Alternating Language Modeling for Cross-Lingual Pre-Training

Jian Yang,[1]* **Shuming Ma,**[2] **Dongdong Zhang,**[2] **ShuangZhi Wu,**[3]* **Zhoujun Li,**[1]† **Ming Zhou**[2]
[1]State Key Lab of Software Development Environment, Beihang University
[2]Microsoft Research Asia
[3]SPPD of Tencent Inc.
{jiaya, lizj}@buaa.edu.cn; {shumma, dozhang, mingzhou}@microsoft.com; frostwu@tencent.com

Abstract

Language model pre-training has achieved success in many natural language processing tasks. Existing methods for cross-lingual pre-training adopt Translation Language Model to predict masked words with the concatenation of the source sentence and its target equivalent. In this work, we introduce a novel cross-lingual pre-training method, called Alternating Language Modeling (ALM). It code-switches sentences of different languages rather than simple concatenation, hoping to capture the rich cross-lingual context of words and phrases. More specifically, we randomly substitute source phrases with target translations to create code-switched sentences. Then, we use these code-switched data to train ALM model to learn to predict words of different languages. We evaluate our pre-training ALM on the downstream tasks of machine translation and cross-lingual classification. Experiments show that ALM can outperform the previous pre-training methods on three benchmarks.[1]

Introduction

Recently language model pre-training methods, including ELMo (Peters et al. 2018), GPT (Radford et al. 2018), GPT2 (Radford et al. 2019), BERT (Devlin et al. 2019), and UniLM (Dong et al. 2019), have achieved impressive results on various natural language processing tasks such as question-answering (Min, Seo, and Hajishirzi 2017; Yang et al. 2019a), machine reading comprehension (Salant and Berant 2018; Yu et al. 2018) and natural language inference (Tay, Luu, and Hui 2018). More recently, XLM (Lample and Conneau 2019) has extended this approach to cross-lingual pre-training, and proven successful in applying language model pre-training in the cross-lingual setting.

Existing methods for supervised cross-lingual pre-training adopt a cross-lingual language model objective, called Translation Language Model (TLM). It makes use of parallel data by predicting the masked words with concatenation of the sentence and its translation. In this way, the cross-lingual pre-training model can learn the relationship between languages.

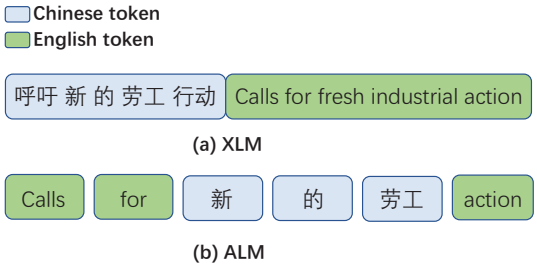


Figure 1: Example of Translation Language Model and Alternating Language Model.

In this work, we propose a novel cross-lingual language model, which alternately predicts words of different languages. Figure 1 shows an example of the proposed Alternating Language Model (ALM). Different from XLM, the input sequence of ALM is mixed with different languages, so it can capture the rich cross-lingual context of words and phrases. Moreover, it forces the language model to predict one language conditioned on the context of the other language. Therefore, it can minor the gap between the embeddings of the source language and the target languages, which is beneficial for the cross-lingual setting.

Based on Alternating Language Model, we introduce a new cross-lingual pre-training method. More specifically, we take the Transformer model (Vaswani et al. 2017) as the backbone model. Then, we construct the training examples for pre-training by replacing the phrases with their translation of the other language. Finally, we pre-train the Transformer model with the constructed examples using the masked language model objective. The pre-trained model can be used to further fine-tune the downstream cross-lingual tasks.

To verify the effectiveness of the proposed method, we evaluate our pre-training method on machine translation and cross-lingual classification. Experiments show that ALM can outperform the previous pre-training methods on three benchmark datasets.

The contributions of this work are as follows:

*Contribution during internship at Microsoft Research Asia.
†Corresponding author.

[1]Code can be found at https://github.com/zddfunseeker/ALM.

