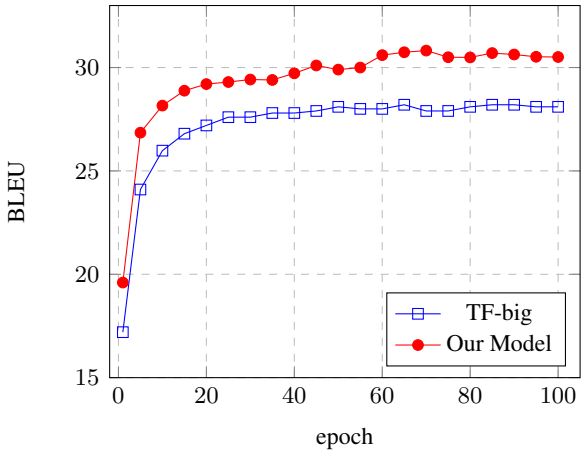


Figure 3: The BLEU score of the Transformer-big and our model on the WMT14 En-De.

works.

3.4 Results

Table 1 shows the results of our model on four NMT tasks. The baseline is Transformer (Vaswani et al., 2017). Our model gets 30.2 and 37.5 BLEU scores on IWSLT14 En-De and De-En tasks, respectively, which outperforms the baseline 1.7 and 3.4 BLEU scores improvements. It shows that our model does improve the performance of poor-resource NMT tasks using PLMs. Our model also gets 31.0 and 44.5 BLEU scores on WMT14 En-De and En-Fr tasks respectively, which outperforms the baseline 2.7 and 3.4 BLEU scores improvements. It shows that our model can also improve the performance of rich-resource NMT tasks with PLMs.

Compared to other models, our model can outperform other works on IWSLT14 De-En with 37.5 BLEU points and get comparable results on IWSLT14 En-De, WMT14 En-De, and WMT14 En-Fr. However, our model uses only monolingual $\text{BERT}_{\text{BASE}}$ (except for the French) while other works often use $\text{BERT}_{\text{LARGE}}$ with more parameters or multilingual BERT with multilingual knowledge. It shows that the improvement of performance achieved by our model is mainly from the structure and method instead of the amount of parameters, which shows the effectiveness of our model. Figure 3 compares the BLEU scores between our model and the Transformer-big and shows that our model can achieve the same results quicker than the Transformer-big.

Models	De-En	Param	En-De	Param
TF-base	34.1	43M	27.3	66M
Our model	34.3	43M	27.2	67M

Table 2: The results of our model without PLMs on IWSLT14 De-En and WMT14 En-De tasks.

4 Ablation

In this section, we perform ablation experiments over a number of facets of our model on IWSLT14 De-En and WMT14 En-De tasks. To finish our ablation experiments with limited resources, the update steps for De-En and En-De are 100,000 and 400,000 respectively, and the coordinator for En-De has 6 layers only to speed up the training.

4.1 Effectiveness of Structure

For the effectiveness of our structures, we evaluate our model without PLMs. In this experiment, our model uses the same configuration as Transformer-base. We only use $\mathcal{L}_{NMT}$ to train our model. We only use representations generated by the top encoder layer and self-attention decoder layer to feed to coordination layers. Table 2 shows that our model can achieve comparable results compared to the baseline, which shows that our model architecture does not hurt the performance, and the improvements are from PLMs and our training objectives.

4.2 Effectiveness of Training Objectives

For the effectiveness of different training objectives, we evaluate our models with different losses. In different experiments, we remove different losses to train our model and evaluate on two tasks. Table 3 shows the results of our model on IWSLT14 De-En and WMT14 En-De with different losses for training. Our model performs worst with $\mathcal{L}_{NMT}$ only. Table 3 shows that MLM is more important for the model compared with knowledge distillation on the encoder. It also shows that all our training objective is necessary for our model. We also replace our half-layers knowledge distillation with a full-layers knowledge distillation and show the result in the last line in Table 3 which shows that our half-layers method is better than a full-layers method.

4.3 Effectiveness of Embedding Transformation

Table 5 shows the results of our model using PLM embedding directly and using embedding pro-

Models	De-En	En-De
Our Model	37.5	30.5
-$\mathcal{L}_{MLM}$	36.8	30.1
-$\mathcal{L}_{T}$	36.7	29.9
-$\mathcal{L}_{kd}$	36.9	30.2
-$\mathcal{L}_{MLM}$&$\mathcal{L}_{kd}$	36.6	29.5
-$\mathcal{L}_{MLM}$&$\mathcal{L}_{T}$&$\mathcal{L}_{kd}$	36.2	28.8
+$\mathcal{L}_{fkd}$-$\mathcal{L}_{kd}$	36.8	29.9

Table 3: The results of our model with different losses on IWSLT14 De-En and WMT14 En-De.

Models	De-En	En-De
Our Model	37.5	30.5
No PLM (Encoder)	35.0	28.9
No PLM (Decoder)	36.5	29.2
No PLM	34.7	28.6
TF-base	34.1	27.3
TF-big	-	28.3

Table 4: The results of our model by removing PLMs from the encoder or decoder on IWSLT14 De-En and WMT14 En-De.

cessed by our FWET. It shows that the performance of our model drops significantly on WMT14 En-De without embedding transformation while the performance of IWSLT14 De-En drops 0.5 BLEU scores. It shows that the embedding transformation does improve the performance of NMT tasks while it also means that the embedding transformation is more important for rich-resource NMT tasks.

Models	De-En	En-De
PLM Embedding	37.0	29.5
+FWET	37.5	30.5

Table 5: The results of our model with embedding transformation on IWSLT14 De-En and WMT14 En-De.

