CROSS-LINGUAL TRANSFER LEARNING FOR PRE-TRAINED CONTEXTUALIZED LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Though the pre-trained contextualized language model (PrLM) has made a significant impact on NLP, training PrLMs in languages other than English can be impractical for two reasons: other languages often lack corpora sufficient for training powerful PrLMs, and because of the commonalities among human languages, computationally expensive PrLM training for different languages is somewhat redundant. In this work, building upon the recent works connecting cross-lingual transfer learning and neural machine translation, we thus propose a novel cross-lingual transfer learning framework for PrLMs: TRELM. To handle the symbol order and sequence length differences between languages, we propose an intermediate "TRILayer" structure that learns from these differences and creates a better transfer in our primary translation direction, as well as a new cross-lingual language modeling objective for transfer training. Additionally, we showcase an embedding aligning that adversarially adapts a PrLM's non-contextualized embedding space and the TRILayer structure to learn a text transformation network across languages, which addresses the vocabulary difference between languages. Experiments on both language understanding and structure parsing tasks show the proposed framework significantly outperforms language models trained from scratch with limited data in both performance and efficiency. Moreover, despite an insignificant performance loss compared to pre-training from scratch in resource-rich scenarios, our transfer learning framework is significantly more economical.

1 INTRODUCTION

Recently, the pre-trained contextualized language model has greatly improved performance in natural language processing tasks and allowed the development of natural language processing to extend beyond the ivory tower of research to more practical scenarios. Despite their convenience of use, PrLMs currently consume and require increasingly more resources and time. In addition, most of these PrLMs are concentrated in English, which prevents the users of different languages from enjoying the fruits of large PrLMs. Thus, the task of transferring the knowledge of language models from one language to another is an important task for two reasons. First, many languages do not have the data resources that English uses to train such massive and data-dependent models. This causes a disparity in the quality of models available to English users and users of other languages. Second, languages share many commonalities - for efficiency's sake, transferring knowledge between models rather than wasting resources training new ones is preferable. Multilingual PrLMs (mPrLMs) also aim to leverage languages' shared commonalities and lessen the amount of language models needed, but they accomplish this by jointly pre-training on multiple languages, which means when they encounter new languages, they need to be pre-trained from scratch again, which causes a waste of resources. This is distinct from using TreLM to adapt models to new languages because TreLM foregoes redoing massive pre-training and instead presents a much more lightweight approach for transferring a PrLM. mPrLMs can risk their multilingualism and finetune on a specific target language, but we will demonstrate that using TreLM to transfer an mPrLM actually leads to better performance than solely finetuning. Therefore, in order to allow more people to benefit from the PrLM, we aim to transfer the knowledge stored in English PrLMs to models for other languages. The differences in training for new languages with mPrLMs and TRELM are shown in Figure 1.

Machine translation, perhaps the most common cross-lingual task, is the task of automatically converting source text in one language to text in another language; that is, the machine translation model

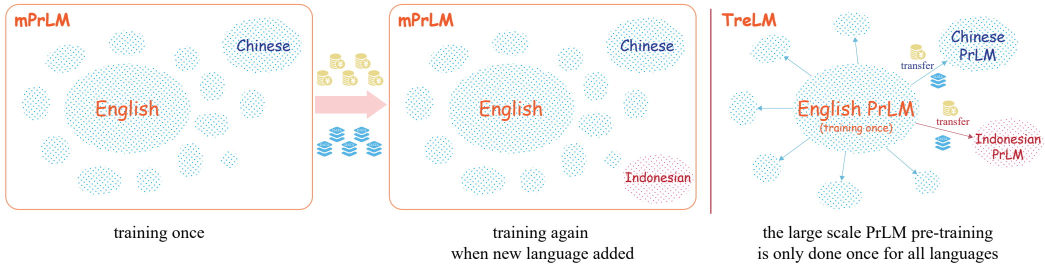


Figure 1: Typical language transfer scenarios for mPrLM and TRELM.

converts the input consisting of a sequence of symbols in some language into a sequence of symbols in another language; i.e., it follows a sequence-to-sequence paradigm. Language has been defined as "*a sequence that is an enumerated collection of symbols in which repetitions are allowed and order does matter*" (Chomsky, 2002). From this definition, we can derive three important differences in the sequences of different languages: *symbol sets*, *symbol order*, and *sequence length*, which can also be seen as three challenges for machine translation and three critical issues that we need to address in migrating a PrLM across languages.

In this work, to resolve these critical differences in language sequences, we propose a novel framework that enables rapid cross-lingual transfer learning for PrLMs and reduces loss when only limited monolingual and bilingual data are available. To address the first aforementioned issue, symbol sets, we employ a new shared vocabulary and adversarially align our target embedding space with the raw embedding of the original PrLMs. For the symbol order and sequence length issues, our approach draws inspiration from neural machine translation methods that overcome the differences between languages (Bahdanau et al., 2014), and we thus propose a new cross-lingual language modeling objective, CdLM, which tasks our model with predicting the tokens for a from its parallel sentence in the target language. To facilitate this, we also propose a new "TRILayer" structure, which acts as an intermediary layer that evenly splits our models' encoder layers set into two halves and serves to convert the source representations to the length and order of the target language. Using parallel corpora for a given language pair, we train two models (one in each translation direction) initialized with the desired pre-trained language model's parameters. Combining the first half of our target-to-source model's encoder layer set and the second half of our source-to-target model's encoder layer set, we are thus able to create a full target-to-target language model. During training, we use three separate phases for the proposed framework, where combinations of Masked Language Modeling (MLM), the proposed CdLM, and other secondary language modeling objectives are used.

We conduct extensive experiments on Chinese and Indonesian, as well as German and Japanese (shown in Appendix 10), in challenging situations with limited data and transfer knowledge from English PrLMs. On several natural language understanding and structure parsing tasks, BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b) PrLM models that we migrate using our proposed framework improve the performance of downstream tasks compared to monolingual models trained from scratch and models pre-trained in a multilingual setting. Moreover, statistics show that our framework also has advantages in terms of training costs.

2 RELATED WORK

Because of neural networks' reliance on heavy amounts of data, transfer learning has been an increasingly popular method of exploiting otherwise irrelevant data in recent years. It has seen many applications and has been used particularly often in Machine Translation (Zoph et al., 2016; Dabre et al., 2017; Qi et al., 2018; Nguyen & Chiang, 2017; Gu et al., 2018; Kocmi & Bojar, 2018; Neubig & Hu, 2018; Kim et al., 2019; Aji et al., 2020), in which transfer learning is generally used to improve translation performance in a low resource scenario using the knowledge of a model trained in a high resource scenario. In addition to cross-lingual situations, transfer learning has also been applied to adapt across domains in the POS tagging (Schnabel & Schütze, 2013) and syntactic parsing (McClosky et al., 2010; Rush et al., 2012) tasks, for example, as well as specifically for adapting language models to downstream tasks (Chronopoulou et al., 2019; Houlsby et al., 2019). One particular difference between our method and many transfer learning methods is that we do not exactly use the popular ”Teacher-Student” framework of transfer learning, which is particularly often used in knowledge distillation (Hinton et al., 2015; Sanh et al., 2020) - transferring knowledge from a larger model to a smaller model. We instead use two ”student” models, and unlike traditional methods, these student models do not share a target space with their teacher (the language is different), and their parameters are initialized with the teacher's parameters rather than being probabilistically guided by the teacher during training.

When using transfer learning for cross-lingual training, there have been various solutions for the vocabulary mismatch. Zoph et al. (2016) did not find vocabulary alignment to be necessary, while Nguyen & Chiang (2017) and Kocmi & Bojar (2018) used joint vocabularies, and Kim et al. (2019) made use of cross-lingual word embeddings. One particular work that inspired us is that of Lample et al. (2018), who also used an adversarial approach to align word embeddings without any supervision while achieving competitive performance for the first time. This succeeded the work of Zhang et al. (2017), who also used an adversarial method but did not achieve the same performance. Also like our aligning method, Xu et al. (2018) took advantage of the similarities in embedding distributions and cross-lingually transferred monolingual word embeddings by simultaneously optimizing based on distributional similarity in the embedding space and the back-translation loss.

Several works have also explored adapting the knowledge of large contextualized pre-trained language models to more languages, which pose a much more complicated problem compared to transferring non-contextualized word embeddings. The previous mainstream approach for accommodating more languages is using mPrLMs. Implicitly joint multilingual models, such as m-BERT (Devlin et al., 2019), XLM (Conneau & Lample, 2019), XLM-R (Conneau et al., 2019), and mBART (Liu et al., 2020), are usually evaluated on multi-lingual benchmarks such as XTREME (Hu et al., 2020) and XGLUE (Liang et al., 2020), while some works use bilingual dictionaries or sentences for explicit cross-lingual modeling with mPrLMs (Schuster et al., 2019; Mulcaire et al., 2019; Liu et al., 2019a; Cao et al., 2020). Transferring monolingual PrLMs, another research branch, is relatively new. Artetxe et al. (2020) presented a monolingual transformer-based masked language model that was competitive with multilingual BERT when transferred to a second language. To facilitate this, they did not rely on a shared vocabulary or joint training (to which multilingual models' performance is often attributed) and instead simply learned a new embedding matrix through MLM in the new language while freezing parameters of all other layers. Tran (2020) used a similar approach, though instead of randomly initialized embeddings, he used a sparse word translation matrix on English embeddings to create word embeddings in the target language, reducing the training cost of the model.

