CSP: Code-Switching Pre-training for Neural Machine Translation

Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang and Qi Ju
Tencent Minority-Mandarin Translation
{zieenyang, bojiehu, ambyera, springhuang, damonju}@tencent.com

Abstract

This paper proposes a new pre-training method, called Code-Switching Pre-training (CSP for short) for Neural Machine Translation (NMT). Unlike traditional pre-training method which randomly masks some fragments of the input sentence, the proposed CSP randomly replaces some words in the source sentence with their translation words in the target language. Specifically, we firstly perform lexicon induction with unsupervised word embedding mapping between the source and target languages, and then randomly replace some words in the input sentence with their translation words according to the extracted translation lexicons. CSP adopts the encoder-decoder framework: its encoder takes the code-mixed sentence as input, and its decoder predicts the replaced fragment of the input sentence. In this way, CSP is able to pre-train the NMT model by explicitly making the most of the cross-lingual alignment information extracted from the source and target monolingual corpus. Additionally, we relieve the pretrain-finetune discrepancy caused by the artificial symbols like [mask]. To verify the effectiveness of the proposed method, we conduct extensive experiments on unsupervised and supervised NMT. Experimental results show that CSP achieves significant improvements over baselines without pre-training or with other pre-training methods.

1 Introduction

Neural machine translation (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2015) which typically follows the encoder-decoder framework, directly applies a single neural network to transform the source sentence into the target sentence. With tens of millions of trainable parameters in the NMT model, translation tasks are usually data-hungry, and many of them are low-resource or even zero-resource in terms of training data. Following the idea of unsupervised and self-supervised pre-training methods in the NLP area (Peters et al., 2018; Radford et al., 2018, 2019; Devlin et al., 2019; Yang et al., 2019), some works are proposed to improve the NMT model with pre-training, by making full use of the widely available monolingual corpora (Lample and Conneau, 2019; Song et al., 2019b; Edunov et al., 2019; Huang et al., 2019; Wang et al., 2019; Rothe et al., 2019; Clinchant et al., 2019). Typically, two different branches of pre-training approaches are proposed for NMT: model-fusion and parameter-initialization.

The model-fusion approaches seek to incorporate the sentence representation provided by the pre-trained model, such as BERT, into the NMT model (Yang et al., 2019b; Clinchant et al., 2019; Weng et al., 2019; Zhu et al., 2020; Lewis et al., 2019; Liu et al., 2020). These approaches are able to leverage the publicly available pre-trained checkpoints in the website but they need to change the NMT model to fuse the sentence embedding calculated by the pre-trained model. Large-scale parameters of the pre-trained model significantly increase the storage cost and inference time, which makes it hard for this branch of approaches to be directly used in production. As opposed to model-fusion approaches, the parameter-initialization approaches aim to directly pre-train the whole or part of the NMT model with tailored objectives, and then initialize the NMT model with pre-trained parameters (Lample and Conneau, 2019; Song et al., 2019b). These approaches are more production-ready since they keep the size and structure of the model same as standard NMT systems.

While achieving substantial improvements, these
pre-training approaches have two main cons. Firstly, as pointed out by Yang et al. (2019), the artificial symbols like [mask] used by these approaches during pre-training are absent from real data at fine-tuning time, resulting in a pretrain-finetune discrepancy. Secondly, while each pre-training step only involves sentences from the same language, these approaches are unable to make use of the cross-lingual alignment information contained in the source and target monolingual corpus. We argue that, as a cross-lingual sequence generation task, NMT requires a tailored pre-training objective which is capable of making use of cross-lingual alignment signals explicitly, e.g., word-pair information extracted from the source and target monolingual corpus, to improve the performance.

To address the limitations mentioned above, we propose Code-Switching Pre-training (CSP) for NMT. We extract the word-pair alignment information from the source and target monolingual corpus automatically, and then apply the extracted alignment information to enhance the pre-training performance. The detailed training process of CSP can be presented in two steps: 1) perform lexicon induction to get translation lexicons by unsupervised word embedding mapping (Artetxe et al., 2018a; Conneau et al., 2018); 2) randomly replace some words in the input sentence with their translation words in the extracted translation lexicons and train the NMT model to predict the replaced words. CSP adopts the encoder-decoder framework: its encoder takes the code-mixed sentence as input, and its decoder predicts the replaced fragments based on the context calculated by the encoder. By predicting the sentence fragment which is replaced on the encoder side, CSP is able to either attend to the remaining words in the source language or to the translation words of the replaced fragment in the target language. Therefore, CSP trains the NMT model to: 1) learn how to build the sentence representation for the input sentence as the traditional pre-training methods do; 2) learn how to perform cross-lingual translation with extracted word-pair alignment information. In summary, we mainly make the following contributions:

- We propose the code-switching pre-training for NMT, which makes full use of the cross-lingual alignment information contained in source and target monolingual corpus to improve the pre-training for NMT.
- We conduct extensive experiments on supervised and unsupervised translation tasks. Experimental results show that the proposed approach consistently achieves substantial improvements.
- Last but not least, we find that CSP can successfully handle the code-switching inputs.

