XLM-E: Cross-lingual Language Model Pre-training via ELECTRA

Anonymous ACL submission

Abstract

In this paper, we introduce ELECTRA-style tasks (Clark et al., 2020b) to cross-lingual language model pre-training. Specifically, we present two pre-training tasks, namely multilingual replaced token detection, and translation replaced token detection. Besides, we pretrain the model, named as XLM-E, on both multilingual and parallel corpora. Our model outperforms the baseline models on various cross-lingual understanding tasks with much less computation cost. Moreover, analysis shows that XLM-E tends to obtain better cross-lingual transferability.

1 Introduction

It has become a de facto trend to use a pretrained language model (Devlin et al., 2019; Dong et al., 2019; Yang et al., 2019b; Bao et al., 2020) for downstream NLP tasks. These models are typically pretrained with masked language modeling objectives, which learn to generate the masked tokens of an input sentence. In addition to monolingual representations, the masked language modeling task is effective for learning cross-lingual representations. By only using multilingual corpora, such pretrained models perform well on zero-shot cross-lingual transfer (Devlin et al., 2019; Conneau et al., 2020), i.e., fine-tuning with English training data while directly applying the model to other target languages. The cross-lingual transferability can be further improved by introducing external pre-training tasks using parallel corpus, such as translation language modeling (Conneau and Lample, 2019), and cross-lingual contrast (Chi et al., 2021b). However, previous cross-lingual pre-training based on masked language modeling usually requires massive computation resources, rendering such models quite expensive. As shown in Figure 1, our proposed XLM-E achieves a huge speedup compared with well-tuned pretrained models.

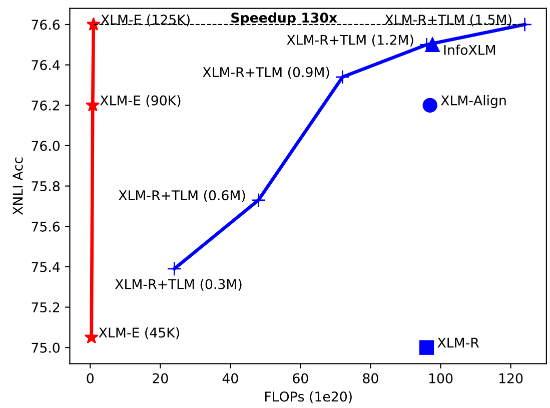


Figure 1: The proposed XLM-E pre-training (red line) achieves 130× speedup compared with an in-house pretrained XLM-R augmented with translation language modeling (XLM-R + TLM; blue line), using the same corpora and code base. The training steps are shown in the brackets. We also present XLM-R (Conneau et al., 2020), InfoXLM (Chi et al., 2021b), and XLM-Align (Chi et al., 2021c). The compared models are all in Base size.

In this paper, we introduce ELECTRA-style tasks (Clark et al., 2020b) to cross-lingual language model pre-training. Specifically, we present two discriminative pre-training tasks, namely multilingual replaced token detection, and translation replaced token detection. Rather than recovering masked tokens, the model learns to distinguish the replaced tokens in the corrupted input sequences. The two tasks build input sequences by replacing tokens in multilingual sentences, and translation pairs, respectively. We also describe the pre-training algorithm of our model, XLM-E, which is pretrained with the above two discriminative tasks. It provides a more compute-efficient and sample-efficient way for cross-lingual language model pre-training.

We conduct extensive experiments on the XTREME cross-lingual understanding benchmark to evaluate and analyze XLM-E. Over seven datasets, our model achieves competitive results with the baseline models, while only using 1% of the computation cost comparing to XLM-R. In addition to the high computational efficiency, our model also shows the cross-lingual transferability that achieves a reasonably low transfer gap. We also show that the discriminative pre-training encourages universal representations, making the text representations better aligned across different languages.

Our contributions are summarized as follows:

- We explore ELECTRA-style tasks for cross-lingual language model pre-training, and pretrain XLM-E with both multilingual corpus and parallel data.
- We demonstrate that XLM-E greatly reduces the computation cost of cross-lingual pre-training.
- We show that discriminative pre-training tends to encourage better cross-lingual transferability.

2 Background: ELECTRA

ELECTRA (Clark et al., 2020b) introduces the replaced token detection task for language model pre-training, with the goal of distinguishing real input tokens from corrupted tokens. That means the text encoders are pretrained as discriminators rather than generators, which is different from the previous pretrained language models, such as BERT (Devlin et al., 2019), that learn to predict the masked tokens.

ELECTRA trains two Transformer (Vaswani et al., 2017) encoders, serving as generator and discriminator, respectively. The generator G is typically a small BERT model trained with the masked language modeling (MLM; Devlin et al. 2019) task. Consider an input sentence $\boldsymbol{x} = \{x_i\}_{i=1}^{n}$ containing n tokens. MLM first randomly selects a subset $\mathcal{M} \subseteq \{1, \ldots, n\}$ as the positions to be masked, and construct the masked sentence $\boldsymbol{x}^{\text{masked}}$ by replacing tokens in $\mathcal{M}$ with `[MASK]`. Then, the generator predicts the probability distributions of the masked tokens $p_G(x|\boldsymbol{x}^{\text{masked}})$. The loss function of the generator G is:

$$\mathcal{L}_G(\boldsymbol{x}; \boldsymbol{\theta}_G) = -\sum_{i \in \mathcal{M}} \log p_G(x_i|\boldsymbol{x}^{\text{masked}}). \quad (1)$$