3 TRELM

Cross-lingual Transfer Learning for Language Modeling (TRELM) is a framework that rapidly migrates existing PrLMs. In this framework, the embedding space of a source language is linearly aligned with that of a target using an adversarial embedding alignment, which we experimentally verified was effective due to shared spatial structure similarities (refer to Appendix A.1 for details). Leveraging joint learning, we propose a novel pre-training objective, CdLM, and unify it with MLM into one format. In regards to model structure, we proposed TRILayer, an intermediary transfer layer, to support language conversion during the CdLM training process.

3.1 TRILAYER AND CdLM

For the disparities in symbol sets of different languages and different pre-trained models, we employ embedding space alignment, while for the issues of the symbol order and sequence length, unlike previous work, we do not assume that the model can implicitly learn these differences, and we instead leverage language embeddings and explicit alignment information and propose a novel Cross-Lingual Language Modeling (CdLM) training objective and a Transfer Learning Intermediate Layer (TRILayer) structure as a pivot layer in the model to bridge the differences of the two languages. To clearly explain our training approach, we take the popular PrLM BERT as a basis for introduction.

In the original BERT (as shown in Figure5(a)), Transformer (Vaswani et al., 2017) is taken as the backbone of model, which takes tokens and their positions in a sequence as input before encoding this sequence into a contextualized representation using multiple stacked multi-head self-attention layers. During the pre-training process, BERT predominantly adopts an MLM training objective, in which a `[MASK]` (also written as `[M]`) token is used to replace a token in the sequence selected by a predetermined probability, and the original token is predicted as the gold target. Formally speaking, given a sentence $X = \{x_1, x_2, ..., x_T\}$ and $\mathcal{M}$, the set of masked positions, the training loss $\mathcal{L}_{MLM}$ for the MLM objective is:

$$\mathcal{L}_{MLM}(\theta_{LM}) = -\sum_{i=1}^{|\mathcal{M}|} \log P_{\theta_{LM}}(x_{\mathcal{M}_i}|X_{\setminus\mathcal{M}}),$$

where θ_{LM} are the parameters of BERT, $|\mathcal{M}|$ is the length of set $\mathcal{M}$, and $X_{\setminus\mathcal{M}}$ indicates the sequence after masking. An example of MLM training is shown in the top-left region of Figure 5.

Much work in the field of machine translation suggests that the best way to transfer learning across languages is through translation learning because the machine translation model must address all three of the above-described language differences in the training process. Therefore, we take inspiration from the design of machine translation, especially the design of non-autoregressive machine translation, and propose a Cross-Lingual Language Modeling (CdLM) objective. CdLM is just like a traditional language modeling objective, except across languages, so given an input of source tokens, it generates tokens in a separate target language. We describe the differences between CdLM and related MLM variants (such as Translation Language Modeling (TLM) and BRidge Language Modeling (BRLM)) in Appendix A.4. With this proposed objective, we aim to make as few changes as possible to the existing PrLM and thus introduce a Translation/Transfer Intermediate Layer ("TRILayer") structure, which bridges two opposing half-models to create our final model.

First, in the modified version of BERT for transfer learning, we add a language embedding E_{lng} following the practice of (Conneau & Lample, 2019) to indicate the current language being processed by the model. This is important because the model will handle both the source and target languages simultaneously in 2 of our 3 training phases (described in next subsection). The new input embedding is:

$$E_{inp} = E_{wrd} + E_{seg} + E_{pos} + E_{lng},$$

where E_{wrd}, E_{seg}, and E_{pos} are the word (token) embedding, segment embedding, and position embedding, respectively.

Next, we denote N as the number of stacked Transformer layers ($L = \{l_1, l_2, ..., l_N\}$) in BERT and split the BERT layers into two halves $L_{\leq \frac{N}{2}} = \{1_1, ..., l_{\frac{N}{2}}\}$ and $L_{> \frac{N}{2}} = \{l_{\frac{N}{2}+1}, l_{\frac{N}{2}+2}, ..., l_N\}$. The TRILayer is placed between the two halves (making the total number of layers $N+1$) and functions as a pivot. In the $L_{\leq \frac{N}{2}}$ half, the input embedding is encoded by its Transformer layers to hidden states $H_i = \text{TRANSFORMER}_i(H_{i-1})$, in which $H_0 = E_{inp}$ and TRANSFORMER_i indicates the i-th Transformer layer in the model.

Before the outputs of the $L_{\leq \frac{N}{2}}$ half are fed into the TRILayer, the source hidden representation $H_{\frac{N}{2}}$ is reordered according to new order O. During CdLM training, for source language sentence $X = \{x_1, x_2, ..., x_T\}$, a possible translation sentence $Y = \{y_1, y_2, ..., y_{T'}\}$ is provided. To find the new order, explicit alignment information between the transfer source and target sentences is obtained using an unsupervised external aligner tool. We define the source-to-target alignment pair set as:

$$\mathcal{A}_{X\rightarrow Y} = \text{ALIGN}(X, Y) = \{(x_{\text{ALNIDX}(y_1)}, y_1), (x_{\text{ALNIDX}}(y_2), y_2), ..., (x_{\text{ALNIDX}}(y_{T'}), y_{T'})\},$$

where $\text{ALNIDX}(\cdot)$ is a function that returns the alignment index in the source language or x_{null} when there is no explicit alignment between the token in the target language and any source language token. x_{null} represents a special placeholder token `[P]` that is always appended to the inputs. Finally, the source hidden representation $H_{\frac{N}{2}}$ is reordered according to the new order $O = \{\text{ALNIDX}(y_1), \text{ALNIDX}(y_2), ..., \text{ALNIDX}(y_{T'})\}$ from alignment set $\mathcal{A}_{X\rightarrow Y}$, creating $H^O_{\frac{N}{2}}$.

Thus, the resultant hidden representation $H^O_{\frac{N}{2}}$ is in the order of the target language and is consistent with the target sequence in length, making it usable for language modeling prediction. Unfortunately, the position information is lost in reordering. To combat this, the position embedding and language embedding will be reintegrated as follows:

$$H_{\text{TL}} = \text{TRANSFORMER}_{\text{TL}}(H^{O}_{\frac{N}{2}} + E_{lng^Y} + E_{pos}),$$

where H_{TL} is the output of TRILayer, $\text{TRANSFORMER}_{\text{TL}}$ is the Transformer structure inside the TRILayer, and $\mathbf{E}_{lng^Y}$ is the target language embedding. Next, the H_{TL} is encoded in the $L_{>\frac{2}{N}}$ half as done for the $L_{\leq\frac{N}{2}}$ half (let $H_{\frac{N}{2}} = H_{\text{TL}}$ for the $L_{>\frac{N}{2}}$ half) to predict the final full sequence of the target language. The model is trained to minimize the loss $\mathcal{L}_{CdLM}$, which is:

$$\mathcal{L}_{CdLM}(\theta_{LM}) = -\sum_{i=1}^{T'} \log P_{\theta_{LM}}(y_i|X, \mathcal{A}_{X\to Y}).$$

To enable MLM and CdLM to train models simultaneously rather than through successive optimization, we provide a unified view for MLM and CdLM language modeling:

$$\mathcal{L}_{ULM}(\theta_{LM}) = -\sum_{i=1}^{T_{\max}} \mathbb{1}(i \in C) \log P_{\theta_{LM}}(w_i|S, \mathcal{A}),$$

where $T_{\max}$ denotes the maximum sequence length for language modeling, S is the input sequence, w_i is the i-th token in output sequence W, C is the set of positions to be predicted, and A is the alignment between the input and output sequence. Both the input and output sequences are padded to the maximum sequence length $T_{\max}$ during training. $\mathbb{1}(i \in C)$ represents the indicator function and equals 1 when i-th position exists in the set for the parse to be predicted and 0 otherwise. In MLM, $S = X_{\setminus C}$, $\mathcal{A} = \{(1, 1), (2, 2), ..., (T_{\max}, T_{\max})\}$ is a successive alignment, and $W = X$, while in CdLM, $S = X$, $\mathcal{A} = \mathcal{A}_{X\to Y}$, and $W = Y$. Due to the unified language modeling abstractions of MLM and CdLM, the input and output forms, as well as the internal logic of their models, are the same. Therefore, models can be trained with the two objectives in the same mini-batch, which enhances the stability of transfer training.

3.2 TRIPLE-PHASE TRAINING

In our TRELM framework, the whole training process is divided into three phases with different purposes but the same design goal: minimize the number of parameter updates as much as possible to speed up convergence and enhance training stability. The three phases are *commonality training*, *transfer training*, and *language-specific training*. In the *commonality learning* phase, only the target language MLM objective is used, while in the *transfer learning* phase, CdLM and target language MLM objectives are both used at the same time, and in the final *language-specific learning* phase, target language MLM and other secondary language modeling objectives are adopted.

Commonality Training Though languages are very different on the surface, they also share a lot of underlying commonalities, often called linguistic universals or cross-linguistic generalizations. We therefore take advantage of these commonalities between languages and jointly learn the transferring source and target languages. In this phase, the parameters of the position, segment embedding, and Transformer layers are initialized with original BERT, the TRILayer is initialized with parameters of Transformer layer $L_{\frac{N}{2}}$, the word embedding is initialized with the output of the adversarial embedding aligning, and orthogonal weight initializations are adopted for the language embedding. For this phase, the model is trained by joint MLM with monolingual inputs from both the source and target languages. Moreover, in this training process, to make convergence fast and stable, the parameters of BERT's backbone (Transformer) layers are fixed; only the embeddings and TRILayer are updated by the gradient-based optimization based on the joint MLM loss. The final model obtained in this phase is denoted as θ^{ct}_{LM}.