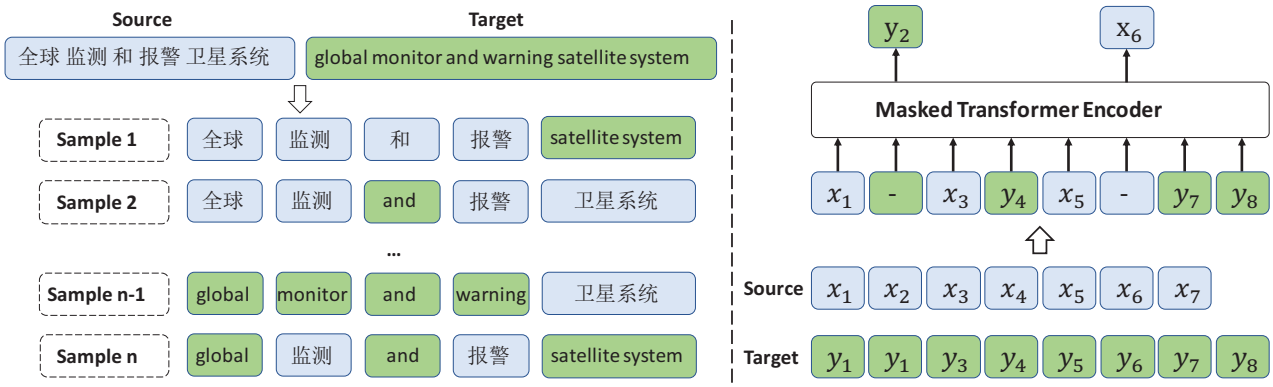


Figure 2: Overview of our ALM cross-lingual pre-training method. Given a pair of bilingual sentences, we yield a set of cross-lingual sentences. These sentences are used to pre-train the Transformer encoder which predicts an English masked word or a Chinese one.

- We propose a novel cross-lingual language model, which alternately predicts words of different languages.
- We introduce a new cross-lingual pre-training method based on the proposed cross-lingual language model, which can be further fine-tuned on downstream tasks.
- Experiments show that ALM outperforms the previous pre-training methods on the benchmark datasets for machine translation and cross-lingual text classification.

Cross-Lingual Pre-Training

Cross-lingual pre-training trains a model that can be further fine-tuned to improve downstream tasks by making use of monolingual data and bilingual data. XLM is a recently proposed model that achieves success in cross-lingual pre-training. It consists of two unsupervised models that relies on monolingual data, and a supervised model that relies on bilingual data. These three models of XLM are Causal Language Model (CLM), Masked Language Model (MLM), and Translation Language Model (TLM), respectively.

Unsupervised Language Modeling

CLM recurrently predicts the next word given the previous context, which is the typical objective of language modeling. GPT (Radford et al. 2018) is the first pre-training model to adopt CLM, and GPT-2 (Radford et al. 2019) further proves the success of CLM for pre-training.

CLM only makes use of the uni-directional context. Different from CLM, MLM uses bidirectional contextual information. It randomly masks some tokens during training and predicts the identity of the masked word. BERT (Devlin et al. 2019) is the first to propose this model and use it for pre-training. Different from the BERT, XLM (Lample and Conneau 2019) uses an arbitrary number of sentences (truncated at 256 tokens) instead of pairs of sentences, and it samples the masked tokens according to a multinomial distribution, whose weights are proportional to the square root of their invert frequencies.

Supervised Language Modeling

XLM also proposes an additional objective that can make use of bilingual data called TLM. TLM concatenates parallel sentences as training samples. Similar to MLM, it randomly masks words of concatenated sentences, so that it can leverage both words in source language and target language translation by predicting the masked words. Moreover, TLM leverages target sentences to predict source words when the source context is insufficient to predict these words.

TLM makes use of bilingual data by concatenating sentences of two languages, so it can learn the relationship between languages. In this work, we mainly focus on improving the supervised pre-training model. We also show that the proposed model can be applied to unsupervised settings in the following section.

Alternating Language Model

We propose Alternating Language Model (ALM) to alternately predict words of different languages. In this section, we present the details of ALM.

Code-Switched Sequence

Given a bilingual sentence pair (X, Y) with the source sentence $X = \{x_1, x_2, ..., x_N\}$ and the target translation $Y = \{y_1, y_2, ..., y_M\}$, where N and M are the lengths of the source and target sentences, we create the code-switched sequence U by composing the phrases of X and Y, where U=$\{u_1, u_2, .., u_L\}$ with the length L.

In details, for each phrase $U_{[i,j]}$, it comes from either source phrase $X_{[a,b]}$ or target phrase $Y_{[c,d]}$ where the constraint is that these two phrases are the linguistic translation counterpart in the parallel sentence (X, Y), $1 \leq a \leq b \leq N$ and $1 \leq c \leq d \leq M$. We denote the proportion of the source words in the alternating language sequence U as α.

Specifically, the constituent of U can be illustrated into four categories:

- Monolingual source language: that is $\alpha = 0$.
- Monolingual target language: that is $\alpha = 1$.

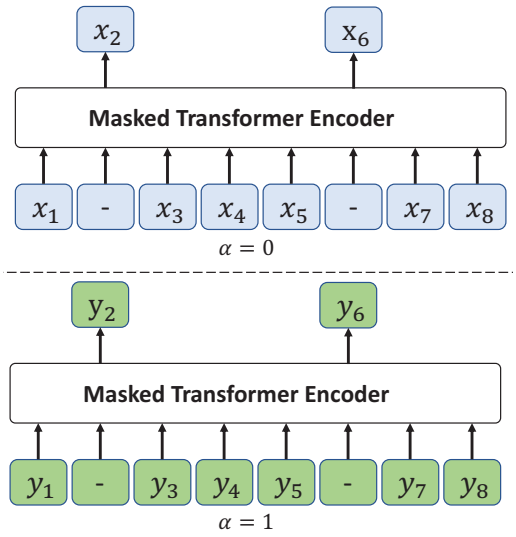


Figure 3: The model architecture of ALM when $\alpha = 0$ and $\alpha = 1$.