4.4 Effectiveness of PLMs in the Different Components

For the effectiveness of PLMs in different parts of our model, we evaluate our model by removing PLMs from the encoder or decoder. Table 4 shows the results of the evaluation. It shows that the performance of our model drops significantly on both tasks by removing PLMs from one part, and our model performs worst without PLMs in the encoder and decoder. It also shows that the model without PLMs in the decoder performs better than the model without PLMs in the encoder, which shows that the PLMs are more important for the encoder than the decoder.

5 Related Work

Various pre-trained language models (Peters et al., 2018; Devlin et al., 2019; Dai et al., 2019; Liu et al., 2019; Lan et al., 2020) have improved the performance of various NLP tasks, which makes designing and training PLM a popular task. It is common to train a PLM for NMT tasks. Conneau and Lample (2019) proposed two methods to learn cross-lingual language models to provide improvements on cross-lingual classification and NMT tasks. Liu et al. (2020) proposed mBART which is a multilingual seq2seq Transformer based denoising auto-encoder and can achieve improvement of performance on poor and medium resources NMT tasks. Chen et al. (2021) proposed SixT model to directly translate unseen languages using two-stage training schedules. Xu et al. (2021) proposed BiBERT which is a bilingual pre-trained language model and can improve the performance on several different NMT tasks. Weng et al. (2019) trained a bi-directional self-attention language model to generate representations and proposed two different mechanisms to fuse representations into the encoder and decoder respectively. Some works also incorporate representations of PLM into NMT model for improvements of performance. Imamura and Sumita (2019) proposed a two-stage optimization to fine-tune BERT for NMT model. Yang et al. (2020) proposed a framework of knowledge distillation and dynamic switch to incorporate BERT into NMT model. Zhu et al. (2020) proposed a BERT-fused model which uses BERT as contextual embedding to extract representations for sentences and incorporate representations into each layer of the encoder and the decoder through attention mechanisms. Weng et al. (2022) proposed a framework to deep fuse the representations into NMT by replacing the encoder with BERT and using a layer-wise coordination structure. AB-Net (Guo et al., 2020) is based on a parallel sequence decoding algorithm to improve the performance of non-autoregressive and autoregressive NMT model.

6 Conclusion

To improve NMT by incorporating pre-trained contextualized representations from PLM during training, we propose a new monolingual PLM-sponsored NMT model to enhance both encoder

and decoder by replacing the randomly initialized encoder and decoder with BERT. Besides, we also propose a frequency-weighted embedding transformation algorithm to effectively exploit PLM embeddings in terms of the representations of NMT decoder. All parameters including PLMs can be trained together with MLM on both encoder and decoder instead of training PLM using NMT data first. We evaluate our model on four NMT tasks,and the experiment results show that our model with PLMs can improve the performance of NMT model. Compared to other works, our model can outperform or achieve comparable results using smaller and monolingual BERT with fewer parameters, which also shows the effectiveness of our model. We also evaluate our model with different model configurations and study the characteristic of our model.

7 Limitations

Although our model can achieve better results compared to other works, there are some limitations of our model:

- Our model cannot use shared dictionary and embedding on the encoder and decoder.
- Though the encoder and decoder can use PLMs, the coordinator cannot use PLM.
- The half-layers knowledge distillation still uses additional frozen BERT.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Guanhua Chen, Shuming Ma, Yun Chen, Li Dong, Dongdong Zhang, Jia Pan, Wenping Wang, and Furu Wei. 2021. Zero-shot cross-lingual transfer of neural machine translation with multilingual pretrained encoders. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 15–26, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *NeurIPS*, pages 7057–7067.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *ACL*, Florence, Italy. ACL.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.

Junliang Guo, Zhirui Zhang, Linli Xu, Hao-Ran Wei, Boxing Chen, and Enhong Chen. 2020. Incorporating BERT into parallel sequence decoding with adapters. In *NeurIPS*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society.

Tianyu He, Xu Tan, Yingce Xia, Di He, Tao Qin, Zhibo Chen, and Tie-Yan Liu. 2018. Layer-wise coordination between encoder and decoder for neural machine translation. In *NeurIPS*, pages 7955–7965.

Kenji Imamura and Eiichiro Sumita. 2019. Recycling a pre-trained BERT encoder for neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 23–31, Hong Kong. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguistics*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke

Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*, pages 2227–2237. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Rongxiang Weng, Heng Yu, Shujian Huang, Weihua Luo, and Jiajun Chen. 2019. Improving neural machine translation with pre-trained representation. *CoRR*, abs/1908.07688.

Rongxiang Weng, Heng Yu, Weihua Luo, and Min Zhang. 2022. Deep fusing pre-trained models into neural machine translation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11468–11476. AAAI Press.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP*, Online. ACL.

Haoran Xu, Benjamin Van Durme, and Kenton W. Murray. 2021. Bert, mbert, or bibert? A study on contextualized embeddings for neural machine translation. In *EMNLP*, pages 6663–6675. ACL.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Weinan Zhang, Yong Yu, and Lei Li. 2020. Towards making the most of bert in neural machine translation. In *AAAI*, volume 34, pages 9378–9385.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2020. Incorporating bert into neural machine translation. In *ICLR*.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7

☑ A2. Did you discuss any potential risks of your work?
7

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3.1, 3.2 and 3.3

☑ B1. Did you cite the creators of artifacts you used?
3.1, 3.2 and 3.3

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
3.1

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3.1 and 3.2

C ☑ Did you run computational experiments?

3 and 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Table 1

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
3.2 and 3.3. 4

☐ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Not applicable. 3.4 and 4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3.1, 3.2, 3.3 and 4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.