Multilingual BERT Post-Pretraining Alignment

Lin Pan†, Chung-Wei Hang†, Haode Qi†, Abhishek Shah†, Saloni Potdar†, Mo Yu‡
†IBM Watson
‡MIT-IBM Watson AI Lab
{panl,hangc}@us.ibm.com, {Haode.Qi, Abhishek.Shah1}@ibm.com, {potdars, yum}@us.ibm.com

Abstract

We propose a simple method to align multilingual contextual embeddings as a post-pretraining step for improved cross-lingual transferability of the pretrained language models. Using parallel data, our method aligns embeddings on the word level through the recently proposed Translation Language Modeling objective as well as on the sentence level via contrastive learning and random input shuffling. We also perform sentence-level code-switching with English when finetuning on downstream tasks. On XNLI, our best model (initialized from mBERT) improves over mBERT by 4.7% in the zero-shot setting and achieves comparable result to XLM for translate-train while using less than 18% of the same parallel data and 31% fewer model parameters. On MLQA, our model outperforms XLM-R$_{\text{Base}}$, which has 57% more parameters than ours.

1 Introduction

Building on the success of monolingual pretrained language models (LM) such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), their multilingual counterparts mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) are trained using the same objectives—**Masked Language Modeling (MLM)** and in the case of mBERT, Next Sentence Prediction (NSP). MLM is applied to monolingual text that covers over 100 languages. Despite the absence of parallel data and explicit alignment signals, these models transfer surprisingly well from high resource languages, such as English, to other languages. On the Natural Language Inference (NLI) task XNLI (Conneau et al., 2018), a text classification model trained on English training data can be directly applied to the other 14 languages and achieve respectable performance. Having a single model that can serve over 100 languages also has important business applications.

Recent work improves upon these pretrained models by adding cross-lingual tasks leveraging parallel data that always involve English. Conneau and Lample (2019) pretrain a new Transformer-based (Vaswani et al., 2017) model from scratch with an MLM objective on monolingual data, and a Translation Language Modeling (TLM) objective on parallel data. Cao et al. (2020) align mBERT embeddings in a post-hoc manner: They first apply a statistical toolkit, FastAlign (Dyer et al., 2013), to create word alignments on parallel sentences. Then, mBERT is tuned via minimizing the mean squared error between the embeddings of English words and those of the corresponding words in other languages. Such post-hoc approach suffers from the limitations of word-alignment toolkits: (1) the noises from FastAlign can lead to error propagation to the rest of the pipeline; (2) FastAlign mainly creates the alignments with word-level translation and usually overlooks the contextual semantic compositions. As a result, the tuned mBERT is biased to shallow cross-lingual correspondence. Importantly, both approaches only involve word-level alignment tasks.

In this work, we focus on self-supervised, alignment-oriented training tasks using minimum parallel data to improve mBERT's cross-lingual transferability. We propose a **Post-Pretraining Alignment (PPA)** method consisting of both word-level and sentence-level alignment, as well as a finetuning technique on downstream tasks that take pairs of text as input, such as NLI and Question Answering (QA). Specifically, we use a slightly different version of TLM as our word-level alignment task and contrastive learning (Hadsell et al., 2006) on mBERT's [CLS] tokens to align sentence-level representations. Both tasks are self-supervised and do not require pre-alignment tools such as FastAlign. Our sentence-level alignment is implemented using MoCo (He et al., 2020), an instance discrimination-based method of contrastive learn-

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 210–219
June 6–11, 2021. ©2021 Association for Computational Linguistics

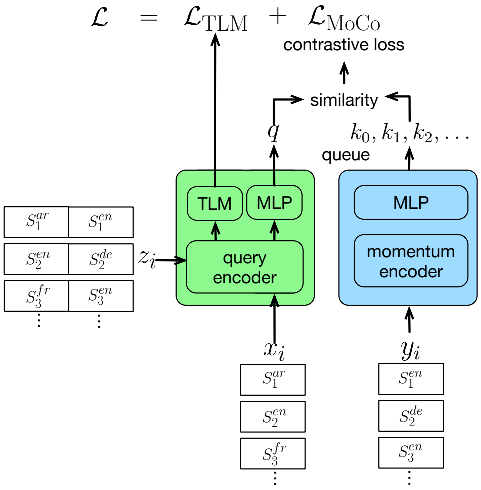

Figure 1: Model structure for our Post-Pretraining Alignment method using parallel data. We use MoCo to implement our sentence-level objective and TLM for our word-level objective. The model is trained in a multi-task manner with both objectives.