- Major source language: that means most of U is derived from X where some source phrases $X_{[a,b]}$ are substituted by their target counterpart phrases $Y_{[c,d]}$ ($\alpha \geq 0.5$).
- Major target language: that means most of U is derived from Y where some target phrases $Y_{[c,d]}$ are substituted by their source counterpart phrases $X_{[a,b]}$ ($\alpha < 0.5$).

Constructing Training Samples

Since there are few natural code-switched sentences, we should construct them from bilingual sentence pairs. First, we perform word alignment with the GIZA toolkit (Och and Ney 2003) between the parallel sentence X and Y, and extract a bilingual phrase table using statistical machine translation techniques (Koehn, Och, and Marcu 2003). Then, for each sentence pair in training corpus, we create the major-source-language samples by substituting some phrases in source sentence with the corresponding target phrases with highest probabilities in phrase table. A similar method creates major-target-language samples by substituting some phrases in target sentence with the corresponding source phrases.

The details of the construction for a sentence pair are:

- Each phrase is limited to less than 5 words for both source language and target language.
- The substituted words are less than 30% of the total words in the sentence. Therefore, the source words dominate the sentence in the major source language, while the target words dominate the sentence in the major target language.
- Each bilingual sentence pair is used to create multiple alternating language sentences by randomly choosing the substituted phrases.

Figure 2 shows an example of constructing code-switched sentences. Given the Chinese sentence and its translation, multiple training samples can be derived from one sentence pair by choosing different phrases to substitute.

Model Architecture and Pre-Training

Figure 2 also shows the overall architecture of our proposed model. Given a parallel sentence pair, we combine two sentences from different languages into a single code-switched sequence as described above. Then we mask out a certain percentage of words in the sequences. We feed the masked sentences into Transformer model to learn to predict the words being masked out.

In details, we sample randomly 15% of the tokens, replace them by a [MASK] token 80% of the time, by a random token 10% of the time, and keep them unchanged 10% of the time.

Figure 3 shows two special cases of ALM. When $\alpha = 0$, the input sequence is purely from source language. It becomes the masked language model for source language. When $\alpha = 1$, the input sequence is purely from target language, so it becomes the masked language model for target language. In this way, the model becomes unsupervised because it only relies on monolingual data.

In practice, we have 10% of training samples with $\alpha = 0$, 10% of samples with $\alpha = 1$, and the rest with $0 < \alpha < 1$. We manually choose a proper value of α which ensures some phrases are replaced with their counterparts by alignment instead of sweeping all values of α ($0 \leq \alpha \leq 1$). In order to ensure the value of α is in a reasonable range, we set max length and max number for phrase substitution.

Applying to Downstream Tasks

After pre-training, we further fine-tune ALM in order to adapt the parameters for the downstream tasks, which are machine translation and cross-lingual classification.

Machine Translation After pre-training, we use ALM as the encoder of machine translation, and construct a Transformer-based decoder conditioned on ALM. We fine-tune the parameters of the total encoder-decoder model on parallel training dataset of machine translation.

Cross-Lingual Classification XNLI (Conneau et al. 2018) is a significant dataset which is similar to the English MultiNLI including several languages. Taking the task of NLI as an example, we concatenate premise and hypothesis as input, and feed them into ALM. On top of ALM, we add a linear classifier and a dropout layer after the first hidden state for last layer. Then, we fine-tune the parameters of ALM on training dataset of cross-lingual classification.

Experiments

We evaluate our proposed method on machine translation and cross-lingual text classification. In this section, we provide the details, results, and analysis of the experiments.

Datasets

Following previous work (Lample and Conneau 2019), we use Wikipedia data by using WikiExtractor and WMT data as monolingual data. For bilingual data, French, Spanish,

Russian, Arabic, and Chinese data are from MultiUN (Ziemski, Junczys-Dowmunt, and Pouliquen 2016). Hindi data is from the IIT Bombay corpus (Kunchukuttan, Mehta, and Bhattacharyya 2018). German and Greek are from the EU-bookshop corpus. Turkish, Vietnamese and Thai are from OpenSubtitles 2018. Urdu and Swahili data are from Tanzil. Swahili data is from GlobalVoices. For most languages, we use the tokenizer provided by Moses (Koehn et al. 2007).