Transfer Training Since the model is not pre-trained from scratch, making the model aware of changes in inputs is a critical factor for a maximally rapid and accurate migration in the case of limited data. Since there is not enough monolingual data in the target language to allow the model to adapt to the new language, we use the supervisory signal from the two languages' differences and leverage parallel corpora to directly train the model. Specifically, we split the original BERT transformer layers into two halves. With a parallel corpus from the source language to the target language and one from the target language to the source language, we train two corresponding models, both of which are initialized using the parameters learned in the previous phase. In the source-to-target model, only the upper half of the encoder layers is trained, and the lower half is kept fixed, while the converse is true for the target-to-source model. TRILayer then provides cross-lingual order and length adjustment, which is similar to the behavior of a neural machine translation model. Thus, we create two reciprocal models: one whose upper half can handle the target language, and one whose lower half can handle it, which we connect via the TRILayer. Finally, the two trained models are combined as θ^{tt}_{LM}. We describe the full procedure in Algorithm 1.

Algorithm 1 Transfer Training of Pre-trained Contextualized Language Models

Input: The commonality pre-trained model parameters θ^{ct}_{LM}, Languages $\mathfrak{L} = \{lng^X, lng^Y\}$, Parallel training set $\mathfrak{P} = \{(X^{\mathfrak{L}_0}_i, X^{\mathfrak{L}_1}_i)\}^{|\mathfrak{P}|}_{i=1}$, Number of training steps $\mathcal{K}$

for j in 0, 1 **do**
  Initialize model parameters $\theta^{\mathfrak{L}_j \to \mathfrak{L}_{(1-j)}}_{\text{LM}} \leftarrow \theta^{ct}_{\text{LM}}$
  if $j == 0$ **then**
    Fix the parameters of $L_{\leq \frac{N}{2}}$ half of $\theta^{\mathfrak{L}_j \to \mathfrak{L}_{(1-j)}}_{\text{LM}}$
  else
    Fix the parameters of $L_{> \frac{N}{2}}$ half of $\theta^{\mathfrak{L}_j \to \mathfrak{L}_{(1-j)}}_{\text{LM}}$
  end if
  for step in 1, 2, 3, ..., $\mathcal{K}$ **do**
    Sample batch $(X^{\mathfrak{L}_j}, X^{\mathfrak{L}_{(1-j)}})$ from $\mathfrak{P}$.
    Alignment information $\mathcal{A}$: $\mathcal{A}_{\mathfrak{L}_j \to \mathfrak{L}_{(1-j)}} \leftarrow \text{ALIGN}(X^{\mathfrak{L}_j}, X^{\mathfrak{L}_{(1-j)}})$
    CdLM Loss: $\mathcal{L}_{\text{CdLM}} \leftarrow -\sum \log P_{\theta^{\mathfrak{L}_j \to \mathfrak{L}_{(1-j)}}_{\text{LM}}}(X^{\mathfrak{L}_{(1-j)}} | X^{\mathfrak{L}_j}, \mathcal{A}_{\mathfrak{L}_j \to \mathfrak{L}_{(1-j)}})$
    Masked version of $X^{\mathfrak{L}_1}$: $X^{\mathfrak{L}_1}_{\setminus M} \leftarrow \text{MASK}(X^{\mathfrak{L}_1})$
    MLM Loss: $\mathcal{L}_{\text{MLM}} \leftarrow -\sum \log P_{\theta^{\mathfrak{L}_j \to \mathfrak{L}_{(1-j)}}_{\text{LM}}}(X^{\mathfrak{L}_1}_{M} | X^{\mathfrak{L}_1}_{\setminus M})$
    CdLM+MLM Update: $\theta^{\mathfrak{L}_j \to \mathfrak{L}_{(1-j)}}_{\text{LM}} \leftarrow \text{optimizer\_update}(\theta^{\mathfrak{L}_j \to \mathfrak{L}_{(1-j)}}_{\text{LM}}, \mathcal{L}_{\text{CdLM}}, \mathcal{L}_{\text{MLM}})$
  end for
end for
Combine the two obtained models as θ^{tt}_{LM} by choosing the $L_{> \frac{N}{2}}$ half model parameters from model $\theta^{\mathfrak{L}_0 \to \mathfrak{L}_1}_{\text{LM}}$ and $L_{\leq \frac{N}{2}}$ half model parameters from model $\theta^{\mathfrak{L}_1 \to \mathfrak{L}_0}_{\text{LM}}$ and average the other parameters (such as embedding and TRILayer parameters) of the two models

Output: Learned model θ^{tt}_{LM}

Language-specific Training During the language-specific training phase, we only use the monolingual corpus of the target language and further strengthen the target language features for the model obtained in the transfer training phase. We accomplish this by using the MLM objective and other secondary objectives such as Next Sentence Prediction (NSP).

4 EXPERIMENTS

In this section, we discuss the details of the experiments undertaken for this work. We conduct experiments based on English PrLMs[1]. We transfer via English-to-Chinese and English-to-Indonesian directions for the purpose of comparing with previous recent work. We describe the training details and parameters in Appendix A.5. From English to Chinese and English to Indonesian, we transfer two pre-trained contextualized language models: BERT and RoBERTa. Our performance evaluation on the migrated models is mainly conducted on two types of downstream tasks: language understanding and language structure parsing. Please refer to Appendix A.6 for introductions of tasks and baselines and Appendix A.7 for an ablation study. We note that the comparisons between models trained using TRELM and the monolingual and multilingual PrLMs trained from scratch on the target language (see Table 1) is only for illustrating the relative performance loss of the model

[1]Our code is available at `https://github.com/agcbi2017/TreLM`.

Models	Sentence-Pair			Single-Sentence			MRC		
	AFQMC (acc)	CMNLI (acc)	CSL (acc)	TNEWS (acc)	IFLYTEK (acc)	WSC (acc)	CMRC18 (EM)	CHID (acc)	C^3 (acc)
Single-task single models on dev									
BERT-base	74.16	79.47	79.63	56.09	60.37	63.48	64.77	82.20	65.70
m-BERT-base	70.29	79.03	79.26	53.71	56.63	62.82	63.93	80.00	63.81
BERT-small	69.71	66.54	69.73	53.22	45.40	53.29	50.23	68.55	56.84
TRI-BERT-base	72.98	79.44	79.34	55.45	58.36	63.00	63.96	80.94	65.06
TRI-BERT-large	73.41	80.50	80.59	56.20	60.99	64.76	66.35	82.61	66.08
TRI-RoBERTa-base	73.51	80.47	80.26	55.98	61.65	63.92	65.76	82.02	65.98
TRI-RoBERTa-large	74.55	81.68	81.35	57.02	62.24	65.16	67.29	83.53	66.79

Table 1: Results on the CLUE development datasets.

Models	CTB 5.1		CoNLL-09 ZH		UD 2.3 ID GSD	
	UAS	LAS	UAS	LAS	UAS	LAS
(Dozat & Manning, 2016)	89.30	88.23	88.90	85.38	85.93*	78.21*
BERT-base	91.48	89.24	92.63	89.59	86.69†	77.97†
m-BERT-base	89.84	87.33	90.98	87.70	87.19	79.10
TRI-BERT-base	89.96	87.43	90.94	87.81	87.56	79.44
TRI-RoBERTa-base	90.30	87.82	91.62	88.29	88.42	79.95
EN-ID-ZH XLM-MLM	89.25	86.98	90.00	86.98	86.64	77.74
EN-ID-ZH XLM-MLM+TLM	89.58	87.16	90.53	87.37	86.96	78.01
TRI-XLM-en-2048	90.66	88.20	91.95	88.61	88.67	80.30

Table 2: Dependency parsing results on the Chinese PTB 5.1, CoNLL-2009 Chinese, and Universal Dependency Indonesian GSD test sets. "∗" indicates that the result was from our own experiments on the UD dataset based on Dozat & Manning (2016)'s model, and "†" indicates that the official BERT paper did not provide Indonesian BERT-base, so we used IndoBERT-base pre-trained by (Wilie et al., 2020).

produced by TreLM. These models are not directly comparable, as we intentionally use less data to train models when using TreLM. Continuing to pre-train the PrLMs on the target language would also obviously further improve their performance, but this is not our main focus.

Language Understanding We first compare the PrLMs transferred by TreLM alongside the results the existing monolingual pre-trained `BERT-base-chinese` and the multilingual pre-trained `BERT-base-multilingual` in Table 1 using the CLUE benchmark.

When comparing with the same model architecture, taking BERT as an example, our model TRI-BERT-base exceeds m-BERT-base and BERT-small and is slightly weaker than original BERT-base. Compared with BERT-small, which is trained from scratch for a longer time, our TRI-BERT-base generally achieves better results on these NLU tasks. This demonstrates that because of the commonalities of languages, models for languages with relatively few resources can benefit from language models pre-trained on languages with richer resources, which confirms our cross-lingual transfer learning framework's effectiveness.

m-BERT is another potential language model migration scheme and has the advantage of supporting multiple languages at the same time; however, in order to be compatible with multiple languages, the unique characteristics of each language are neglected. Our TRI-BERT, which is built on top of BERT-base, instead focuses on and highlights language differences during the transfer learning process, which leads to an increase in performance compared to m-BERT. When TRI-BERT and TRI-RoBERTa have the same model size, TRI-RoBERTa outperforms TRI-BERT, which is consistent with the performance differences between the original RoBERTa and BERT, indicating that our migration approach maintains the performance advantages of PrLMs.

Models	CoNLL-09		
	P	R	F_1
(Cai et al., 2018)	84.7	84.0	84.3
+BERT-base	86.86	87.48	87.17
+m-BERT-base	85.17	85.53	85.34
+TRI-BERT-base	86.15	85.58	85.86
+TRI-RoBERTa-base	87.08	86.99	87.03
+TRI-RoBERTa-base (w/o CdLM)	85.77	85.62	85.69

Table 3: Dependency SRL results on the CoNLL-2009 Chinese benchmark.