ing that was recently proposed for self-supervised representation learning in computer vision. Lastly, when finetuning on NLI and QA tasks for non-English languages, we perform sentence-level code-switching with English as a form of both alignment and data augmentation. We conduct controlled experiments on XNLI and MLQA (Lewis et al., 2020), leveraging varying amounts of parallel data during alignment. We then conduct an ablation study that shows the effectiveness of our method. On XNLI, our aligned mBERT improves over the original mBERT by 4.7% for zero-shot transfer, and outperforms Cao et al. (2020) while using the same amount of parallel data from the same source. For translate-train, where translation of English training data is available in the target language, our model achieves comparable performance to XLM while using far fewer resources. On MLQA, we get 2.3% improvement over mBERT and outperform XLM-R$_{\text{Base}}$ for zero-shot transfer.

2 Method

This section introduces our proposed Post-Pretraining Alignment (PPA) method. We first describe the MoCo contrastive learning framework and how we use it for sentence-level alignment. Next, we describe the finer-grained word-level alignment with TLM. Finally, when training data in the target language is available, we incorporate sentence-level code-switching as a form of both alignment and data augmentation to complement PPA. Figure 1 shows our overall model structure.

Background: Contrastive Learning Instance discrimination-based contrastive learning aims to bring two views of the same source image closer to each other in the representation space while encouraging views of different source images to be dissimilar through a contrastive loss. Recent advances in this area, such as SimCLR (Chen et al., 2020) and MoCo (He et al., 2020) have bridged the gap in performance between self-supervised representation learning and fully-supervised methods on the ImageNet (Deng et al., 2009) dataset. As a key feature for both methods, a large number of negative examples per instance are necessary for the models to learn such good representations. SimCLR uses in-batch negative example sampling, thus requiring a large batch size, whereas MoCo stores negative examples in a queue and casts the contrastive learning task as dictionary (query-key) lookup. In what follows, we first describe MoCo and then how we use it for sentence-level alignment.

Concretely, MoCo employs a dual-encoder architecture. Given two views v_1 and v_2 of the same image, v_1 is encoded by the query encoder f_q and v_2 by the momentum encoder f_k. v_1 and v_2 form a positive pair. Negative examples are views of different source images, and are stored in a queue $\in K$, which is randomly initialized. K is usually a large number (e.g., $K = 65,536$ for ImageNet). Negative pairs are formed by comparing v_1 with each item in the queue. Similarity between pairs is measured by dot product. MoCo uses the InfoNCE loss (van den Oord et al., 2019) to bring positive pairs closer to each other and push negative pairs apart. After a batch of view pairs are processed, those encoded by the momentum encoder are added to the queue as negative examples for future queries. During training, the query encoder is updated by the optimizer while the momentum encoder is updated by the exponential moving average of the query encoder's parameters to maintain queue consistency:

$$\theta_k = m\theta_k + (1 - m)\theta_q \qquad (1)$$

where θ_q and θ_k are model parameters of f_q and f_k, respectively. m is the momentum coefficient.

2.1 Sentence-Level Alignment Objective

Our sentence-level alignment falls under the general problem of bringing two views of inputs from the same source closer in the representation space while keeping those from different sources dissimilar through a contrastive loss. From a cross-lingual alignment perspective, we treat an English sequence S_i^{en} and its translation S_i^{tr} in another language $tr \in L$ as two manifestations of the same semantics. At the same time, sentences that are not translations of each other should be further apart in the representation space. Given parallel corpora consisting of $\{(S_1^{en}, S_1^{tr}), \ldots, (S_N^{en}, S_N^{tr})\}$, we align sentence representations in all the different languages together using MoCo.

We use the pretrained mBERT model to initialize both the query and momentum encoders. mBERT is made of 12 Transformer blocks, 12 attention heads, and hidden size $d_h = 768$. For input, instead of feeding the query encoder with English examples and the momentum encoder with translation examples or vice versa, we propose a **random input shuffling** approach. Specifically, we randomly shuffle the order of S_i^{en} and S_i^{tr} when feeding the two encoders, so that the query encoder sees both English and translation examples. We observe that this is a crucial step towards learning good multilingual representations using our method. The final hidden state $h \in \mathbb{R}^{1 \times d_h}$ of the `[CLS]` token, normalized with L_2 norm, is treated as the sentence representation [1]. Following Chen et al. (2020), we add a non-linear projection layer on top of h:

$$z = W_2 ReLU(W_1 h), \tag{2}$$

where $W_1 \in \mathbb{R}^{d_h \times d_h}$, $W_2 \in \mathbb{R}^{d_k \times d_h}$, and d_k is set to 300. The model is trained using the InfoNCE loss:

$$\mathcal{L}_{\text{MoCo}} = -\log \frac{\exp(z_q \cdot z_{k+}/\tau)}{\sum_{k=1}^{K} \exp(z_q \cdot z_k/\tau)}, \tag{3}$$

where τ is a temperature parameter. In our implementation, we use a relatively small batch size of 128, resulting in more frequent parameter updates than if a large batch size were used. Items enqueued early on can thus become outdated with a large queue, so we scale down the queue size to $K = 32,000$ to prevent the queue from becoming stale.