Pre-Training Details

We use byte pair encoding (BPE) (Sennrich, Haddow, and Birch 2016). The vocabulary contains 95K byte pair encoding tokens. We pre-train our model with both 1024 embedding and hidden units, 8 heads, a dropout rate of 0.1 and learned positional embeddings. We use an Adam optimizer with parameters of $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We set the inverse sqrt learning rate schedule with a linear warmup where the number of warmup step is 4000 and a learning rate of 0.0005.

For pre-training data, we use source language monolingual data ($\alpha = 0$) and target language monolingual data ($\alpha = 1$). Besides, we also split parallel data to expand monolingual data. For the monolingual data, we regard source language mono-lingual data as $\alpha = 0$ and target language mono-lingual data as $\alpha = 1$, which could be classified into a special situation of ALM. To construct monolingual dataset, we use Wikipedia data as monolingual data by using WikiExtractor. Our pre-training samples includes monolingual data and parallel data, we use original parallel data to generate 20 times code-switched sentences than original parallel data. More specifically, we separately obtain the alternating language sentences of source language and target language, which are 40 times than original parallel data in total. Considering that there exist some bad cases in alternating language sentences, we filter some low-quality code-switched sentences of which length is too long or too short, and randomly drop some sentences. At last, nearly 1.5 billion code-switched sentences are used for pre-training.

Fine-Tuning on Machine Translation

We fine-tune the pre-trained ALM on two datasets: WMT14 English-German machine translation and IWSLT14 German-English machine translation. WMT14 English-German machine translation dataset has 4.5 million sentence pairs for training. newsdev2014 is used as the validation set, while the newstest2014 is the testing set. IWSLT14 German-English machine translation dataset contains 160 thousand sentence pairs. They are collected from TED talks. We use iwslt14 devset as the validation set and the iwslt14 testset as the testing set.

We build a Transformer decoder conditioned on ALM encoder. We feed the source language into ALM, and generate the target language with decoder. We reload the parameters of word embedding and encoder parameters which are also used to initialize the decoder for our in-house NMT code from pre-trained model. We evaluate the performance of the translated sentences. The evaluation metric is BLEU (Papineni et al. 2002).

Baselines We compare our methods with state-of-the-art supervised methods and the pre-training methods, which are described as follows:

- **Transformer** (Vaswani et al. 2017): We implement Transformer model with our in-house tensorflow code, and the experimental settings are the same as Transformer (Vaswani et al. 2017)
- **ConvS2S** (Gehring et al. 2017): We report the results referring to the paper of convolutional sequence to sequence model(ConvS2S).
- **Weighted Transformer** (Ahmed, Keskar, and Socher 2017): It uses self-attention branches in place of multi-head attention. The branches replace multiple heads in attention mechanism of the original Transformer network.
- **Layer-wise Transformer** (He et al. 2018): It explicitly coordinates the learning of hidden representations of the encoder and decoder, gradually from low level to high level.
- **RNMT+** (Chen et al. 2018): It combines the advantages of both the recurrent structure and Transformer architecture.
- **LightConv and DynamicConv** (Wu et al. 2019): LightConv uses a lightweight convolution which can perform competitively to the best reported self-attention results. Furthermore, they introduce dynamic convolutions (DynamicConv) which are simpler and more efficient than self-attention.
- **Multilingual BERT** (Devlin et al. 2019): Multilingual BERT (mBERT) extends the BERT model to different languages. We download the pre-trained model provided by the authors, and fine-tune on the machine translation datasets.
- **XLM** (Lample and Conneau 2019): We use the released code[2] and the pre-trained data provided by XLM, and further fine-tune the pre-trained model on the corresponding data.
- **MASS** (Song et al. 2019): We conduct experiments with the codes provided by the authors. We set the fragment length k as 50% of the total number of masked tokens in the sentence.

Details We fine-tune our ALM with the Adam optimizer (Kingma and Ba 2015) with a linear warmup (Vaswani et al. 2017). We tune the learning rates based on the performance on the validation set, and the learning rates are 5×10^{-4} for IWSLT14 German-English and 10^{-3} for WMT14 English-German. We use the averaged perplexity over all languages as a criterion for early stopping. The batch size is set to 8192 tokens for all experiments. During decoding, we set the beam size to 8.

Results To prove the effectiveness of ALM, we perform experiments on the English-German and German-English

[2]https://github.com/facebookresearch/XLM

En → De	BLEU(%)
Transformer (Vaswani et al. 2017)	28.40
ConvS2S (Gehring et al. 2017)	25.16
Weighted Transformer (Ahmed, Keskar, and Socher 2017)	28.90
Layer-wise Transformer (He et al. 2018)	29.01
RNMT+ (Chen et al. 2018)	28.50
mBERT (Devlin et al. 2019)	28.64
MASS (Song et al. 2019)	28.92
XLM (Lample and Conneau 2019)	28.88
ALM (this work)	**29.22**

Table 1: Results on WMT14 English-German machine translation task.