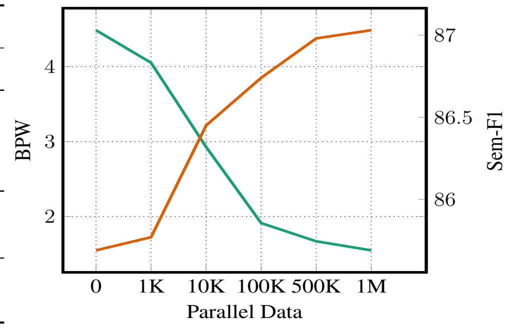


Figure 2: Language modeling effects vs. Parallel data size on the evaluation set.

Language Structure Parsing We report results on dependency parsing for Chinese and Indonesian in Table 2. As shown in the results, the baseline model has been greatly improved for the PrLM. In Chinese, the performance of BERT-base is far superior to m-BERT-base, which highlights the importance of the unique nature of the language for downstream tasks, especially for refined structural analysis tasks. In Indonesian, IndoBERT (Wilie et al., 2020) performs worse than m-BERT, which we suspect is due to IndoBERT's insufficient pre-training. We also compare TRI-BERT-base and IndoBERT-base on Indonesian, whose ready-to-use language resources are relatively small compared to English. We find that although pre-training PrLMs on the available corpora is possible, because of the size of language resources, engineering implementation, etc., our migrated model is more effective than the model pre-trained from scratch. This shows that migrating from the ready-made language models produced from large-scale language training and extensively validated by the community is more effective than pre-training on relatively small and limited language resources. In addition, we also conduct experiments for these pre-trained and migrated models on Chinese SRL.

mPrLMs are another important and competitive approach that can adapt to cross-lingual PrLM applications, so we also include several mPrLMs in our comparison on dependency parsing. Specifically, we used XLM, a monolingual and multilingual PrLM pre-training framework, as our basis. For TreLM, we used `XLM-en-2048`, officially provided by Conneau & Lample (2019), as the source model. The data amount used and the number of training steps are consistent with TRI-BERT/TRI-RoBERTa. In mPrLM, we combined EN, ID, and ZH sentences (including monolingual and parallel sentences) together (10M sentences in total) to train an EN-ID-ZH mPrLM with MLM and TLM objectives. The performance comparison of these three PrLMs on the dependency parsing task is shown in the lower part of Table 2.

From the results, we see mPrLMs pre-trained from scratch have no special performance advantage over TreLM when corpus size is constant, and especially when not using the cross-lingual transfer learning objective TLM, which models parallel sentences. In fact, our TRI-XLM-en-2048 solidly outperforms its two multilingual XLM counterparts. Monolingual PrLMs generally outperform mPrLMs, which likely leads to the performance advantages shown with monolingual migration. Additionally, like our TreLM, mPrLMs can also finetune on only the target language to improve performance, and leveraging TreLM to transfer an mPrLM leads to even further gains, as seen in Table 9 in the appendix.

While the two approaches can compete with each other, they have their own advantages in general. In particular, TreLM is more suitable for transferring additional languages that were not considered in the initial pre-training phase and for low-resource scenarios, while mPrLMs have the advantage of being able to train and adapt to multiple languages at once.

In Table 3, we compared a model migrated without CdLM to the full one. To compensate for the removal of CdLM, we added a monolingual corpus with the same size as the parallel corpora and trained the model with an extra 80K steps, but despite using more target monolingual data and training steps, the performance was still much better when CdLM was included.

5 Discussion

Effects of Parallel Data Scale Since the proposed TreLM framework relies on parallel corpora to learn the language differences explicitly, the sizes of the parallel corpora used are also of concern. We explored the influence of different parallel corpus sizes on the performance of the models transferred with the TRI-RoBERTa-base architecture. The variation curve of BPW score with the size of parallel data is shown in Figure 2. We see that with increasingly more parallel data, BPW gradually decreases, but this decrease slows as the data grows. The effect of the parallel corpora for cross-lingual transfer therefore has a upper bound because when the parallel corpora reaches a certain size, the errors from the alignment extraction tools cannot be ignored, and additionally, due to how lightweight the TRILayer structure is, TRILayers can only contain so much cross-lingual transfer information, which further restricts the growth of the migration performance.

Pre-training Cost vs. Migration Training Cost The training cost is an important factor for choosing whether to pre-train from scratch or to migrate from an existing PrLM. We listed the training data size, model parameters, training hardware, and training time of several public PrLM models and compared them with our models. The comparisons are shown in Table 4. Although the training hardware and engineering implementation of various PrLM models are different, this can still be used as a general reference. When model size is the same, our proposed transfer learning is much faster than pre-training from scratch, and less data is used in the transfer learning process. In addition, the total training time of our large model migration training is less than that of even the base model pre-training when hardware is kept the same. Therefore, the framework we proposed can be used as a good supplementary scheme for the PrLM in situations when time or computing resources are restricted.

Model	**Data**	**BSZ**	**Steps**	**Params**	**Hardware**	**Train Time**	**G/TPU·Days**
ELMo	≈4GB	–	–	96M	3 GTX 1080 GPUs	14d	42
GPT	≈4.5GB	64	≈1.2M	117M	8 P6000 GPUs	25d	200
BERT-base	16GB	256	1M	110M	16 TPUs / 8 V100 GPUs	4d / 11d	64 / 88
BERT-large	16GB	256	1M	340M	64 TPUs / 8 V100 GPUs	4d / 21d	256 / 168
RoBERTa-large	160GB	8K	500K	340M	1024 V100 GPUs	1d	1024
XLNet-large	126GB	2K	500K	360M	512 TPUs	2.5d	1280
ELECTRA-small	16GB	128	1M	14M	1 V100 GPU	4d	4
ELECTRA-base	16GB	256	766K	110M	64 TPUs	4d	256
BERT-small	≈2.4GB	256	240K	15M	8 V100 GPUs	1.5d	12
TRI-BERT-base	↓	128	120K	154M	↓	2.5d	20
TRI-BERT-large		128	120K	398M		5d	40
TRI-RoBERTa-base		128	120K	154M		2.5d	20
TRI-RoBERTa-large		128	120K	398M		5d	40

Table 4: Comparison of the training and migration costs of the PrLMs.

6 Conclusion and Future Work

In this work, we present an effective method of transferring knowledge from a given language's pre-trained contextualized language model to a model in another language. This is an important accomplishment because it allows more languages to benefit from the massive improvements arising from these models, which have been primarily concentrated in English. As a further plus, this method also enables more efficient model training, as languages have commonalities, and models in the target language can exploit these commonalities and quickly adopt these common features rather than learning them from scratch. In future work, we plan to use our framework to transfer other models such as ALBERT and models for more languages. We also aim to develop an unsupervised cross-lingual transfer learning objective to remove the reliance on parallel sentences.

References

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7701–7710, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.688. URL `https://www.aclweb.org/anthology/2020.acl-main.688`.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4623–4637, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.421. URL `https://www.aclweb.org/anthology/2020.acl-main.421`.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. ISSN 2307-387X.

Jiaxun Cai, Shexia He, Zuchao Li, and Hai Zhao. A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware? In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2753–2765, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/C18-1233`.

Steven Cao, Nikita Kitaev, and Dan Klein. Multilingual alignment of contextual word representations. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=r1xCMyBtPS`.

Noam Chomsky. *Syntactic structures*. Walter de Gruyter, 2002.

Alexandra Chronopoulou, Christos Baziotis, and Alexandros Potamianos. An embarrassingly simple approach for transfer learning from pretrained language models. 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)*, pp. 2089–2095, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1213. URL `https://www.aclweb.org/anthology/N19-1213`.

LTD IFLYTEK CO. Iflytek: a multiple categories chinese text classifier. competition official website, 2019.

Alexis Conneau and Guillaume Lample. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pp. 7059–7069, 2019.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

Yiming Cui, Ting Liu, Wanxiang Che, Li Xiao, Zhipeng Chen, Wentao Ma, Shijin Wang, and Guoping Hu. A span-extraction dataset for Chinese machine reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5883–5889, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1600. URL `https://www.aclweb.org/anthology/D19-1600`.

Raj Dabre, Tetsuji Nakagawa, and Hideto Kazawa. An empirical study of language relatedness for transfer learning in neural machine translation. In *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation*, pp. 282–286. The National University (Phillippines), November 2017. URL `https://www.aclweb.org/anthology/Y17-1038`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://www.aclweb.org/anthology/N19-1423`.

Yerai Doval, Jose Camacho-Collados, Luis Espinosa-Anke, and Steven Schockaert. Improving cross-lingual word embeddings by meeting in the middle. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 294–304, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1027. URL `https://www.aclweb.org/anthology/D18-1027`.

Timothy Dozat and Christopher D Manning. Deep biaffine attention for neural dependency parsing. 2016.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 644–648, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/N13-1073`.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL `https://www.aclweb.org/anthology/L18-1550`.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3622–3631, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1398. URL `https://www.aclweb.org/anthology/D18-1398`.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *CoRR*, abs/2003.11080, 2020.

Baijun Ji, Zhirui Zhang, Xiangyu Duan, Min Zhang, Boxing Chen, and Weihua Luo. Cross-lingual pre-training based transfer for zero-shot neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 115–122, 2020.

Yunsu Kim, Yingbo Gao, and Hermann Ney. Effective cross-lingual transfer of neural machine translation models without shared vocabularies, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Tom Kocmi and Ondřej Bojar. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 244–252, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6325. URL `https://www.aclweb.org/anthology/W18-6325`.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *International Conference on Learning Representations*, 2018.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. Xglue: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *arXiv*, abs/2004.01401, 2020.