2.2 Word-Level Alignment Objective

We use TLM for word-level alignment. TLM is an extension of MLM that operates on bilingual data—parallel sentences are concatenated and MLM is applied to the combined bilingual sequence. Different from Conneau and Lample (2019), we do not reset positional embeddings when forming the bilingual sequence, and we also do not use language embeddings. In addition, the order of S_i^{en} and S_i^{tr} during concatenation is determined by the random input shuffling from the sentence-level alignment step and we add a `[SEP]` token between S_i^{en} and S_i^{tr}.

We randomly mask 15% of the WordPiece tokens in each combined sequence. Masking is done by using a special `[MASK]` token 80% of the times, a random token in the vocabulary 10% of the times, and unchanged for the remaining 10%. TLM is performed using the query encoder of MoCo. Our final PPA model is trained in a multi-task manner with both sentence-level objective and TLM:

$$\mathcal{L} = \mathcal{L}_{\text{MoCo}} + \mathcal{L}_{\text{TLM}}, \tag{4}$$

2.3 Finetuning on Downstream Tasks

After an alignment model is trained with PPA, we extract the query encoder from MoCo and finetune it on downstream tasks for evaluation. We follow the standard way of finetuning BERT-like models for sequence classification and QA tasks: (1) on XNLI, we concatenate the premise with the hypothesis, and add a `[SEP]` token in between. A softmax classifier is added on top of the final hidden state of the `[CLS]` token; (2) on MLQA, we concatenate the question with the context, and add a `[SEP]` token in between. We add two linear layers on top of mBERT followed by softmax over the context tokens to predict answer start and end positions, respectively.

We conduct experiments in two settings: 1. *Zero-shot cross-lingual transfer*, where training data is available in English but not in target languages. 2. *Translate-train*, where the English training set is (machine) translated to all the target languages. For the latter setting, we perform data augmentation with code-switched inputs, when training on languages other than English. For example, a Spanish question q_{es} and context c_{es} pair can be augmented to *two* question-context pairs (q_{es}, c_{en}) and (q_{en}, c_{es}) with code-switching, resulting in 2x training data [2]. The same goes for XNLI with premises and hypotheses. The code-switching is always between English, and a target language. During training, we

[1] Alternatively, we also experimented with mean-pooling of the last layer's embeddings as the sentence representation, but it performed slightly worse than using the `[CLS]` token.

[2] The original question-context pair (q_{es}, c_{es}) is not used for training as it did not help improve model performance in our experiments.

ensure the two augmented pairs appear in the same batch.

3 Experimental Settings

3.1 Parallel Data for Post-Pretraining

Parallel Data All parallel data we use involve English as the source language. Specifically, we collect en-fr, en-es, en-de parallel pairs from Europarl, en-ar, en-zh from MultiUN (Ziemski et al., 2016), en-hi from IITB (Kunchukuttan et al., 2018), and en-bg from both Europarl and EUbookshop. All datasets were downloaded from the OPUS[3] website (Tiedemann, 2012). In our experiments, we vary the number of parallel sentence pairs for PPA. For each language, we take the first 250k, 600k, and 2M English-translation parallel sentence pairs except for those too short (where either sentence has less than 10 WordPiece tokens), or too long (where both sentences concatenated together have more than 128 WordPiece tokens). Table 1 shows the actual number of parallel pairs in each of our 250k, 600k, and 2M settings.

3.2 Evaluation Benchmarks

XNLI is an evaluation dataset for cross-lingual NLI that covers 15 languages. The dataset is human-translated from the development and test sets of the English MultiNLI dataset (Williams et al., 2018). Given a sentence pair of premise and hypothesis, the task is to classify their relationship as *entailment*, *contradiction*, and *neutral*. For zero-shot cross-lingual transfer, we train on the English MultiNLI training set, and apply the model to the test sets of the other languages. For translate-train, we train on translation data that come with the dataset [4].