2 Related works

Several approaches have been proposed to improve NMT with pre-training. Edunov et al. (2019) proposed to feed the last layer of ELMo to the encoder of NMT and investigated several different ways to add pre-trained language model representations to the NMT model. Weng et al. (2019) proposed a bi-directional self-attention language model to get sentence representation and introduced two individual methods, namely weighted-fusion mechanism and knowledge transfer paradigm, to enhance the encoder and decoder. Yang et al. (2019b) proposed a concerted training framework to make the most use of BERT in NMT. Zhu et al. (2020) proposed to fuse the representations from BERT with each layer of the encoder and decoder of the NMT model through attention mechanisms. Large-scale parameters of the pre-trained model in these approaches discussed above significantly increase the storage cost and inference time, which makes these approaches a little far from production. The other branch of approaches aims to keep the structure and size the same to the standard NMT system and designs some pre-training objectives tailored for NMT. Lample and Conneau (2019) proposed Cross-Lingual Language Model (XLM) objective and built a universal cross-lingual encoder. To improve the cross-lingual pre-training, they introduced supervised translation language modeling objective relying on the parallel data available. Song et al. (2019b) proposed the MASS objective to pre-train the whole NMT model instead of only pre-training the encoder by XLM. CSP builds on top of Lample and Conneau (2019) and Song et al. (2019b), and it explicitly makes full use of the alignment information extracted from the source and target monolingual corpus to enhance pre-training.

There have also been works on applying pre-specified translation lexicons to improve the performance of NMT. Hokamp and Liu (2017) and Post

1To be used in production easily, these models need to be distilled into a student model with the structure and size same as standard NMT systems.
and Vilar (2018) proposed an altered beam search algorithm, which took target-side pre-specified translations as lexical constraints during beam search. Song et al. (2019a) investigated a data augmentation method, making code-switched training data by replacing source phrases with their target translations according to the pre-specified translation lexicons. Recently, motivated by the success of unsupervised cross-lingual embeddings, Artetxe et al. (2018b), Lample et al. (2018a) and Yang et al. (2018) applied the pre-trained translation lexicons to initialize the word embeddings of the unsupervised NMT model. Sun et al. (2019) applied translation equivalences between word pairs from two languages with the same shared vocabulary created automatically extracted from the monolingual corpus. The motivation behind is two-fold: Firstly, with processing the source and target languages by the shared vocabulary, the encoder of the NMT model is able to share the same vocabulary with the decoder. Sharing the vocabulary between the encoder and decoder makes it possible for CSP to replace the source words in the input sentence with their translation words in the target language. Secondly, as pointed out by Lample and Conneau (2019), the shared vocabulary greatly improves the alignment of embedding spaces.

3 CSP

In this section, we firstly describe how to build the shared vocabulary for the NMT model; then we present the way extracting the probabilistic translation lexicons; and we introduce the detailed training process of CSP finally.

3.1 Shared sub-word vocabulary

This paper processes the source and target languages with the same shared vocabulary created through the sub-word toolkits, such as Sentence-Piece (SP) and Byte-Pair Encoding (BPE) (Sennrich et al., 2016b). We learn the sub-word splits on the concatenation of the sentences equally sampled from the source and target corpus. The motivation behind is two-fold: Firstly, with processing the source and target languages by the shared vocabulary, the encoder of the NMT model is able to share the same vocabulary with the decoder. Sharing the vocabulary between the encoder and decoder makes it possible for CSP to replace the source words in the input sentence with their translation words in the target language. Secondly, as pointed out by Lample and Conneau (2019), the shared vocabulary greatly improves the alignment of embedding spaces.