Pierre Lison and Jörg Tiedemann. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 923–929, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL `https://www.aclweb.org/anthology/L16-1147`.

Qianchu Liu, Diana McCarthy, Ivan Vulić, and Anna Korhonen. Investigating cross-lingual alignment methods for contextualized embeddings with token-level evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 33–43, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/K19-1004. URL `https://www.aclweb.org/anthology/K19-1004`.

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

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*, 2020.

David McClosky, Eugene Charniak, and Mark Johnson. Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 28–36, Los Angeles, California, June 2010. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/N10-1004`.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Phoebe Mulcaire, Jungo Kasai, and Noah A. Smith. Polyglot contextual representations improve crosslingual transfer. 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)*, pp. 3912–3918, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1392. URL `https://www.aclweb.org/anthology/N19-1392`.

Graham Neubig and Junjie Hu. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 875–880, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1103. URL `https://www.aclweb.org/anthology/D18-1103`.

Toan Q. Nguyen and David Chiang. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 296–301, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing. URL `https://www.aclweb.org/anthology/I17-2050`.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 529–535, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2084. URL `https://www.aclweb.org/anthology/N18-2084`.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

Alexander Rush, Roi Reichart, Michael Collins, and Amir Globerson. Improved parsing and POS tagging using inter-sentence consistency constraints. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1434–1444, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D12-1131.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

Tobias Schnabel and Hinrich Schütze. Towards robust cross-domain domain adaptation for part-of-speech tagging. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pp. 198–206, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing. URL https://www.aclweb.org/anthology/I13-1023.

Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5149–5152. IEEE, 2012.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. 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)*, pp. 1599–1613, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1162. URL https://www.aclweb.org/anthology/N19-1162.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. Probing prior knowledge needed in challenging chinese machine reading comprehension. *arXiv preprint arXiv:1904.09679*, 2019.

Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pp. 2214–2218, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf.

Jörg Tiedemann and Santhosh Thottingal. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal, 2020.

Ke Tran. From english to foreign languages: Transferring pre-trained language models, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Bryan Wilie, Karissa Vincentio, Genta Indra Winata, Samuel Cahyawijaya, Xiaohong Li, Zhi Yuan Lim, Sidik Soleman, Rahmad Mahendra, Pascale Fung, Syafri Bahar, et al. Indonlu: Benchmark and resources for evaluating indonesian natural language understanding. *arXiv preprint arXiv:2009.05387*, 2020.

Liang Xu, Xuanwei Zhang, Lu Li, Hai Hu, Chenjie Cao, Weitang Liu, Junyi Li, Yudong Li, Kai Sun, Yechen Xu, et al. Clue: A chinese language understanding evaluation benchmark. *arXiv preprint arXiv:2004.05986*, 2020.

Ruochen Xu, Yiming Yang, Naoki Otani, and Yuexin Wu. Unsupervised cross-lingual transfer of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2465–2474, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1268. URL https://www.aclweb.org/anthology/D18-1268.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pp. 5753–5763, 2019.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pp. 1–21, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-2001. URL https://www.aclweb.org/anthology/K18-2001.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. Adversarial training for unsupervised bilingual lexicon induction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1959–1970, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1179. URL https://www.aclweb.org/anthology/P17-1179.

Chujie Zheng, Minlie Huang, and Aixin Sun. ChID: A large-scale Chinese IDiom dataset for cloze test. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 778–787, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1075. URL https://www.aclweb.org/anthology/P19-1075.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 3530–3534, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://www.aclweb.org/anthology/L16-1561.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. Transfer learning for low-resource neural machine translation, 2016.

A Appendix

A.1 Adversarial Embedding Aligning

Since the symbol sets in different languages are different, the first step in the cross-lingual migration of PrLMs is to supplement or even replace their vocabularies. In our proposed framework, to make the best use of the commonalities between languages, we choose to use a shared vocabulary with multiple languages rather than replace the original language vocabulary with one for the new language. In addition, in current PrLMs, a subword vocabulary is generally adopted in order to better mitigate out-of-vocabulary (OOV) problems caused by limited vocabulary size. To accommodate the introduction of a shared vocabulary, it is necessary to jointly re-train the subword model to ensure that some common words in different languages are consistent in subword segmentation, which leads to the problem that some tokens in the newly acquired subword vocabulary are different from those in the original subword vocabulary, though they belong to the same language. To address this issue, we consider the most complicated case, in which the vocabulary is completely replaced by a new one. Consequently, we assume that there are two embedding spaces: one is the embedding of the original vocabulary, which is well-trained in the language pre-training process, and the other is the embedding of the new vocabulary, yet to be trained.

When considering raw embeddings and non-contextualized embeddings (e.g. Word2vec), it is easy to see their training objectives are similar in theory. The only differences are the addition of context and the change in model structure to accommodate language prediction. Despite these differences, non-contextualized embeddings can be used to simulate the raw embeddings in a PrLM that we aim to replace (refer to Appendix A.2 for a detailed explanation). Although the two embedding spaces we consider are similar in structure, they may be at different positions in the whole real embedding space, so an extra alignment process is required, and although common tokens may exist, due to the inconsistent token granularity from using byte-level byte-pair encoding (BBPE) (Radford et al., 2019), a matching token of the two embedding spaces cannot be utilized for embedding space alignment, as it is likely to represent different meanings. Therefore, inspired by (Lample et al., 2018), we present an adversarial approach for aligning the word2vec embedding space to the PrLM's raw embedding space without supervision. With this approach, we aim to minimize the differences between the two embedding spaces brought about by different similarity forms.

We define $\mathcal{U} = \{u_1, u_2, ..., u_m\}$ and $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ as the two embedding spaces of m and n tokens from the PrLM and word2vec training, respectively. In the adversarial training approach, a linear mapping W is trained to make the spaces $W\mathcal{V} = \{Wv_1, Wv_2, ..., Wv_n\}$ and $\mathcal{U}$ close as possible, while a discriminator D is employed to discriminate between tokens randomly sampled from spaces $W\mathcal{V}$ and $\mathcal{U}$. Let θ_{adv} denote the parameters of the adversarial training model and the probabilities $P_{\theta_{adv}}(\mathbb{1}(z)|z)$ and $P_{\theta_{adv}}(\mathbb{0}(z)|z)$ indicate whether or not the sampling source prediction is the same as its real space for a vector z. Therefore, the discrimination training loss $\mathcal{L}_D(\theta_D|W)$ and the mapping training loss $\mathcal{L}_D(W|\theta_D)$ are defined as:

$$\mathcal{L}_D(\theta_D|W) = -\frac{1}{n}\sum_{i=1}^{n}\log P_{\theta_{adv}}(\mathbb{1}(Wv_i)|Wv_i) - \frac{1}{m}\sum_{i=0}^{m}\log P_{\theta_{adv}}(\mathbb{1}(u_i)|u_i),$$

$$\mathcal{L}_W(W|\theta_D) = -\frac{1}{n}\sum_{i=1}^{n}\log P_{\theta_{adv}}(\mathbb{0}(Wv_i)|Wv_i) - \frac{1}{m}\sum_{i=0}^{m}\log P_{\theta_{adv}}(\mathbb{0}(u_i)|u_i),$$

where θ_D are the parameters of discriminator D, which is implemented as a multilayer perceptron (MLP) with two hidden layers and Leaky-ReLU as the activation function.

During the adversarial training, the discriminator parameters θ_D and W are optimized successively with discrimination training loss and mapping training loss. To enhance the effect of embedding space alignment, we adopted the same techniques of iterative refinement and cross-domain similarity local scaling as Lample et al. (2018) did. While the two embedding spaces in (Lample et al., 2018) both can be updated by gradient, we consider $\mathcal{U}$ as the goal spatial structure and hence fix $\mathcal{U}$ throughout the training process, and we update $\mathcal{W}$ to better align $\mathcal{V}$.

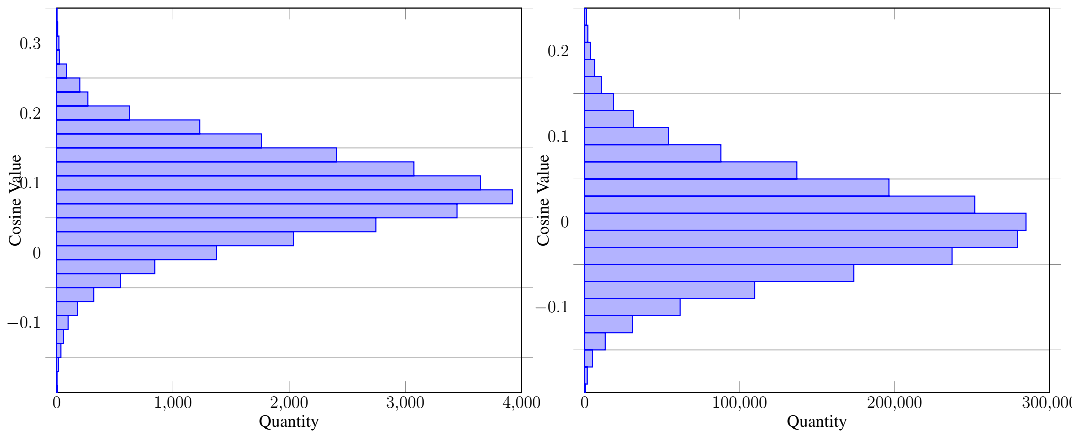


Figure 3: Individual histogram plots of cosine similarity of a single term "genes" with other terms in the vocabularies of BERT-base-cased (left) and FastText cc.en.300d (right).