MLQA is an evaluation dataset for QA that covers seven languages. The dataset is derived from a three step process. (1) Parallel sentence mining from Wikipedia of the languages. (2) English question annotation and answer span annotation on English context. (3) Professional translation of English questions to the other languages as well as answer span annotation. MLQA has two evaluation tasks: (a) Cross-lingual transfer (XLT), where the question and context are in the same language. (b) Generalized cross-lingual transfer (G-XLT), where the question and context are in different languages. We focus on XLT in this work. For zero-shot cross-lingual transfer, we train on the English SQuAD v1.1 (Rajpurkar et al., 2016) training set. For translate-train, we train on translation data provided in Hu et al. (2020) [5]

3.3 Training Details

For both PPA and finetuning on downstream tasks, we use the AdamW optimizer with 0.01 weight decay and a linear learning rate scheduler. For PPA, we use a batch size of 128, mBERT max sequence length 128 and learning rate warmup for the first 10% of the total iterations, peaking at 0.00003. The MoCo momentum is set to 0.999, queue size 32000 and temperature 0.05. Our PPA models are trained for 10 epochs, except for the 2M setting where 5 epochs are trained. On XNLI, we use a batch size of 32, mBERT max sequence length 128 and finetune the PPA model for 2 epochs. Learning rate peaks at 0.00005 and warmup is done to the first 1000 iterations. On MLQA, mBERT max sequence length is set to 386 and peak learning rate 0.00003. The other parameters are the same as XNLI. Our experiments are run on a single 32 GB V100 GPU, except for PPA training that involves either MLM or TLM, where two such GPUs are used. We also use mixed-precision training to save on GPU memory and speed up experiments.

4 Results

We report results on the test set of XNLI and MLQA and we do hyperparameter searching on the development set. All the experiments for translate-train were done using the code-switching technique introduced in Section 2.

XNLI Table 2 shows results on XNLI measured by accuracy. Devlin et al. (2019) only provide results on a few languages[6], so we use the mBERT results from Hu et al. (2020) as our baseline for zero-shot cross-lingual transfer, and Wu and Dredze (2019) for translate-train. Our best model, trained with 2M parallel sentences per language improves over mBERT baseline by 4.7% for zero-shot transfer, and 3.2% for translate-train.

Compared to Cao et al. (2020), which use 250k parallel sentences per language from the same sources as we do for post-pretraining alignment,

[3] http://opus.nlpl.eu/

[4] https://cims.nyu.edu/~sbowman/xnli/

[5] https://github.com/google-research/xtreme

[6] https://github.com/google-research/bert/blob/master/multilingual.md

Resource	**fr**	**es**	**de**	**bg**	**ar**	**zh**	**hi**	**total**
Original data								
MultiUN	14.2M	12.2M	-	-	10.6M	10.5M	-	
Europarl	2.1M	2.0M	2.0M	0.4M	-	-	-	
EUbookshop	-	-	9.6M	0.2M	-	-	-	
IITB	-	-	-	-	-	-	1.6M	
Considered in this paper								
MultiUN	-	-	-	-	10.6M	10.5M	-	
Europarl	2.1M	2.0M	2.0M	0.4M	-	-	-	
EUbookshop	-	-	-	0.2M	-	-	-	
IITB	-	-	-	-	-	-	1.6M	
Total	2.1M	2.0M	2.0M	0.6M	10.6M	10.5M	1.6M	
Used for our post-pretraining alignment (PPA)								
Ours (250k)	250k	250k	250k	250k	250k	250k	250k	1.8M
Ours (600k)	600k	600k	600k	467k	600k	600k	600k	4.1M
Ours (2M)	1.8M	1.7M	1.7M	467k	2.0M	2.0M	0.8M	10.5M
Used by other approaches								
Cao et al. (2020)[a]	250k	250k	250k	250k	250k	250k	250k	1.8M
Artetxe and Schwenk (2019)[b]	-	-	-	-	-	-	-	223M
XLM (Conneau and Lample, 2019)[c]	14.2M	12.2M	9.6M	0.2M	10.6M	10.5M	1.6M	58.9M

Table 1: Parallel data statistics. All parallel data involve English as source language. We use Europarl for en-fr, en-es, and en-de, both Europarl and EUbookshop for en-bg, MultiUN for en-ar, en-zh, and IITB for en-hi. Our 250k setting uses an equal amount of data from the same source as Cao et al. (2020). Our 2M setting uses approximately 63% and 17.8% of the parallel data Artetxe and Schwenk (2019) and Conneau and Lample (2019) use, respectively.

[a] Cao et al. (2020) uses the same 250k parallel corpora as our 250k setting, thus giving an apple-to-apple comparison.
[b] Artetxe and Schwenk (2019)'s number includes a total of 93 languages.
[c] We only list the number of parallel sentences XLM uses for the languages we consider.

our 250k model does better for all languages considered and we do not rely on the word-to-word pre-alignment step using FastAlign, which is prone to error propagation to the rest of the pipeline.