3.2 Probabilistic translation lexicons

Recently, some works successfully learned translation equivalences between word pairs from two monolingual corpus and extracted translation lexicons (Artetxe et al., 2018a; Conneau et al., 2018). Following Artetxe et al. (2018a), we utilize unsupervised word embedding mapping to extract probabilistic translation lexicons with monolingual corpus only. The probabilistic translation lexicons in this paper are defined as one-to-many source-target word translations. Specifically, given separate source and target word embeddings, i.e., \( X_c \) and \( Y_c \) trained on source and target monolingual corpus \( X \) and \( Y \), unsupervised word embedding mapping utilizes self-learning or adversarial-training to learn a mapping function \( f(X) = WX \), which transforms source and target word embeddings to a shared embedding space. With word embeddings in the same latent space, we measure the similarities between source and target words with the cosine distance of word embeddings. Then, we extract the probabilistic translation lexicons by selecting the top \( k \) nearest neighbors in the shared embedding space. Formally, considering the word \( x_u \) in the source language, its top \( k \) nearest neighbor words in the target language, denoted as \( y_{1u}, y_{2u}, \ldots, y_{ku} \), are extracted as its translation words, and the corresponding normalized similarities \( s_{u1}, s_{u2}, \ldots, s_{uk} \) are defined as the translation probabilities.

3.3 Training process of CSP

CSP only requires monolingual data to pre-train the NMT model. Given an unpaired source sentence \( x \in X \), where \( x = (x_1, x_2, \ldots, x_m) \) is the source sentence with \( m \) tokens, we denote \( x_{[u:v]} \) as the sentence fragment of \( x \) from \( u \) to \( v \) where \( 0 < u < v < m \), and denote \( x_{[u:v]} \) as modified version of \( x \) where its fragment from position \( u \) to \( v \) are replaced with their translation words according to the probabilistic translation lexicons. Formally, \( x_{[u:v]} \) is represented as:

\[
x_{[u:v]} = (x_1, \ldots, x_{u-1}, y'_u, \ldots, y'_v, x_{v+1} \ldots, x_m)
\]

where \( x_{[u:v]} = (y'_u, \ldots, y'_v) \) is sampled based on the extracted probabilistic translation lexicons presented on Section 3.2. Here, we take the replacing process from \( x_u \) to \( y'_u \) as an example. Considering the source word \( x_u \), its top \( k \) translation words \( y_{1u}, y_{2u}, \ldots, y_{ku} \) and the translation probabilities \( s_{u1}, s_{u2}, \ldots, s_{uk} \), \( y'_u \) is calculated as:

\[
y'_u = y_{uj} (1 \leq j \leq k)
\]

where \( y_{uj} \) is decided by performing multinomial sampling on the distribution defined by translation probabilities \( s_{u1}, s_{u2}, \ldots, s_{uk} \). With higher translation probability \( s_{uj} \), the translation word \( y_{uj} \) is more likely to be selected.
Translation Model

Encoder

\[ \begin{array}{cccccc}
    x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \\
    y_1' & y_2' & s_1' & s_2' & y_3' & y_4' \\
    \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
    y_k' & y_k' & s_k' & s_k' & y_k' & y_k' \\
\end{array} \]

Attention

Decoder

\[ x_3 x_4 x_5 x_6 \]

Sampling

Figure 1: The training example of our proposed CSP which randomly replaces some words in the source input with their translation words based on the probabilistic translation lexicons. Identical to MAS, the token ‘-‘ represents the padding in the decoder. The attention module represents the attention between the encoder and decoder.

Similar to Song et al. (2019b), CSP pre-trains a sequence to sequence model by predicting the sentence fragment \( x_{[w:v]} \) with the modified sequence \( x^{\backslash u:v} \) as input. With the log likelihood as the objective function, CSP trains the NMT model on the monolingual corpora \( X \) as:

\[
L(\theta; X) = \frac{1}{|X|} \sum_{x \in X} \log P(x_{[w:v]} | x^{\backslash u:v}; \theta) = \frac{1}{|X|} \sum_{x \in X} \log \prod_{t=u}^v P(x_t | x_{<t}, x^{\backslash u:v}; \theta)
\]

(3)

Figure 1 shows an example for CSP training, where the original source sentence \((x_1, x_2, x_3, x_4, x_5, x_6)\) with the fragment \((x_3, x_4, x_5, x_6)\) being replaced with their translation words, i.e., \((y'_3, y'_4, y'_5, y'_6)\) sampled from the extracted probabilistic translation lexicons. The encoder takes the code-mixed source sentence as input, and the decoder only predicts the replaced fragment \((x'_3, x'_4, x'_5, x'_6)\).

4 Experiments and Results

This section describes the experimental details about CSP pre-training and fine-tuning on the supervised and unsupervised NMT tasks. To test the effectiveness and generality of CSP, we conduct extensive experiments on English-German, English-French and Chinese-to-English translation tasks.