A.2 Analyzing Non-contextualized Embeddings and PrLMs' Raw Embeddings

Bidirectional PrLMs such as BERT (Devlin et al., 2019) use Masked Language Modeling (MLM) as the training objective, in which the model is required to predict a masked part of the sentence. This training paradigm has no essential difference with word2vec (Mikolov et al., 2013). Word2vec employed a simple single-layer perceptron neural network and restricted the context for the masked part to the sliding window, while recent mainstream PrLMs adopted self-attention-based Transformer as the context encoder, which can utilize the whole sentence as context. Because of this, we speculate that BERT's raw embeddings and word2vec embeddings have a similar nature, and that we can simulate BERT's raw embeddings with the word2vec embeddings through some special designs.

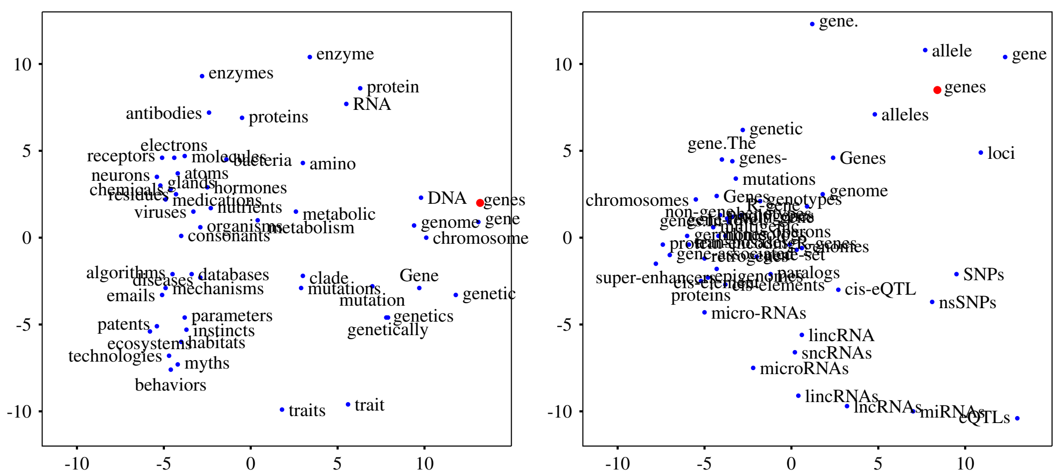
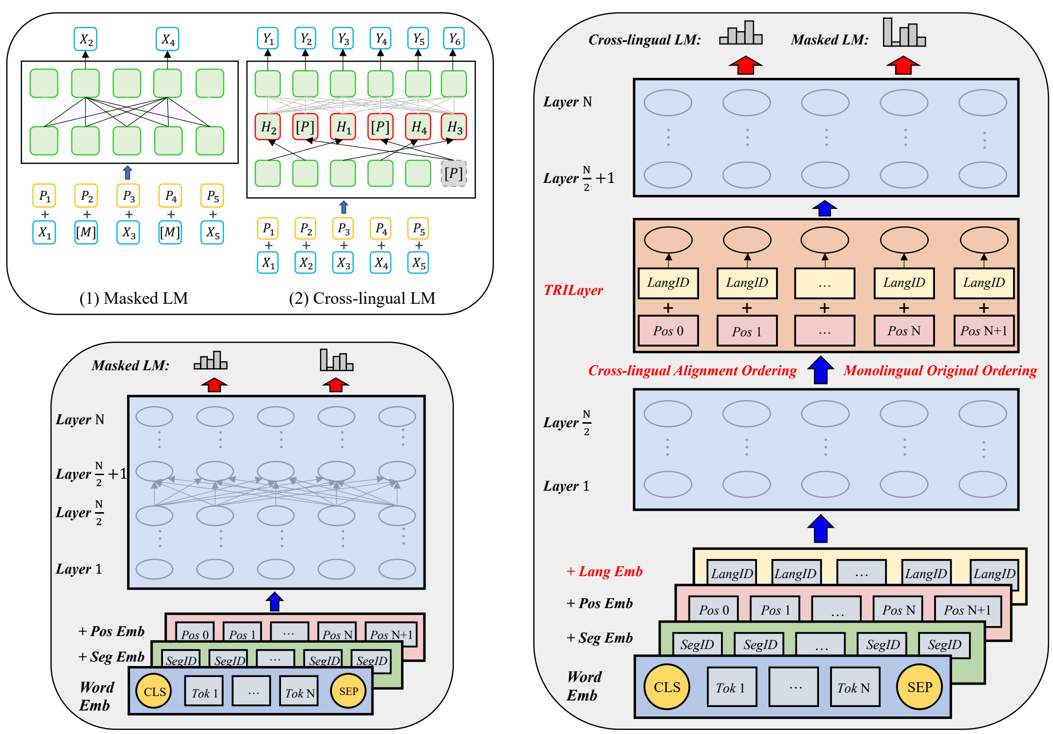


Figure 4: Two-dimensional PCA projection of the embedding vectors representing "genes" and top-50 similar terms in vocabularies of BERT-base-cased and FastText cc.en.300d.

To verify our theory, we studied the important relational nature of embeddings. Specifically, we chose BERT-base-cased's raw embeddings and word2vec-based FastText cc.en.300d embeddings (Grave et al., 2018) and evaluated the cosine similarity of single terms compared to other terms in their vocabularies. An example histogram for the term "genes" is shown in Figure 3. Examining the two types of embeddings, we found that the learned vectors, regardless of the type of similarity (semantic/syntactic/inflections/spelling/etc.) they capture, have a very similar distribution shape. This showed us that the two embedding spaces are similar, and words within them may just have different relations to each other. Thus, our work focuses on aligning the new word2vec embedding space by learning a mapping to the original embedding space to simulate the original embedding allow for a cross-lingual migration of the PrLM.

To illustrate the necessity of embedding alignment, we also took out the top-50 terms closest to the term "genes" in the two embedding spaces, used principal component analysis (PCA) to reduce the vector dimension to 2, and presented it in a two-dimensional figure, as shown in Figure 4. As can be seen from the figure, due to the different language modeling architectures and contexts in FastText and BERT, corresponding points are distributed at different locations in the embedding space. This is why compatibility problems exist when we use the original non-contextualized embeddings to simulate the new embedding and hence why we need to align the embeddings.

A.3 Model architecture in TreLM



(a) Original BERT architecture

(b) Our proposed TRI-BERT architecture

Figure 5: The proposed model architecture, taking BERT as an example.

A.4 MLM, TLM, BRLM, and CdLM

As stated in the original MLM objective, the model can only learn from monolingual data. Though a joint MLM training can be performed across languages, there is still a lack of explicit language cues for guiding the model in distinguishing language differences. Conneau & Lample (2019) proposed a Translation Language Modeling (TLM) objective as an extension of the MLM objective. The TLM objective leverages bilingual parallel sentences by concatenating them into single sequences as in the original BERT and predicts the tokens masked in the concatenated sequence. This encourages the model to predict the masked part in a bilingual context. Ji et al. (2020) further proposed a BRidge Language Modeling (BRLM) built on the TLM, benefiting from explicit alignment information or additional attention layers that encourage word representation alignment across different languages. These MLM variants drive models to learn explicit or implicit token alignment information across languages and have been shown effective in machine translation compared to the original MLM, but for the cross-lingual transfer learning of PrLMs, modeling the order difference and semantic equivalence in different languages is still not enough. Since both contexts in MLM variants have been exposed to the model, whether the prediction of the masked part depends on the cross-lingual context or the context of its own language is unknown, as it lacks explicit clues for cross-lingual training. In our proposed CdLM, we use sentence alignment information for explicit ordering. The model is exposed to both the transfer source and transfer target languages at the same time, during which the input is a sequence of the source language, and the prediction goal is a sequence of the target language. Thus, we convert translation into a cross-language modeling objective, which gives a clear supervision signal for cross-lingual transfer learning.

A.5 Training Details

The initial weights for the migration are `BERT-base-cased`, `BERT-large-cased`, `RoBERTa-base`, and `RoBERTa-large`, which are taken from their official sources. We use English Wikipedia, Chinese Wikipedia, Chinese News, and Indonesian CommonCrawl Corpora for the monolingual pre-training data. For all models migrated in the same direction, regardless of their original vocabulary, we used the same single vocabulary that we trained on the joint language data using the WordPiece Subword scheme (Schuster & Nakajima, 2012). In English-to-Chinese, the vocabulary size is set to 80K and the alphabet size is limited to 30K, while in English-to-Indonesian, the vocabulary size is set to 50K, and the alphabet size is limited to 1K. With the WordPiece vocabulary, we tokenized the monolingual corpus to train the non-contextualized word2vec embedding of subwords. Using the `fastText` (Bojanowski et al., 2017) tool and skipgram representation mode, three embedding sizes 128, 768, and 1024 were trained to be compatible the respective pre-trained language models.

In the "commonality" training phase, we sampled 1M sentences of English Wikipedia and either 1M sentences of Chinese Wikipedia or 1M sentences of Indonesian CommonCrawl for the English-to-Chinese and English-to-Indonesian models. We trained the model with 20K update steps with total batch size 128 and set the peak learning rate to 3e-5.

For the "transfer" training phase, we sampled 1M parallel sentences from the UN Corpus (Ziemski et al., 2016) for English-to-Chinese and 1M parallel sentences from OpenSubtitles Corpus (Lison & Tiedemann, 2016) for English-to-Indonesian. We use the `fastalign` toolkit (Dyer et al., 2013) to extract the tokenized subword alignments for CdLM. The two half models are optimized over 20K update steps, and the batch size and peak learning rate are set to 128 and 3e-5, respectively.

In the final phase, "language-specific" training, 2M Chinese and Indonesian sentences were sampled to update their respective models, training for 80K steps with total batch size 128 and initial learning rate 2e-5. In all the above training phases, the maximum sequence length was set to 512, weight decay was 0.01, and we used Adam (Kingma & Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.999$.