Compared to XLM, our 250k, 600k and 2M settings represent 3.1%, 7% and 17.8% of the parallel data used by XLM, respectively (see Table 1). The XLM model also has 45% more parameters than ours as Table 3 shows. Furthermore, XLM trained with MLM only is already significantly better than mBERT even though the source of its training data is the same as mBERT from Wikipedia. One reason could be that XLM contains 45% more model parameters than mBERT as model depth and capacity are shown to be key to cross-lingual success (K et al., 2020). Additionally, Wu and Dredze (2019) hypothesize that limiting pretraining to the languages used by downstream tasks may be beneficial since XLM models are pretrained on the 15 XNLI languages only. Our 2M model bridges the gap between mBERT and XLM from 7.5% to 2.8% for zero-shot transfer. Note that, for bg, our total processed pool of en-bg data consists of 456k parallel sentences, so there is no difference in en-bg data between our 600k and 2M settings. For translate-train, our model achieves comparable performance to XLM with the further help of code-switching during finetuning.

Our alignment-oriented method is, to a large degree, upper-bounded by the English performance, since all our parallel data involve English and all the other languages are implicitly aligning with English through our PPA objectives. Our 2M model is able to improve the English performance to 82.4 from the mBERT baseline, but it is still lower than XLM (MLM), and much lower than XLM (MLM+TLM). We hypothesize that more high-quality monolingual data and model capacity are needed to further improve our English performance, thereby helping other languages better align with it.

MLQA Table 4 shows results on MLQA measured by F1 score. We notice the mBERT baseline from the original MLQA paper is significantly lower than that from Hu et al. (2020), so we use the latter as our baseline. Our 2M model outperforms the baseline by 2.3% for zero-shot and is also 0.2% better than XLM-R$_{\text{Base}}$, which uses 57% more model parameters than mBERT as Table 3 shows. For translate-train, our 250k model is 1.3% better than the baseline.

Comparing our model performance using vary-

Model	**en**	**fr**	**es**	**de**	**bg**	**ar**	**zh**	**hi**	**avg**
Zero-shot cross-lingual transfer									
mBERT (Devlin et al., 2019)	81.4	-	74.3	70.5	-	62.1	63.8	-	-
mBERT from (Hu et al., 2020)	80.8	73.4	73.5	70.0	68.0	64.3	67.8	58.9	69.6
Cao et al. (2020)	80.1	74.5	75.5	73.1	73.4	-	-	-	-
Artetxe and Schwenk (2019)	73.9	71.9	72.9	72.6	74.2	**71.4**	71.4	65.5	71.7
Ours (250k)	82.4	75.5	76.2	73.3	**74.6**	68.2	71.7	62.8	73.1
Ours (600k)	82.4	**76.7**	76.4	74.0	74.1	69.1	72.3	66.9	74.0
Ours (2M)	**82.8**	76.6	**76.7**	**74.2**	73.8	70.3	**72.8**	**66.9**	**74.3**
XLM (MLM)	83.2	76.5	76.3	74.2	74.0	68.5	71.9	65.7	73.8
XLM (MLM + TLM)	85.0	78.7	78.9	77.8	77.4	73.1	76.5	69.6	77.1
Translate-train									
mBERT (Devlin et al., 2019)	81.9	-	77.8	75.9	-	70.7	76.6	-	-
mBERT from (Wu and Dredze, 2019)	82.1	76.9	78.5	74.8	75.4	70.8	76.2	65.3	75.0
Ours (250k)	82.4	78.8	79.0	**78.7**	78.4	74.0	77.9	69.6	77.4
Ours (600k)	82.4	79.7	79.7	77.9	**79.0**	75.2	77.8	71.5	77.9
Ours (2M)	**82.8**	**79.7**	**80.6**	78.6	78.8	**75.2**	**78.0**	**72.0**	**78.2**
XLM (Conneau and Lample, 2019)	85.0	80.2	80.8	80.3	79.3	76.5	78.6	72.3	79.1

Table 2: XNLI accuracy scores for each language. After alignment, our best model improves over mBERT by 4.7% for zero-shot transfer, and achieves comparable performance to XLM for translate-train. Artetxe and Schwenk (2019) use 223M parallel sentences covering 93 languages. XLM uses 58.9M parallel sentences (for the seven languages we consider) with 40% more parameters. Our approach (250k, 600k, and 2M per language) uses a total of 1.8M, 4.1M, and 10.5M parallel sentences, respectively.