4.1 CSP pre-training

Model configuration We choose Transformer as the basic model structure. Following the base model in Vaswani et al. (2017), we set the dimension of word embedding as 512, dropout rate as 0.1 and the head number as 8. To be comparable with previous works, we set the model as 4-layer encoder and 4-layer decoder for unsupervised NMT, and 6-layer encoder and 6-layer decoder for supervised NMT. The encoder and decoder share the same word embeddings.

Datasets and pre-processing Following the work of Song et al. (2019b), we use the monolingual data sampled from WMT News Crawl datasets for English, German and French, with 50M sentences for each language. For Chinese, we choose 10M sentences from the combination of LDC and WMT2018 corpora. For each translation task, the source and target languages are jointly tokenized into sub-word units with BPE (Sennrich et al., 2016b). The vocabulary is extracted from the tokenized corpora and shared by the source and target languages. For English-German and English-French translation tasks, we set the vocabulary size as 32k. For Chinese-English, the vocabulary size is set as 60k since few tokens are shared by Chinese
<table>
<thead>
<tr>
<th>System</th>
<th>en-de</th>
<th>de-en</th>
<th>en-fr</th>
<th>fr-en</th>
<th>zh-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lample et al. (2018b)</td>
<td>17.16</td>
<td>21.0</td>
<td>25.14</td>
<td>24.18</td>
<td>-</td>
</tr>
<tr>
<td>Lample and Conneau (2019)</td>
<td>27.0</td>
<td>34.3</td>
<td>33.4</td>
<td>33.3</td>
<td>-</td>
</tr>
<tr>
<td>Song et al. (2019b)</td>
<td>28.1</td>
<td>35.0</td>
<td>37.5</td>
<td>34.6</td>
<td>-</td>
</tr>
<tr>
<td>Lample and Conneau (2019) (our reproduction)</td>
<td>27.3</td>
<td>33.8</td>
<td>32.9</td>
<td>33.5</td>
<td>22.1</td>
</tr>
<tr>
<td>Song et al. (2019b) (our reproduction)</td>
<td>27.9</td>
<td>34.7</td>
<td>37.3</td>
<td>34.1</td>
<td>22.8</td>
</tr>
<tr>
<td>CSP and fine-tuning (ours)</td>
<td>28.7</td>
<td>35.7</td>
<td>37.9</td>
<td>34.5</td>
<td>23.9</td>
</tr>
</tbody>
</table>

Table 1: The translation performance of the fine-tuned unsupervised NMT models. To reproduce the results of Lample and Conneau (2019) and Song et al. (2019b), we directly run their released codes on the website.3

and English. To extract the probabilistic translation lexicons, we utilize the monolingual corpora described above to train the embeddings for each language independently by using word2vec (Mikolov et al., 2013). We then apply the public implementation of the method proposed by Artetxe et al. (2017) to map the source and target word embeddings to a shared-latent space.4

Training details We replace the consecutive tokens in the source input with their translation words sampled from the probabilistic translation lexicons, with random start position $u$. Following Song et al. (2019b), the length of the replaced fragment is empirically set as roughly 50% of the total number of tokens in the sentence, and the replaced tokens in the encoder will be the translation tokens 80% of the time, a random token 10% of the time and an unchanged token 10% of the time.5 In the extracted probabilistic translation lexicons, we only keep top three translation words for each source word and also investigate how the number of translation words produces an effect on the training process. All of the models are implemented on Py-Torch and trained on 8 P40 GPU cards.6 We use Adam optimizer with a learning rate of 0.0005 for pre-training.

4.2 Fine-tuning on unsupervised NMT

In this section, we describe the experiments on the unsupervised NMT, where we only utilize monolingual data to fine-tune the NMT model based on the pre-trained model.

Experimental settings For the unsupervised English-German and English-French translation tasks, we take the similar experimental settings to Lample and Conneau (2019); Song et al. (2019b). Specifically, we randomly sample 5M monolingual sentences from the monolingual data used during pre-training and report BLEU scores on WMT14 English-French and WMT16 English-German. For fine-tuning on the unsupervised Chinese-to-English translation task, we also randomly sample 1.6M monolingual sentences for Chinese and English respectively similar to Yang et al. (2018). We take NIST02 as the development set and report the BLEU score averaged on the test sets NIST03, NIST04 and NIST05. To be consistent with the baseline systems, we apply the script multi-bleu.pl to evaluate the translation performance for all of the translation tasks.