In addition to our migrated pre-trained models, we also pre-trained a `BERT-small`[2] model from scratch with data of the same size as our migration process to compare the performance differences between migration and scratch training. For the `BERT-small` model, we started with the `BERT-base` hyper-parameters and vocabulary but shortened the maximum sequence length from 512 to 128, reduced the model's hidden and token embedding dimension size from 768 to 256, set the batch size to 256, and extended the training steps to 240K.

Our TRI-BERT -* and TRI-RoBERTa-* all used the same amount of training data (2M target language monolingual sentences, 1M source language monolingual sentences, and 1M parallel sentences). BERT-small pre-trained from scratch on only the target language, using 5M target language sentences to ensure the training data amount was the same. Compared with the original model, the TRI-* model only has an extra TRI-layer added and some changes in the embedding layer. BERT-base-chinese and m-BERT-base models were downloaded from the official repository, which trained with 25M sentence (much more than our 5M sentences) and more training steps.

[2] The performance of `BERT-base` for pre-training from scratch with this limited data is inferior to that of `BERT-small`, so we do not compare it with our migrated models.

A.6 Downstream Tasks

Following previous contextualized language model pre-training, we evaluated the English-to-Chinese migrated language models on the CLUE benchmark. The Chinese Language Understanding Evaluation (CLUE) benchmark (Xu et al., 2020) consists of six different natural language understanding tasks: Ant Financial Question Matching (AFQMC), TouTiao Text Classification for News Titles (TNEWS), IFLYTEK (CO, 2019), Chinese-translated Multi-Genre Natural Language Inference (CMNLI), Chinese Winograd Schema Challenge (WSC), and Chinese Scientific Literature (CSL) and three machine reading comprehension tasks: Chinese Machine Reading Comprehension (CMRC) 2018 (Cui et al., 2019), Chinese IDiom cloze test (CHID) Zheng et al. (2019), and Chinese multiple-Choice machine reading Comprehension (C^3) (Sun et al., 2019). We built baselines for the natural language understanding tasks by adding a linear classifier on top of the "`[CLS]`" token to predict label probabilities. For the extractive question answering task, CMRC, we packed the question and passage tokens together with special tokens to form the input: "`[CLS]` $Question$ `[SEP]` $Passage$ `[SEP]`", and employed two linear output layers to predict the probability of each token being the start and end positions of the answer span following the practice for BERT (Devlin et al., 2019). Finally, in the multi-choice reading comprehension tasks, CHILD and C^3, we concatenated the passage, question, and each candidate answer ("`[CLS]` $Question \,||\, Answer$ `[SEP]` $Passage$ `[SEP]`"), input this to the models, and also predicted the probability of each answer on the representations from the "`[CLS]`" token following prior works (Yang et al., 2019; Liu et al., 2019b).

In addition to these language understanding tasks, language structure analysis tasks are also a very important part of natural language processing. Therefore, we also evaluated the PrLMs on syntactic dependency parsing and semantic role labeling, a type of semantic parsing. The baselines we selected for dependency parsing and semantic role labeling are from (Dozat & Manning, 2016) and Cai et al. (2018), respectively. These two baseline models are very strong and efficient and rely only on pure model structures to obtain advanced parsing performance. Our approach to integrate the PrLM with the two baselines is to replace the BiLSTM encoder in the baseline with the encoder of the PrLM. We took the first subword or character representation of a word as the representation of a word, which solved the PrLM's inconsistent granularity issue that impeded parsing.

For the English-to-Indonesian migrated language models, since the language understanding tasks in Indonesian are very limited, we chose to use the Universal Dependency (UD) parsing task (v2.3, Zeman et al., 2018), in which the treebanks of the world's languages were built by an international cooperative project, as the downstream task for evaluation.

A.7 Ablations

Effects of Different Embedding Initialization To show the effectiveness of non-contextualized simulation and adversarial embedding space alignment, we compare the TRI-RoBERTa-base models obtained in the commonality training phase of our framework under four different embedding initialization configurations: *random*, *random+adversarial align*, *fastText pre-trained*, and *fastText pre-trained+adversarial align*. In addition, to lessen the influence of training different amounts during different initializations, we trained an additional 40K update steps in the commonality training phase. We selected newstest2020-enzh.ref.zh in the WMT-20 news translation task as the evaluation set with a total of 1418 sentences to avoid potential overlapping with the training set. The subword-level bits-per-word (BPW) was used as the evaluation metric for the model's MLM performance[3].

The BPW results on the evaluation set are presented in Table 5. The non-contextualized fastText embedding simulation and adversarial embedding alignment setting achieves better BPW scores than other configurations, which shows the effectiveness of our proposed approach. In addition, comparing the embedding initialization of *random+adversarial align* and *fastText pre-trained* shows that pre-training non-contextualized embeddings using language data is more effective than direct embedding space alignment. Considering different training 20K steps versus 40K steps, longer training leads to lower BPW, but the performance gains are less than what our method brings.

[3]We do this because these models in comparison use the same vocabulary, and the masked parts on the evaluation set are identical, making the BPW scores comparable.

Configurations	20K	40K
Rand	6.9877	6.3264
Rand + Adv Align	6.7725	5.9982
fastText PT	6.5288	5.7957
fastText PT + Adv Align	5.9330	5.2679

Table 5: Evaluation of the subword-level BPW performance for the MLM objective of TRI-RoBERTa-base on various embedding initialization configurations after the commonality training.

Models	EN→ZH	ZH→EN
Transformer-base NMT	65.0 / 42.1 / 28.2 / 19.8	57.3 / 28.7 / 16.6 / 10.3
TRI-RoBERTa-base	46.6 / 5.7 / 0.8 / 0.1	36.4 / 5.2 / 0.7 / 0.1
w/o CdLM	1.3 / 0.1 / 0.0 / 0.0	1.7 / 0.1 / 0.0 / 0.0
TRI-RoBERTa-large	47.2 / 5.9 / 0.9 / 0.1	36.8 / 5.5 / 0.7 / 0.1
w/o CdLM	1.5 / 0.1 / 0.0 / 0.0	2.0 / 0.1 / 0.0 / 0.0

Table 6: Evaluation of the translation performance of our migrated language models on the WMT *newstest2020* test set with BLEU-1/2/3/4 metrics.

Effects of Cross-lingual Transfer Learning in TRELM We conduct further ablation studies to analyze our proposed TRELM framework's cross-lingual transfer learning design choices, including introducing the novel training objective, CdLM, and the TRILayer structure. The translation performance evaluation results are shown in Table 6. Using the newstest2020 en-zh and zh-en test sets, we evaluate the TRI-RoBERTa-base and TRI-RoBERTa-large models at the end of their transfer training phases. Since there is no alignment information available during the evaluation phase, we use the same successive alignment that MLM uses. For the sequence generated by the model, continuous repetitions were removed and the `[SEP]` token was taken as the stop mark to obtain the final translation sequence. In the EN→ZH translation direction, we report character-level BLEU, while in ZH→EN, we report word-level BLEU. The Transformer-base NMT models for comparison are from Tiedemann & Thottingal (2020) and were trained on the OPUS corpora (Tiedemann, 2012).

As seen from the results, our TRI-RoBERTa-base and TRI-RoBERTa-large with CdLM were able to obtain very good BLEU-1 scores, indicating that the mapping between the transferring source language and target language was explicitly captured by the model. When CdLM is removed and we only use the traditional joint MLM and TLM for training on the same size parallel data, we find that the BLEU-1 score significantly decreases, demonstrating that joint MLM and TLM do not learn explicit alignment information. The BLEU-1 score is lower than that of the Transformer-base NMT model, but this is because the Transformer-base model uses more parallel corpora as well as a more complex model design compared to our non-autoregressive translation pattern and lightweight TRILayer structure. In addition, compared with BLEU-2/3/4, it can be seen that although Transformer-base can accurately translate some tokens, many tokens are not translated or are translated in the wrong order due to the lack of word ordering information and the differing sequence lengths, which result in a very low score. This also shows that word order is a very important factor in translation.

Since the TRELM framework is evaluated using the existed pre-trained models, our migrated models are always larger than the original ones. Additional parameters arise in two places: embedding layer parameters grow due to a larger vocabulary and language embeddings, and the TRILayer structure

Models	Params	BPW	UAS	LAS
TRI-RoBERTa-base	154M	1.548	90.30	87.82
w/o CdLM	154M	3.028	88.16	85.20
w/o CdLM & w/o TRILayer	148M	3.469	87.45	84.69

Table 7: Language modeling effects of the CdLM objective and TRILayer structure for the Chinese TRI-RoBERTa-base model. UAS and LAS scores are given for the CTB 5.1 test set.

TRI-RoBERTa-base	BPW	UAS	LAS
w/ CdLM	1.548	90.30	87.82
w/ CdLM*	–	90.02	87.49
w/ TLM	3.028	88.16	85.20
w/ BRLM	2.932	89.85	87.27

Table 8: Effects of different cross-lingual transfer learning objectives. * indicates that a separate vocabulary is used.

adds parameters. The embedding layer growth is necessary, but the TRILayer structure is optional, as it is only used for cross-lingual transfer training. Therefore, for this ablation, we test removing the TRILayer structure for a fairer comparison[4] and show the results in Table 7. Comparing the evaluation set BPW scores of the final models obtained from RoBERTa-base under different migration methods, we found that our TRELM framework is stronger in cross-lingual transfer learning compared to jointly using MLM and TLM, and it does not simply rely on the extra parameters of the TRILayer. Furthermore, applying these pre-trained language models to the downstream task, dependency parsing on the CTB 5.1 treebank, achieves corresponding effects in BPW, which shows that the BPW score does describe the performance of PrLMs and that the pre-training performance will greatly affect performance in downstream tasks.