De → En	BLEU(%)
Transformer (Vaswani et al. 2017)	34.49
LightConv (Wu et al. 2019)	34.80
DynamicConv (Wu et al. 2019)	35.20
Advsoft (Wang, Gong, and Liu 2019)	35.18
Layer-wise Transformer (He et al. 2018)	35.07
mBERT (Devlin et al. 2019)	34.82
MASS (Song et al. 2019)	35.14
XLM (Lample and Conneau 2019)	35.22
ALM (this work)	**35.53**

Table 2: Results on IWSLT14 German-English machine translation task.

translation tasks. Table 1 and Table 2 show that our ALM has significant improvements over baselines without pre-training or with pre-training methods.

In Table 1, we report the performance of ALM and the baseline models in the WMT14 English-German machine translation dataset. Transformer is an important baseline, and it obtains 28.40 in BLEU score. We also compare ALM with the convolutional baseline ConvS2S, which achieves 25.16. Weighted Transformer and Layer-wise Transformer are two methods to improve the Transformer model, and they get 28.90 and 29.01 in terms of BLEU score. RNMT+ combines the recurrent structure and the multi-head attention components, which yields an improvement to 28.50 BLEU score. Our ALM significantly outperforms these baseline models. We also compare our model with three state-of-the-art pre-training models. mBERT and MASS are unsupervised pre-training models. They achieve 28.64 BLEU score and 28.92 BLEU score, respectively. XLM is a mixture of unsupervised and supervised pre-training models, achieving 28.88 BLEU score. Our ALM reaches 29.22 BLEU score, yielding an improvement of +0.58, +0.30, and +0.34 BLEU scores.

In Table 2, we report the performance of ALM and the baseline models in IWSLT14 German-English machine translation dataset. We first compare our ALM with the supervised models without pre-training. Transformer and its variant Layer-wise Transformer achieves 34.49 and 35.07 in terms of BLEU score. The convolution-based models, LightConv and DynamicConv, achieve 34.80 and 35.20, respectively. Advsoft gets a BLEU score of 35.18. ALM outperforms these baselines, achieving 35.53 in BLEU score. We also compare ALM with three pre-training baselines. It shows that our ALM obtains the best performance and reaches 35.53 BLEU score in this task, outperforming the previous baseline mBERT, MASS, and XLM by +0.71 and +0.39, and +0.31 in terms of BLEU score.

In general, our ALM could achieve significant improvements over all baseline models on two translation tasks. As our method pre-trains the encoder on a large scale cross-lingual corpus, the word representations and encoder could acquire sufficient cross-lingual information. For example, the target phrase can see both its source and target context. This cross-lingual context is helpful for target word generation and understanding the source sentence in a cross-lingual way.

Fine-Tuning on Cross-Lingual Classification

We fine-tune the pre-trained ALM model on XNLI dataset to evaluate the effectiveness of our model. We build a linear classifier on the top of the pre-trained ALM to project the first hidden state of ALM output into the probabilities of each class. We concatenate premise and hypothesis, and feed them into ALM. We evaluate the performance of the fine-tuned model in 15 XNLI languages. Following previous work (Lample and Conneau 2019), we evaluate the model in three different settings: "TRANSLATE-TRAIN", "TRANSLATE-TEST", and "CROSS-LINGUAL TEST". The evaluation metric is the accuracy of the predicted NLI class.

Baselines We compare our methods with three strong baselines, including a supervised method without pre-training, and two pre-training methods:

- **Conneau**: Conneau (Conneau et al. 2018) proposes a BiLSTM model to set up a baseline for XNLI. We report the scores directly from their paper.
- **Multilingual BERT** (Devlin et al. 2019): Multilingual BERT (mBERT) extends the BERT model to different languages, which is also a strong baseline.
- **XLM** (Lample and Conneau 2019): XLM is the state-of-the-art model for cross-lingual pre-training. We report the results of XLM directly from their paper.

Details We fine-tune our ALM with the Adam optimizer (Kingma and Ba 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.997$. We tune the learning rates based on the performance on the validation set, and the learning rates are set to 5×10^{-6}. We set the batch size to 24, and we limit the sentences up to 256 tokens. We set a rate of dropout 0.15 of last layer. We evaluate our model for every 1000 sentences.