Baseline systems We take the following four strong baseline systems. Lample et al. (2018b) achieved state-of-the-art (SOTA) translation performance on unsupervised English-German and English-French translation tasks, by utilizing cross-lingual vocabulary, denoising auto-encoding and back-translation. Yang et al. (2018) proposed the weight-sharing architecture for unsupervised NMT and achieved SOTA results on unsupervised Chinese-to-English translation task. Lample and Conneau (2019) and Song et al. (2019b) are among the first endeavors to apply pre-training to unsupervised NMT, and both of them achieved substantial improvements compared to the methods without utilizing pre-training.

Results Table 1 shows the experimental results on the unsupervised NMT. From Table 1, we can find that the proposed CSP outperforms all of the previous works on English-to-German, German-to-
English, English-to-French and Chinese-to-English unsupervised translation tasks, with as high as +0.7 BLEU points improvement in German-to-English translation task. In French-to-English translation direction, CSP also achieves comparable results with the SOTA baseline of Song et al. (2019b). In Chinese-to-English translation task, CSP even achieves +1.1 BLEU points improvement compared to the reproduced result of Song et al. (2019b). These results indicate that fine-tuning unsupervised NMT on the model pre-trained by CSP consistently outperforms the previous unsupervised NMT systems with or without pre-training.

4.3 Fine-tuning on supervised NMT

This section describes our experiments on supervised NMT where we fine-tune the pre-trained model with bilingual data.

Experimental settings For supervised NMT, we conduct experiments on the publicly available data sets, i.e., WMT14 English-French, WMT14 English-German and LDC Chinese-to-English corpora, which are used extensively as benchmarks for NMT systems. We use the full WMT14 English-German and WMT14 English-French corpus as our training sets, which contain 4.5M and 36M sentence pairs respectively. For Chinese-to-English translation task, our training data consists of 1.6M sentence pairs randomly extracted from LDC corpora. All of the sentences are encoded with the same BPE codes utilized in pre-training.

Baseline systems For supervised NMT, we consider the following three baseline systems. The first one is the work of Vaswani et al. (2017), which achieves SOTA results on WMT14 English-German and English-French translation tasks. The other two baseline systems are proposed by Lample and Conneau (2019) and Song et al. (2019b), both of which fine-tune the supervised NMT tasks on the pre-trained models. Furthermore, we compare with the back-translation method which has shown its great effectiveness on improving NMT model with monolingual data (Sennrich et al., 2016a). Specifically, for each baseline system, we translate the target monolingual data used during pre-training back to the source language by a reversely-trained model, and get the pseudo-parallel corpus by combining the translation with its original data. At last, the training data which includes pseudo and parallel sentence pairs is shuffled and used to train the NMT system.

Results The experimental results on supervised NMT are presented at Table 2. We report the BLEU scores on English-to-German, English-to-French and Chinese-to-English translation directions. For each translation task, we report the BLEU scores for the standard NMT model and the model trained with back-translation respectively. As shown in Table 2, compared to the baseline system without pre-training (Vaswani et al., 2017), the proposed model achieves +1.6 and +0.7 BLEU points improvements on English-to-German and English-to-French translation directions respectively. Even compared to stronger baseline system with pre-training (Song et al., 2019b), we also achieve +0.5 and +0.4 BLEU points improvements respectively on these two translation directions. On Chinese-to-English translation task, the proposed model achieves +0.7 BLEU points improvement compared to the baseline system of Song et al. (2019b). With back-translation, the proposed model still outperforms all of the baseline systems. Experimental results above show that fine-tuning the supervised

<table>
<thead>
<tr>
<th>System</th>
<th>en-de</th>
<th>en-fr</th>
<th>zh-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vaswani et al. (2017) (our reproduction) / + BT</td>
<td>27.0 / 28.6</td>
<td>37.9 / 39.3</td>
<td>42.1 / 43.7</td>
</tr>
<tr>
<td>Lample and Conneau (2019) (our reproduction) / + BT</td>
<td>28.1 / 29.4</td>
<td>38.3 / 39.6</td>
<td>42.0 / 43.7</td>
</tr>
<tr>
<td>Song et al. (2019b) (our reproduction) / + BT</td>
<td>28.4 / 29.6</td>
<td>38.4 / 39.6</td>
<td>42.5 / 44.1</td>
</tr>
<tr>
<td>CSP and fine-tuning (ours) / + BT</td>
<td>28.9 / 30.0</td>
<td>38.8 / 39.9</td>
<td>43.2 / 44.6</td>
</tr>
</tbody>
</table>

Table 2: The translation performance of supervised NMT on English-German, English-French and Chinese-to-English test sets. (+ BT: trains the model with back-translation method.)
NMT on the model pre-trained by CSP achieves substantial improvements over previous supervised NMT systems with or without pre-training. Additionally, it has been verified that CSP is able to work together with back-translation.