Comparison of Different Cross-lingual Transfer Learning Objectives As discussed in Appendix A.4, CdLM, TLM, and TLM variants such as BRLM are typical objectives of cross-lingual transfer learning, in which parallel sentences are utilized for cross-lingual optimization. In order to compare the differences between these objectives empirically, we conducted a comparative experiment on TRI-RoBERTa-base. For this experiment, instead of using the transfer learning objective CdLM in the second stage of training like our other models, we use TLM or BRLM instead. In addition, we follow (Artetxe et al., 2020) in experimenting with the effects of joint vocabulary versus a separate vocabulary in cross-lingual transfer learning, and we include a model, CdLM*, with a separate vocabulary in this comparison as well. Specifically, for this model, we forego language embeddings and adopt independent token embeddings for difference languages. CdLM and MLM alternately optimize the model.

The empirical comparison of these objectives is listed in Table 8. The migration target language is Chinese, and BPW score is used to compare the performance of the migrated model. We also show the dependency parsing performance on the CTB 5.1 dataset for the obtained model. Looking at CdLM and CdLM*, in our TRELM framework, using a joint vocabulary leads to better performance than using a separate vocabulary strategy, which is not consistent with Artetxe et al. (2020) 's conclusion. We attribute this difference to the fact that (Artetxe et al., 2020)'s model uses joint MLM pre-training of multiple languages to achieve implicit transfer learning, so maintaining independent embeddings is important for distinguishing the language. In TRELM, because it trains two half-models, the explicit conversion signal guides the model's migration training in discerning the language. When using separate vocabularies, some common information (such as punctuation, loanwords, etc.) are ignored, lessening the impact of CdLM. Second, comparing TLM, BRLM, and CdLM, we note that CdLM takes the source and target language sequences as input and output, respectively, which cooperates with the TRILayer and half-model training strategy much better, whereas TL and BRLM combine the source and target sentences as input and predict a masked sentence as in MLM, which is much less conducive to the half-model training strategy. Because the source and target language sentences are separate in CdLM, the model is much more able to differentiate the two languages, which makes CdLM a stronger cross-lingual transfer learning objective.

Comparison with Cross-lingual Transfer Learning Related Works on mPrLM Although we propose our method as an alternative to mPrLMs for cross-lingual transferring, it can also be applied to transfer the learning of mPrLMs. When transferring mPrLMs, the vocabulary replacement and embedding re-initialization are no longer needed, which makes our framework more simple.

[4]In this setting, we train the model with same number of update steps using joint MLM and TLM when leveraging parallel sentences.

Models	UAS	LAS
m-BERT-base	89.84	87.33
+Target-Language Finetune	90.28	87.76
+ROSITAWORD (Mulcaire et al., 2019)	89.88	87.36
+MIM (Liu et al., 2019a)	90.09	87.42
+Word-Alignment Finetune (Cao et al., 2020)	90.33	87.79
TRI-BERT-base	89.96	87.43
TRI-BERT-base (400K)	90.85	88.39
TRI-m-BERT-base	90.68	88.24
TRI-m-BERT-base (400K)	90.72	88.29

Table 9: Performance of different cross-lingual transfer learning approaches on dependency parsing on CTB 5.1.

We examine four main related approaches in the line of cross-lingual transfer learning based on PrLMs. The first approach is trivial: using data from the target language and MLM to finetune a mPrLM. This helps specify the mPrLM as a PrLM specifically for the target language.

The second is ROSITAWORD (Mulcaire et al., 2019). In this method, the contextualized embeddings of mPrLM are concatenated with non-contextualized multilingual word embeddings. This representation is then aligned across languages in a supervisory manner using a parallel corpus, biasing the model toward cross-lingual feature sharing. The third, proposed by Liu et al. (2019a), makes use of MIM (Meeting-In-the-Middle) (Doval et al., 2018), which uses a linear mapping to refine the embedding alignment, and is somewhat similar to our first step's adversarial embedding alignment, but because (Liu et al., 2019a) only migrate the contextualized embedding of an mPrLM, it is not a true migration of the model. Specifically, their post-processing trained linear mapping after the contextualized embedding of mPrLM is completely different from our new initialization of the raw embedding of PrLM. The fourth approach, Word-alignment Finetune, is similar in motivation to our CdLM, which uses the alignment information of the parallel corpora to perform finetuning training on the model (whereas ROSITAWORD and MIM focus on language-specific post-processing on the contextualized embedding of mPrLM). The difference is that Word-Alignment Finetune uses contextualized embedding similarity measurement for alignment to calculate the loss, and our method is inspired by machine translation, which uses language-to-language sequence translation for cross-lingual language modeling.

We evaluate the effectiveness of these methods on dependency parsing as shown in Table 9. We chose the widely used m-BERT-base as the base mPrLM and Chinese as the target language for these experiments. The resulting models were evaluated on the CTB 5.1 data of the dependency parsing task. For ROSITAWORD, we used the word-level embedding trained by Fastext and aligned by MUSE, as done in the original paper. For MIM, the number of training steps for the linear mapping is kept the same as in our first stage's adversarial embedding alignment training, and both train for 5 epochs. Target-Language Finetuning and Word-Alignment Finetuning use the same data as our main experiments and the same 120K update as well. We also listed a model migrated from a monolingual PrLM (TRI-BERT) to compare the performance differences between transfer learning from monolinguals and multilingual PrLMs. Since the migrated mPrLM is simpler - it does not need to re-initialize or train embeddings and can converge faster, we train the migrated PrLM model longer steps (400K total training steps) to more fairly compare them.

Comparing our TRELM with similar methods, the concatenation of cross-lingual aligned word-level embeddings in ROSITAWORD seems to have limited effect. MIM, which uses mapping for post-processing, leads to some improvement, but compared to Target-Language Finetune and Word-Alignment Finetune, it is obviously a weaker option. The results of TRI-m-BERT-base, Word-Alignment Finetune, and Target-Languagde Finetune suggest that using explicit alignment signals is advantageous compared to using the target language monolingual data when finetuning a limited amount of update steps, though when data is sufficient and training time is long enough, the performance for cross-lingually transferred models will approach the performance of monolingually pre-trained models regardless of transfer method. Thus, the methods primarily differ in how they perform with limited data, computing resources, or time. Our TRI-m-BERT-base outperforms +Word-Alignment Finetune, which shows that our CdLM, a language sequence modeling method

Models	DE GSD		ID GSD		JA GSD	
	UAS	LAS	UAS	LAS	UAS	LAS
(Dozat & Manning, 2016)	86.84*	81.31*	85.93*	78.21*	86.24*	84.52*
BERT-base	89.28	84.60	86.69	77.97	93.85‡	91.62‡
					94.66§	92.65§
m-BERT-base	88.37	83.22	87.19	79.10	94.12	92.35
TRI-BERT-base	89.12	84.34	87.56	79.44	94.30	92.56
TRI-RoBERTa-base	89.79	84.95	88.42	79.95	94.98	93.01

Table 10: Universal Dependency v2.3 parsing performance. * means that the results are evaluated based on our own implementation, not reported by Dozat & Manning (2016). We use the following PrLMs not provided by the official (Devlin et al., 2019), third-party BERT-base PrLMs: Deepset BERT-base-german, IndoBERT-base (Wilie et al., 2020), CL-TOHOKU BERT-base-japanese (‡), and NICT BERT-base-japanese (§).

inspired by machine translation, is more effective than solely deriving loss from an embedding space alignment. The results of TRI-BERT-base and TRI-m-BERT-base demonstrate that the simpler migration for m-BERT-base provides an initial performance boost when both models are trained 120k steps due to its faster convergence, but when they are trained to a longer 400K steps, TRI-BERT-base actually shows better performance compared to TRI-m-BERT-base.

More Languages for a More Comprehensive Evaluation In order to demonstrate the generalization ability of the cross-lingual transfer learning of the proposed TRELM framework, we also migrate to German (DE) and Japanese (JA) in addition to Chinese and Indonesian. We also experimented with these languages on the Universal dependency parsing task.

The migrated German and Japanese TRI-BERT-base and TRI-RoBERTa-base use the same corpus size and training steps as their respective Chinese and Indonesian models. We show the results of German, Indonesian, and Japanese on UD in Table 10. Since there are no official BERT-base models for these three language, we use third-party pre-trained models: Deepset BERT-base-german[5], IndoBERT-base (Wilie et al., 2020), CL-TOHOKU BERT-base-japanese[6], and NICT BERT-base-japanese[7].

First, according to the results in the table, our TRI-BERT-base achieves quite similar performance compared to the third-party BERT-base models and even exceeds the third-party models in some instances. This demonstrates that our TRELM is a general cross-lingual transfer learning framework. Second, comparing third-party pre-trained BERT-base models and the official m-BERT-base, we found that some third-party BERTs are even less effective than m-BERT (Generally speaking, m-BERT is not as good as monolingual BERT when the data and training time are sufficient). This shows that in some scenarios, pre-training from scratch is not a very good choice, potentially due to insufficient data, unsatisfactory pre-training resource quality, and/or insufficient pre-training time. Compared with the well-trained monolingual BERT models, our migrated models are very competitive and can exceed PrLMs suffering from poor pre-training. In addition, in DE and JA, we also observed that the effect of TRI-RoBERTa was stronger than that of the TRI-BERT, indicating that our migration process maintained the performance advantage of the original model.

[5] `https://deepset.ai/german-bert`

[6] `https://github.com/cl-tohoku/bert-japanese`

[7] `https://alaginrc.nict.go.jp/nict-bert/index.html`