Results Table 3 shows the experimental results of our proposed ALM and the baseline models. Following the work of XNLI (Conneau et al. 2018), we evaluate these models in three different settings: "TRANSLATE-TRAIN", "TRANSLATE-TEST", and "CROSS-LINGUAL TEST". In the setting "TRANSLATE-TRAIN", we translate the training set of the English MultiNLI dataset into each XNLI

	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	avg.
Machine translation baselines (TRANSLATE-TRAIN)																
Conneau (Conneau et al. 2018)	73.7	68.3	68.8	66.5	66.4	67.4	66.5	64.5	65.8	66.0	62.8	67.0	62.1	58.2	56.6	65.4
mBERT (Devlin et al. 2019)	81.9	-	77.8	75.9	-	-	-	-	70.7	-	-	76.6	-	-	61.6	-
XLM (Lample and Conneau 2019)	85.0	80.2	80.8	80.3	78.1	79.3	78.1	74.7	76.5	76.6	75.5	78.6	72.3	70.9	63.2	76.7
ALM (this work)	**85.2**	**81.1**	**82.0**	**82.3**	**78.3**	**79.8**	**78.4**	**74.9**	**76.7**	**76.8**	**75.6**	**78.7**	**72.5**	**71.5**	**63.4**	**77.2**
Machine translation baselines (TRANSLATE-TEST)																
Conneau (Conneau et al. 2018)	73.7	70.4	70.7	68.7	69.1	70.4	67.8	66.3	66.8	66.5	64.4	68.3	64.2	61.8	59.3	67.2
mBERT (Devlin et al. 2019)	81.4	-	74.9	74.4	-	-	-	-	70.4	-	-	70.1	-	-	62.1	-
XLM (Lample and Conneau 2019)	85.0	79.0	79.5	78.1	77.8	77.6	75.5	73.7	73.7	70.8	70.4	73.6	69.0	64.7	65.1	74.2
ALM (this work)	**85.2**	**79.1**	**80.0**	**78.4**	**78.0**	**77.8**	**77.1**	**73.9**	**74.2**	**71.2**	**70.5**	**73.8**	**69.2**	**64.8**	**65.3**	**74.6**
Evaluation of cross-lingual sentence encoders (CROSS-LINGUAL TEST)																
Conneau (Conneau et al. 2018)	73.7	67.7	68.7	67.7	68.9	67.9	65.4	64.2	64.8	66.4	64.1	65.8	64.1	55.7	58.4	65.6
mBERT (Devlin et al. 2019)	81.4	-	74.3	70.5	-	-	-	-	62.1	-	-	63.8	-	-	58.3	-
XLM (Lample and Conneau 2019)	85.0	78.7	78.9	77.8	76.6	77.4	75.3	72.5	73.1	76.1	73.2	76.5	69.6	68.4	**67.3**	75.1
ALM (this work)	**85.2**	**79.3**	**79.2**	**78.0**	**76.7**	**78.1**	**76.5**	**73.0**	**73.2**	**76.4**	**73.5**	**78.6**	**69.8**	**69.0**	66.8	**75.6**

Table 3: Cross-lingual natural language inference (XNLI) test accuracy for the 15 languages.

languages (except English), and fine-tune the models on the translated training set. In the setting "TRANSLATE-TEST", we translate the testing set of each XNLI language to English, and evaluate the performance of the models in each translated testing set. In the setting "CROSS-LINGUAL TEST", we fine-tune the models on the English XNLI training set, and evaluate the performance directly in each testing set. We compare our model with Conneau's baseline model, mBERT, and XLM in these three settings.

In the "CROSS-LINGUAL TEST" setting, our ALM significantly outperforms the baseline models. More precisely, ALM obtains 75.6% accuracy on average, while Conneau's baseline achieves 65.6% accuracy, and XLM gets 75.1%. On the Russian and Turkish languages, we outperform the baselines by 1.2% and 0.5% respectively. ALM gets 85.2% accuracy in English testing set, outperforming Conneau's baseline model by 11.5%, BERT by 3.8%, and XLM by 0.2% in terms of accuracy.

In the "TRANSLATE-TRAIN" setting, our ALM reaches 77.2% accuracy in average across different languages, which indicates that ALM can be fine-tuned for any languages to achieve good performance. On the German and French languages, we outperform the baselines by 2.0% and 1.9% respectively. Besides, our ALM achieves higher accuracy than XLM in 15 languages.

In the "TRANSLATE-TEST" setting, our ALM obtains 74.6% average accuracy, while Conneau's baseline achieves 67.2% accuracy, and XLM gets 74.2%. In general, our ALM can outperform these three baselines across different experiment settings.

Discussions and Analysis

We further analyze the advantages of our pre-trained model. We visualize the distribution of our model's word embedding, and compare it with that of Transformer baseline model. We evaluate the performance of our ALM given different parallel data, in order to analyze the benefits of pre-training in the low-resource setting.

Word Embedding Distribution Figure 4 shows the word embedding distributions of Transformer (without pre-training) and ALM (with pre-training). We project the learned word embeddings from high dimension to 2 dimension with the PCA method. We plot both the Chinese word embeddings and the English word embeddings in the same space. The hollow cycles denote Chinese words, while the solid cycles denote English words.