5 Analysis

5.1 Study the number of translation words

In CSP, the probabilistic translation lexicons only keep the top $k$ translation words for each source word. For each word in the translation lexicons, the number of translation words $k$ is viewed as an important hyper-parameter and can be set carefully during the process of pre-training. A natural question is that how much of translation words do we need to keep for each source word? Intuitively, if $k$ is set as a small number, the model may lose its generality since each source word can be replaced with only a few translation words, which severely limits the diversity of the context. And if otherwise, the accuracy of the extracted probabilistic translation lexicons may get significantly diminished, which shall introduce too much noise for pre-training. Therefore, there is a trade-off between the generality and accuracy. We investigate this problem by studying the translation performance of unsupervised NMT with different $k$, where we vary $k$ from 1 to 10 with the interval

2. We observe both the performance of CSP after pre-training and the translation performance after fine-tuning on the unsupervised NMT tasks, including the English-to-German and English-to-French translation directions. For each translation direction, we firstly present the perplexity (PPL) score of the pre-trained model averaged on the monolingual validation sets of the source and target languages. And then we show the BLEU score of the fine-tuned model on the bilingual validation set. Figure 2 (a) and (c) illustrate the PPL score of the pre-trained model and BLEU score of the fine-tuned unsupervised NMT model respectively on English-to-German translation. Figure 2 (b) and (d) present the PPL and BLEU score respectively for English-to-French translation. From Figure 2, it can be seen that, when $k$ is set around 3, the pre-trained model achieves the best validation PPL scores on both of the English-to-German and English-to-French translation directions. Similarly, CSP also achieves the best BLEU scores on the unsupervised translation tasks when $k$ is set around 3.

5.2 Ablation study

To understand the importance of different components of the model pre-trained by CSP, we perform an ablation study by training multiple versions of

9 We randomly select the target monolingual data with the same size to the bilingual data.

10 For English-German translation, the monolingual validation set for English is built by including all English sentences in the bilingual English-German validation set, and the monolingual validation set for German is built in the same way.
the supervised NMT model with some components initialized randomly: the word embeddings, the encoder, the attention module between the encoder and decoder, and the decoder. Experiments are conducted on English-to-German and English-to-French translation tasks. All models are trained without back-translation and results are reported in Table 3. We can find that the two most critical components are the pre-trained encoder and attention module. It shows that CSP enhances NMT not only on the ability of building sentence representation for the input sentence, but also on the ability of aligning the source and target languages with the help of word-pair alignment information. Additionally, the experimental results indicate that the pre-trained decoder shows little effect on the translation performance. This is mainly because the decoder only predicts the source-side words during pre-training but predicts the target-side words during fine-tuning. This pretrain-finetune mismatch makes the pre-trained decoder less helpful for performance improvement.

Table 3: Ablation study on English-German and English-French translation tasks. The embeddings include the source-side and target-side word embeddings.

<table>
<thead>
<tr>
<th>System</th>
<th>en-de</th>
<th>en-fr</th>
</tr>
</thead>
<tbody>
<tr>
<td>No pre-trained embeddings</td>
<td>28.4</td>
<td>38.5</td>
</tr>
<tr>
<td>No pre-trained encoder</td>
<td>27.9</td>
<td>38.2</td>
</tr>
<tr>
<td>No pre-trained attention module</td>
<td>28.1</td>
<td>38.3</td>
</tr>
<tr>
<td>No pre-trained decoder</td>
<td>28.8</td>
<td>38.8</td>
</tr>
<tr>
<td>Full model pre-trained by CSP</td>
<td>28.9</td>
<td>38.8</td>
</tr>
</tbody>
</table>

5.3 Code-switching translation

Code-switching, which contains words from different languages in single input, has aroused more and more attention in NMT (Johnson et al., 2017; Menacer et al., 2019). In this section, we show that the proposed CSP is able to enhance the ability of the fine-tuned NMT model on handling the code-switching input. To present quantitative results, we build two test sets for the supervised Chinese-to-English translation task to evaluate the performance of the translation model on handling code-switching inputs. We randomly select 200 Chinese-English sentence pairs from NIST02, based on which we build two code-switching test sets. The first test set, referred to as test A, is built by randomly replacing some phrases in each Chinese sentence with their counterpart English phrases, where the English phrase is the translation result by feeding the corresponding Chinese phrase to the Google Chinese-to-English translator; The second test set, referred to as test B, is constructed by randomly replacing parts of the words in each Chinese sentence with their nearest target words in the shared latent embedding space (the same way used by CSP in Section 3.2). Table 4 shows the translation performance of NMT systems on the two code-switching test sets. Besides the baseline systems mentioned in section 4.3, we also train a Chinese-English multi-lingual system (Johnson et al., 2017) based on Transformer, which has shown the ability of handling code-switching inputs. From Table 4, We can find that the proposed approach achieves significant improvements over previous works. Compared to multi-lingual system, we achieve +2.3 and +3.0 BLEU points improvements respectively on test A and test B. The case study can be found in appendix D.