Model	**# langs**	**L**	**H_m**	**H_{ff}**	**A**	**V**	**# params**
mBERT	104	12	768	3072	12	110k	172M
XLM	15	12	1024	4096	8	95k	250M
XLM-R_{Base}	100	12	768	3072	12	250k	270M
Ours	104	12	768	3072	12	110k	172M

Table 3: Model architecture and sizes from Conneau et al. (2020). L is the number of Transformer layers, H_m is the hidden size, H_{ff} is the dimension of the feed-forward layer, A is the number of attention heads, and V is the vocabulary size.

ing amounts of parallel data, we observe that 600k per language is our sweet spot considering the trade-off between resource and performance. Going up to 2M helps on XNLI, but less significantly compared to the gain going from 250k to 600k. On MLQA, surprisingly, 250k slightly outperforms the other two for translate-train.

Ablation Table 5 shows the contribution of each component of our method on XNLI. Removing TLM (*-TLM*) consistently leads to about 1% accuracy drop across the board, showing positive effects of the word-alignment objective. To better understand TLM's consistent improvement, we replace TLM with MLM (*repl TLM w/ MLM*), where we treat S_i^{en} and S_i^{tr} from the parallel corpora as separate monolingual sequences and perform MLM on each of them. The masking scheme is the same as TLM described in Section 2. We observe that MLM does not bring significant improvement. This confirms that the improvement of TLM is not from the encoders being trained with more data and iterations. Instead, the word-alignment nature of TLM does help the multilingual training.

Comparing our model without word-level alignment, i.e., *-TLM*, to the baseline mBERT in Table 2, we get 2–4% improvement in the zero-shot setting and 1–2% improvement in translate-train as the amount of parallel data is increased. These are relatively large improvements considering the fact that only sentence-level alignment is used. This also conforms to our intuition that sentence-level alignment is a good fit here since XNLI is a sentence-level task.

In the zero-shot setting, removing MoCo (*-MoCo*) performs similarly to *-TLM*, where we observe an accuracy drop of about 1% compared to our full system. In translate-train, *-MoCo* outperforms *-TLM* and even matches the full system performance for 250k.

Finally, we show ablation result for our code-switching in translate-train. On average, code-switching provides an additional gain of 1%.

5 Related Work

Training Multilingual LMs with Shared Vocabulary mBERT (Devlin et al., 2019) is trained us-

Model	en	ar	de	es	hi	zh	avg
Zero-shot cross-lingual transfer							
mBERT from (Lewis et al., 2020)	77.7	45.7	57.9	64.3	43.8	57.5	57.8
mBERT from (Hu et al., 2020)	**80.2**	52.3	59.0	67.4	50.2	59.6	61.5
Ours (250k)	80.0	52.6	**63.2**	67.7	54.1	60.5	63.0
Ours (600k)	79.7	52.4	62.8	67.6	58.3	60.4	63.5
Ours (2M)	79.8	53.8	62.3	67.7	57.9	61.5	**63.8**
XLM from (Lewis et al., 2020)	74.9	54.8	62.2	**68.0**	48.8	61.1	61.6
XLM-R$_{\text{Base}}$ (Conneau et al., 2020)	77.1	**54.9**	60.9	67.4	**59.4**	**61.8**	63.6
Translate-train							
mBERT from (Lewis et al., 2020)	77.7	51.8	62.0	53.9	55.0	61.4	60.3
mBERT from (Hu et al., 2020)	**80.2**	55.0	64.6	70.0	60.1	63.9	65.6
Ours (250k)	80.0	58.0	**65.7**	**71.0**	62.0	**64.4**	**66.9**
Ours (600k)	79.7	58.1	65.2	70.5	**63.4**	64.1	66.8
Ours (2M)	79.8	**58.2**	64.7	70.6	63.1	64.4	66.8
XLM from (Lewis et al., 2020)	74.9	54.0	61.4	65.2	50.7	59.8	61.0

Table 4: MLQA F1 scores for each language. After alignment, our best model improves over mBERT baseline by 2.3% and outperforms XLM-R$_{\text{Base}}$ for zero-shot transfer. Our model trained with the smallest amount of parallel data is 1.3% better than mBERT baseline for translate-train.

ing MLM and NSP objectives on Wikipedia data in 104 languages with a shared vocabulary. Several works study what makes this pretrained model multilingual, and why it works well for cross-lingual transfer. Pires et al. (2020) hypothesize that having a shared vocabulary for all languages helps mapping tokens to a shared space. However, K et al. (2020) train several bilingual BERT models such as en-es, and *enfake*-es, where data for *enfake* is constructed by Unicode shifting of the English data such that there is no character overlap with data of the other language. Result shows that *enfake*-es still transfers well to Spanish and the contribution from shared vocabulary is very small. The authors point out that model depth and capacity instead are the key factors contributing to mBERT's cross-lingual transferability. XLM-R (Conneau et al., 2020) improves over mBERT by training longer with more data from CommonCrawl, and without the NSP objective. In terms of model size, XLM-R uses over 3x more parameters than mBERT. Its base version, XLM-R$_{\text{Base}}$, is more comparable to mBERT with the same hidden size and number of attention heads, but a larger shared vocabulary.