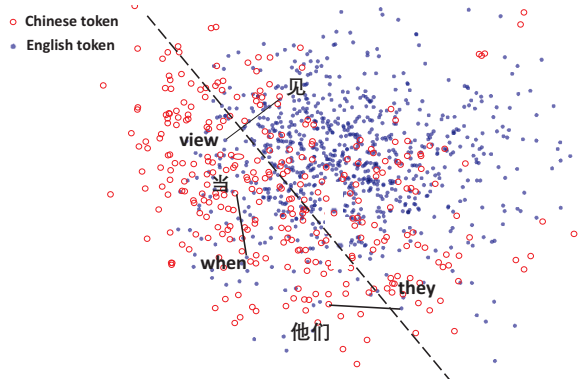
As for the Transformer baseline, the distribution of the Chinese word embeddings is very different from that of the English word embeddings. We draw a dashed line to illustrate the separation of the Chinese word embeddings and the English word embeddings.

As for the pre-trained ALM, the distribution of Chinese word embeddings is similar to that of the English word embeddings. The reason is that we mix Chinese words and English words during training, so the embeddings of both source language and target language can distribute in the same space.

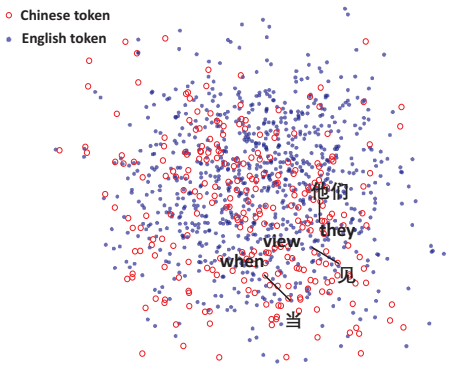
According to Figure 4, it also indicates that the source words and its translated target words have closer distance than that of the Transformer baseline model. There are some cases which are very close to each other in ALM's embedding space but far from each other in the Transformer's embedding space.

It concludes that ALM pre-training method can minor the gap between the embeddings of source language and target language, which is beneficial for the cross-lingual setting.

Low Resource Setting We would like to further analyze the performance of our pre-trained ALM given different sizes of parallel data. Therefore, we randomly shuffle the full parallel training set in the task of IWSLT14 German-to-English translation dataset. Then, we extract the random $K\%$ samples as the fine-tuned parallel data. We set $K = \{10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%, 100\%\}$, and compare our ALM with Transformer baseline model. We randomly extract specific data from the whole sentence pairs. Figure 5 shows the BLEU scores of our models and the baseline. When the parallel data size is small,



(a) Transformer



(b) ALM

Figure 4: Visualization of word embedding in Transformer and ALM.

ALM can outperform Transformer model by a large margin. With the increase of parallel data, the margin gets narrow because of the upper bound of the model capacity. It concludes that ALM pre-training can benefit the performance of Transformer model especially when the training samples are not sufficient.

Related Work

Pre-training and transfer learning are widely used in many tasks of natural language processing. ELMo (Peters et al. 2018) is proposed as a kind of deep contextualized word representation that is pre-trained in the large scale corpus and can be transferred to other tasks. Universal Language Model Fine-tuning (ULMFiT) (Howard and Ruder 2018) is an effective transfer learning method that can be applied to any task in NLP, and includes techniques that are key for fine-tuning a language model. BERT (Devlin et al. 2019) achieves state-of-the-art performance among various pre-training approaches to monolingual NLP tasks. Furthermore, XLM and MASS (Song et al. 2019) obtain more great success in language understanding by pre-training. Unlike BERT that pre-trains only the encoder or the decoder, MASS is carefully designed to pre-train the encoder and decoder jointly by predicting the fragment of the sentence that is masked on the encoder side and predict the masked tokens in the decoder side. By masking the input tokens of

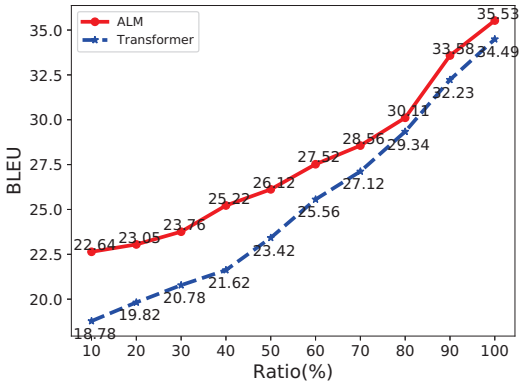


Figure 5: Results of ALM vs Transformer fine-tuning on low-resource data.

the decoder that are unmasked in the source side, MASS can force the decoder to rely more on the source representation other than the previous tokens in the target side for the next token prediction by pre-training with monolingual data. More recently, XLNet (Yang et al. 2019b) proposes a generalized auto-aggressive pre-training method that enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order. RoBERTa (Liu et al. 2019) presents a replication study of BERT pre-training that carefully measures the impact of many key hyperparameters and training data size.