Table 4: The performance of Chinese-to-English translation on in-house code-switching test sets.

<table>
<thead>
<tr>
<th>System</th>
<th>test A</th>
<th>test B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vaswani et al. (2017)</td>
<td>28.17</td>
<td>32.51</td>
</tr>
<tr>
<td>Lample and Conneau (2019)</td>
<td>28.82</td>
<td>32.90</td>
</tr>
<tr>
<td>Song et al. (2019b)</td>
<td>28.70</td>
<td>33.21</td>
</tr>
<tr>
<td>Multi-lingual system</td>
<td>30.51</td>
<td>35.10</td>
</tr>
<tr>
<td>CSP and fine-tuning</td>
<td>32.84</td>
<td>38.17</td>
</tr>
</tbody>
</table>

6 Conclusions and Future work

This work proposes a simple yet effective pre-training approach, i.e., CSP for NMT, which randomly replaces some words in the source sentence with their translation words in the probabilistic translation lexicons extracted from monolingual corpus only. To verify the effectiveness of CSP, we investigate two downstream tasks, supervised and unsupervised NMT, on English-German, English-French and Chinese-to-English translation tasks. Experimental results show that the proposed approach achieves substantial improvements over strong baselines consistently. Additionally, we show that CSP is able to enhance the ability of NMT on handling code-switching inputs. There are two promising directions for the future work. Firstly, we are interested in applying CSP to other...

The two in-house code-switching test sets can be found in the attached files.
related NLP areas for code-switching problems. Secondly, we plan to investigate the pre-training objectives which are more effective in utilizing the cross-lingual alignment information for NMT.

Acknowledgement
We sincerely thank the anonymous reviewers for their thorough reviewing and valuable suggestions.

References


A The configurations for the open-source toolkit

A.1 Get word embeddings with word2vec

We train the word embedding using the following script:

```
word2vec -train text -output embedding.txt -cbow 0 -size 512 -window 10 -negative 10 -hs 0 -sample 1e-5 -threads 50 -binary 0 -min-count 5 -iter 10
```

A.2 Word embedding mapping with Vecmap

After we get the embeddings for both the source and target languages, namely s_embedding.txt and t_embedding.txt, we use the open-source Vecmap to map these embeddings to a shared-latent space with the following scripts:

```
python normalize_embeddings.py unit center -i s_embedding.txt -o s_embedding.normalized.txt
```

```
python map_embeddings.py -orthogonal s_embedding.normalized.txt t_embedding.normalized.txt
```

B Study of different length of the replaced segment

The length of the replaced fragment is an hyperparameter which can be set by the user beforehand. We are curious to know how the length of the replaced fragment shows effect on CSP. In this section, we study the performance of CSP with different length of the replaced fragment, where we set the length of the replaced fragment from 10% to 90% percentage of the sentence length with a step size of 10%. Similar to section 5.1, we report both the performance of CSP after pre-training and the translation performance after fine-tuning on the unsupervised NMT tasks, including the English-to-German and English-to-French translation directions. For each translation direction, we firstly present the perplexity (PPL) score of the pre-trained model averaged on the monolingual validation sets of the source and target languages. And then we show the BLEU score of the fine-tuned model on the bilingual validation set. Figure 3 (a) and (c) illustrate the PPL score of the pre-trained model on the bilingual validation set. Figure 3 (b) and (d) present the PPL and BLEU score respectively for English-to-German translation direction. Figure 3 (b) and (d) present the PPL and BLEU score respectively for English-to-French translation direction. We can find that when the length of the replaced fragment is set nearly 50% of the sentence length, CSP achieves best performance not only on the pre-training task but also on the downstream unsupervised NMT task. Therefore, we set the length of the replaced fragment as 50% of the sentence length in our experiments.

```
https://github.com/artetxem/vecmap
```

12https://github.com/artetxem/vecmap
Figure 3: The performance of CSP with different length of the replaced fragment, which includes: (a) the PPL score of the pre-trained English-to-German model; (b) the PPL score of the pre-trained English-to-French model; (c) the BLEU score of the fine-tuned unsupervised English-to-German NMT model; (d) the BLEU score of the fine-tuned unsupervised English-to-French NMT model.