Training Multilingual LMs with Parallel Sentences In addition to MLM on monolingual data, XLM (Conneau and Lample, 2019) further improves their cross-lingual LM pretraining by introducing a new TLM objective on parallel data. TLM concatenates source and target sentences together, and predicts randomly masked tokens. Our work uses a slightly different version of TLM together with a contrastive objective to post-pretrain mBERT. Unlike XLM, our TLM does not reset positions of target sentences, and does not use language embeddings. We also randomly shuffle the order of source and target sentences. Another difference between XLM and our work is XLM has 45% more parameters and uses more training data. Similar to XLM, Unicoder (Huang et al., 2019) pretrains LMs on multilingual corpora. In addition to MLM and TLM, they introduce three additional cross-lingual pretraining tasks: word recover, paraphrase classification, and mask language model. Yang et al. (2020) propose Alternating Language Modeling (ALM). On a pair of bilingual sequences, instead of TLM, they perform phrase-level code-switching and MLM on the code-switched sequence. ALM is pretrained on both monolingual Wikipedia data and 1.5B code-switched sentences.

Training mBERT with Word Alignments Cao et al. (2020) post-align mBERT embeddings by first generating word alignments on parallel sentences that involve English. For each aligned word pair, the L_2 distance between their embeddings is minimized to train the model. In order to maintain original transferability to downstream tasks, a regularization term is added to prevent the target language embeddings from deviating too much from their mBERT initialization. Our approach post-aligns mBERT with two self-supervised signals from parallel data without using pre-alignment tools. Wang et al. (2019) also align mBERT em-

Model	en	fr	es	de	bg	ar	zh	hi	avg
Zero-shot cross-lingual transfer									
Our full system (250k)	**82.4**	**75.5**	**76.2**	**73.3**	**74.6**	**68.2**	**71.7**	62.8	**73.1**
- MoCo	82.2	75.3	75.8	73.0	71.3	67.1	71.3	61.8	72.2
- TLM	80.5	74.7	75.2	71.4	72.7	66.2	68.9	**64.0**	71.7
repl TLM w/ MLM	81.5	75.0	75.2	70.8	72.5	66.2	69.0	61.9	71.5
Our full system (600k)	**82.4**	**76.7**	**76.4**	**74.0**	**74.1**	**69.1**	**72.3**	**66.9**	**74.0**
- MoCo	82.0	75.5	75.9	72.8	72.1	68.5	72.1	64.5	72.9
- TLM	81.2	75.1	75.4	71.9	73.3	68.2	71.0	65.8	72.7
repl TLM w/ MLM	82.2	75.7	75.5	73.0	73.3	68.5	71.1	66.5	73.2
Our full system (2M)	**82.8**	**76.6**	**76.7**	**74.2**	**73.8**	**70.3**	**72.8**	**66.9**	**74.3**
- MoCo	82.5	75.2	76.3	72.4	71.9	67.9	71.4	65.2	72.9
- TLM	81.3	76.2	76.4	73.2	72.9	69.0	71.5	66.1	73.3
repl TLM w/ MLM	82.0	75.8	75.8	73.2	73.5	68.7	70.6	65.8	73.2
Translate-train									
Our full system (250k)	**82.4**	78.8	79.0	**78.7**	78.4	**74.0**	**77.9**	69.6	**77.4**
- MoCo	82.2	**79.8**	**79.8**	77.8	**78.9**	73.8	77.3	69.8	77.4
- TLM	80.5	78.3	77.8	77.5	77.4	72.4	77.2	69.5	76.3
repl TLM w/ MLM	81.5	78.4	79.4	78.3	78.2	73.4	76.9	**69.9**	77.0
- CS	82.4	77.8	79.5	76.2	76.2	73.2	77.5	67.9	76.3
Our full system (600k)	**82.4**	**79.7**	**79.7**	77.9	**79.0**	**75.2**	77.8	**71.5**	**77.9**
- MoCo	82.0	79.5	79.2	**78.1**	78.9	74.1	**78.1**	71.0	77.6
- TLM	81.2	78.5	78.6	78.1	77.7	73.7	76.6	70.8	76.9
repl TLM w/ MLM	82.2	78.4	78.4	77.1	78.0	73.9	76.9	70.8	77.0
- CS	82.4	79.2	78.3	77.5	77.0	73.6	77.3	69.9	76.9
Our full system (2M)	**82.8**	**79.7**	**80.6**	78.6	**78.8**	75.2	**78.0**	**72.0**	**78.2**
- MoCo	82.5	79.1	80.0	**79.1**	78.5	**75.3**	77.7	70.5	77.8
- TLM	81.3	78.9	79.4	78.0	77.8	74.4	77.2	70.0	77.1
repl TLM w/ MLM	82.0	79.1	79.0	78.2	77.8	74.3	77.7	70.4	77.3
- CS	82.8	79.1	79.0	78.0	77.5	73.6	77.1	69.5	77.1