Conclusions

In this work, we propose a novel cross-lingual pre-training method, called Alternating Language Modeling (ALM). First, we randomly substitute the source phrases with the target equivalents to create code-switched sentences. Then, we use these code-switched data to train ALM model to learn to predict words of different languages. We evaluate our pre-training ALM on the downstreams tasks of machine translation and cross-lingual classification. Experiments show that ALM can outperform the previous pre-training methods on three benchmark datasets. In the future work, we will explore the effect of code-switched sentences being used for MASS-like pre-training method.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grand Nos. U1636211, 61672081, 61370126), Beijing Advanced Innovation Center for Imaging Technology (No.BAICIT-2016001) and the Fund of the State Key Laboratory of Software Development Environment (No.SKLSDE-2019ZX-17).

References

Ahmed, K.; Keskar, N. S.; and Socher, R. 2017. Weighted transformer network for machine translation. *CoRR* abs/1711.02132.

Chen, M. X.; Firat, O.; Bapna, A.; Johnson, M.; Macherey, W.; Foster, G.; Jones, L.; Schuster, M.; Shazeer, N.; Parmar, N.; Vaswani, A.; Uszkoreit, J.; Kaiser, L.; Chen, Z.; Wu, Y.; and Hughes, M. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *ACL 2018*, 76–86.

Conneau, A.; Rinott, R.; Lample, G.; Williams, A.; Bowman, S. R.; Schwenk, H.; and Stoyanov, V. 2018. XNLI: evaluating cross-lingual sentence representations. In *EMNLP 2018*, 2475–2485.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL 2019*, 4171–4186.

Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; and Hon, H. 2019. Unified language model pre-training for natural language understanding and generation. *CoRR* abs/1905.03197.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional sequence to sequence learning. In *ICML 2017*, 1243–1252.

He, T.; Tan, X.; Xia, Y.; He, D.; Qin, T.; Chen, Z.; and Liu, T. 2018. Layer-wise coordination between encoder and decoder for neural machine translation. In *NeurIPS 2018*, 7955–7965.

Howard, J., and Ruder, S. 2018. Universal language model fine-tuning for text classification. In *ACL 2018*, 328–339.

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

Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; Dyer, C.; Bojar, O.; Constantin, A.; and Herbst, E. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007*, 177–180.

Koehn, P.; Och, F. J.; and Marcu, D. 2003. Statistical phrase-based translation. In *NAACL 2003*, 48–54.

Kunchukuttan, A.; Mehta, P.; and Bhattacharyya, P. 2018. The IIT bombay english-hindi parallel corpus. In *LREC 2018*, 3473–3476.

Lample, G., and Conneau, A. 2019. Cross-lingual language model pretraining. *CoRR* abs/1901.07291.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR* abs/1907.11692.

Min, S.; Seo, M. J.; and Hajishirzi, H. 2017. Question answering through transfer learning from large fine-grained supervision data. In *ACL 2017*, 510–517.

Och, F. J., and Ney, H. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics* 29(1):19–51.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL 2002*, 311–318.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *NAACL 2018*, 2227–2237.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/language-unsupervised/language_ understanding_paper. pdf*.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Salant, S., and Berant, J. 2018. Contextualized word representations for reading comprehension. In *NAACL 2018*, 554–559.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural machine translation of rare words with subword units. In *ACL 2016*, 1715–1725.

Song, K.; Tan, X.; Qin, T.; Lu, J.; and Liu, T. 2019. MASS: masked sequence to sequence pre-training for language generation. In *ICML 2019*, 5926–5936.

Tay, Y.; Luu, A. T.; and Hui, S. C. 2018. Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference. In *EMNLP 2018*, 1565–1575.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS 2017*, 5998–6008.

Wang, D.; Gong, C.; and Liu, Q. 2019. Improving neural language modeling via adversarial training. In *ICML 2019*, 6555–6565.

Wu, F.; Fan, A.; Baevski, A.; Dauphin, Y. N.; and Auli, M. 2019. Pay less attention with lightweight and dynamic convolutions. In *ICLR 2019*.

Yang, W.; Xie, Y.; Lin, A.; Li, X.; Tan, L.; Xiong, K.; Li, M.; and Lin, J. 2019a. End-to-end open-domain question answering with bertserini. In *NAACL 2019*, 72–77.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J. G.; Salakhutdinov, R.; and Le, Q. V. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR* abs/1906.08237.

Yu, S.; Indurthi, S. R.; Back, S.; and Lee, H. 2018. A multi-stage memory augmented neural network for machine reading comprehension. In *ACL 2018*, 21–30.

Ziemski, M.; Junczys-Dowmunt, M.; and Pouliquen, B. 2016. The united nations parallel corpus v1.0. In *LREC 2016*, 3530–3534.