Table 5: The comparison between CSP and model-fusion approaches. We get the translation result of Zhu et al. (2020) by directly running their released codes on the website.\footnote{https://github.com/bert-nmt/bert-nmt}

<table>
<thead>
<tr>
<th>System</th>
<th>en-de</th>
<th>en-fr</th>
<th>zh-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhu et al. (2020)</td>
<td>29.2</td>
<td>39.1</td>
<td>43.8</td>
</tr>
<tr>
<td>+Knowledge Distillation</td>
<td>28.7</td>
<td>38.5</td>
<td>43.4</td>
</tr>
<tr>
<td>CSP and fine-tuning (ours)</td>
<td><strong>28.9</strong></td>
<td><strong>38.8</strong></td>
<td><strong>43.2</strong></td>
</tr>
</tbody>
</table>

C Compared to model-fusion approaches

In this section, we compare the proposed CSP with model-fusion approaches. We conduct experiments on supervised NMT where we fine-tune the pre-trained model with bilingual data. Experimental settings are identical to the settings in section 4.3. We report the performance of English-to-German, English-to-French and Chinese-to-English translation tasks. Since Zhu et al. (2020) released their code which makes their results reproducible, we take their system as the baseline. To make the comparison more fair, we distill the model of Zhu et al. (2020) to a student model which has the same size and structure to standard NMT model. For knowledge distillation, we utilized the sequence-level knowledge distillation proposed by Kim and Rush (2016).\footnote{While variant distillation methods have been proposed recently, we only test the most simple and standard one.} Experimental results are presented in Table 5. We can find that, compared to the distilled student model of Zhu et al. (2020), CSP achieves better translation performance on two of three translation tasks.

D Case study for code-switching

In this section, we compare the performance of different NMT systems by case study. We randomly select some examples of the code-switching inputs and get the outputs by feeding the code-switching inputs into different NMT systems. The results are presented in Table 6. We can find that, for the two code-switching input sentences in Table 6, the standard Transformer and the multi-lingual system are both easily to give insufficient translations with some semantic contents untranslated. We assume that this is mainly because these systems are weak in encoding the full context of the code-switching input. Compared to the baseline systems, our system gives more sufficient and fluent translations. This shows that CSP enhances the model’s ability...
<table>
<thead>
<tr>
<th>Source sentence</th>
<th>切尼在访问所有海湾国家以后星期一到科威特。But even this most loyal US ally is opposed to attacking Baghdad.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output of Transformer</td>
<td>cheney arrived kuwait after visiting all Gulf, But even this most employed US</td>
</tr>
<tr>
<td>Output of Multi-lingual system</td>
<td>Cheney arrived in Kuwait after visiting, but even this most loyal is opposed to attacking Baghdad.</td>
</tr>
<tr>
<td>Output of our system</td>
<td>Cheney arrived in Kuwait on Monday after visiting all Gulf countries, but even the most loyal US ally is opposed to attacking Baghdad.</td>
</tr>
<tr>
<td>Reference</td>
<td>cheney arrived kuwait on monday after visiting all other gulf states, however, even this most loyal ally to u.s. opposes an attack on baghdad.</td>
</tr>
<tr>
<td>Source sentence</td>
<td>对于日本《朝日新闻》报道说 Megawati will send a personal letter from Kim Dae Jung to Kim Jong Il, 韩国政府方面则予以否认。</td>
</tr>
<tr>
<td>Output of Transformer</td>
<td>as japan says, Megawati send a personal letter to Kim Jong, the south korea denied.</td>
</tr>
<tr>
<td>Output of Multi-lingual system</td>
<td>as for the news released in japan asahi that megawati will hand a letter from kim dae jung in his own handwriting to kim, the south korea denied this.</td>
</tr>
<tr>
<td>Output of our system</td>
<td>as for the news released in the japanese newspaper asahi that will hand a personal letter from kim dae jung in his own handwriting to kim jong, the south korean government denied.</td>
</tr>
<tr>
<td>Reference</td>
<td>as for the news released in the japanese newspaper asahi that megawati will hand a personal letter from kim dae jung in his own handwriting to kim jong - il, the south korean government denied this.</td>
</tr>
</tbody>
</table>

Table 6: Examples of the code-switching inputs and outputs of different NMT systems.

on encoding code-switching inputs.