Table 5: Ablation Study on XNLI. 250k, 600k, 2M refer to the maximum number of parallel sentence pairs per language used in PPA. *MoCo* refers to our sentence-level alignment task using contrastive learning. *TLM* refers to our word-level alignment task with translation language modeling. *CS* stands for code-switching. We conduct an additional study *repl TLM w/ MLM*, which means instead of TLM training, we augment our sentence-level alignment with regular MLM on monolingual text. This ablation confirms that the TLM objective helps because of its word alignment capability, not because we train the encoders with more data and iterations.

beddings using parallel data. They learn a linear transformation that maps a word embedding in a target language to the embedding of the aligned word in the source language. They show that their transformed embeddings are more effective on zero-shot cross-lingual dependency parsing.

Besides the aforementioned three major directions, Artetxe and Schwenk (2019) train a multilingual sentence encoder on 93 languages. Their stacked BiLSTM encoder is trained by first generating embedding of a source sentence and then decoding the embedding into the target sentence in other languages.

Concurrent to our work, Chi et al. (2020), Feng et al. (2020) and Wei et al. (2020) also leverage variants of contrastive learning for cross-lingual alignment. We focus on a smaller model and improve on it using as little parallel data as possible. We also explore code-switching during finetuning on downstream tasks to complement the post-pretraining alignment objectives.

6 Conclusion

Post-pretraining embedding alignment is an efficient means of improving cross-lingual transferability of pretrained multilingual LMs, especially when pretraining from scratch is not feasible. We showed that our self-supervised sentence-level and word-level alignment tasks can greatly improve mBERT's performance on downstream tasks of NLI and QA, and the method can potentially be applied to improve other pretrained multilingual LMs.

In addition to zero-shot cross-lingual transfer, we also showed that code-switching with English during finetuning provides additional alignment signals, when training data is available for the target language.

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics (TACL)*, 7:597–610.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *Proceedings of the 8th International Conference on Learning Representation (ICLR)*, Addis Ababa, Ethiopia.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, pages 1–18.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2020. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. *arXiv preprint arXiv:2007.07834*, pages 1–11.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8440–8451, Seattle.

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

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2475–2485, Brussels.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Processings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, Miami.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 20th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, Minneapolis. The Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 644–648, Atlanta. The Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT sentence embedding. *arXiv preprint arXiv:2007.01852*, pages 1–13.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1735–1742, New York.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, Seattle.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *arXiv preprint arXiv:2003.11080*, pages 1–20.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2485–2494, Hong Kong.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: An empirical study. In *Proceedings of the 8th International Conference on Learning Representation (ICLR)*, Addis Ababa, Ethiopia.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, pages 3473–3476, Miyazaki, Japan.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1–16, Seattle.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, pages 1–13.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2020. How multilingual is multilingual BERT? *arXiv preprint arXiv:1906.01502*, pages 1–6.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2383–2392, Austin, Texas.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, pages 2214–2218, Istanbul.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, pages 1–13.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008, Long Beach, CA.

Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu. 2019. Cross-lingual BERT transformation for zero-shot dependency parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5721–5727, Hong Kong.

Xiangpeng Wei, Yue Hu, Rongxiang Weng, Luxi Xing, Heng Yu, and Weihua Luo. 2020. On learning universal representations across languages. *arXiv preprint arXiv:2007.15960*, pages 1–13.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 19th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*, pages 1112–1122, New Orleans. The Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, Bentz, Becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong.

Jian Yang, Shuming Ma, Dongdong Zhang, ShuangZhi Wu, Zhoujun Li, and Ming Zhou. 2020. Alternating language modeling for cross-lingual pre-training. In *Proceedings of the 34th Conference on Artificial Intelligence (AAAI)*, pages 9386–9393, New York. AAAI Press.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, pages 3530–3534, Portorož, Slovenia.