The discriminator D is trained with the replaced token detection task. Specifically, the discriminator takes the corrupted sentences $\boldsymbol{x}^{\text{corrupt}}$ as input, which is constructed by replacing the tokens in $\mathcal{M}$ with the tokens sampled from the generator G:

$$\begin{cases} x_i^{\text{corrupt}} \sim p_G(x_i|\boldsymbol{x}^{\text{masked}}), & i \in \mathcal{M} \\ x_i^{\text{corrupt}} = x_i, & i \notin \mathcal{M} \end{cases} \quad (2)$$

Then, the discriminator predicts whether x_i^{corrupt} is original or sampled from the generator. The loss function of the discriminator D is

$$\mathcal{L}_D(\boldsymbol{x}; \boldsymbol{\theta}_D) = -\sum_{i=1}^{n} \log p_D(z_i|\boldsymbol{x}^{\text{corrupt}}) \quad (3)$$

where z_i represents the label of whether x_i^{corrupt} is the original token or the replaced one. The final loss function of ELECTRA is the combined loss of the generator and discriminator losses, $\mathcal{L}_E = \mathcal{L}_G + \lambda \mathcal{L}_D$.

Compared to generative pre-training, ELECTRA uses more model parameters and training FLOPs per step, because it contains a generator and a discriminator during pre-training. However, only the discriminator is used for fine-tuning on downstream tasks, so the size of the final checkpoint is similar to BERT-like models in practice.

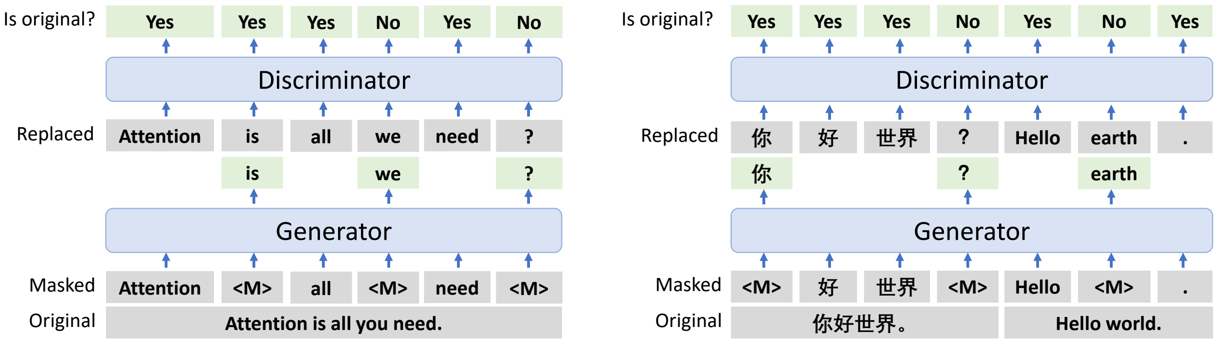
3 Methods

Figure 2 shows an overview of the two discriminative tasks used for pre-training XLM-E. Similar to ELECTRA described in Section 2, XLM-E has two Transformer components, i.e., generator and discriminator. The generator predicts the masked tokens given the masked sentence or translation pair, and the discriminator distinguishes whether the tokens are replaced by the generator.

3.1 Pre-training Tasks

The pre-training tasks of XLM-E are multilingual replaced token detection (MRTD), and translation replaced token detection (TRTD).

Multilingual Replaced Token Detection The multilingual replaced token detection task requires the model to distinguish real input tokens from corrupted multilingual sentences. Both the generator and the discriminator are shared across languages. The vocabulary is also shared for different languages. The task is the same as in monolingual ELECTRA pre-training (Section 2). The only



(a) Multilingual replaced token detection (MRTD)

(b) Translation replaced token detection (TRTD)

Figure 2: Overview of two pre-training tasks of XLM-E, i.e., multilingual replaced token detection, and translation replaced token detection. The generator predicts the masked tokens given a masked sentence or a masked translation pair, and the discriminator distinguishes whether the tokens are replaced by the generator.

difference is that the input texts can be in various languages.

We use uniform masking to produce the corrupted positions. We also tried span masking (Joshi et al., 2019; Bao et al., 2020) in our preliminary experiments. The results indicate that span masking significantly weakens the generator's prediction accuracy, which in turn harms pre-training.

Translation Replaced Token Detection Parallel corpora are easily accessible and proved to be effective for learning cross-lingual language models (Conneau and Lample, 2019; Chi et al., 2021b), while it is under-studied how to improve discriminative pre-training with parallel corpora. We introduce the translation replaced token detection task that aims to distinguish real input tokens from translation pairs. Given an input translation pair, the generator predicts the masked tokens in both languages. Consider an input translation pair $(\boldsymbol{e}, \boldsymbol{f})$. We construct the input sequence by concatenating the translation pair as a single sentence. The loss function of the generator G is:

$$
\begin{aligned}
\mathcal{L}_G(\boldsymbol{e}, \boldsymbol{f}; \boldsymbol{\theta}_G) = &- \sum_{i \in \mathcal{M}_e} \log p_G(e_i | [\boldsymbol{e}; \boldsymbol{f}]^{\text{masked}}) \\
&- \sum_{i \in \mathcal{M}_f} \log p_G(f_i | [\boldsymbol{e}; \boldsymbol{f}]^{\text{masked}})
\end{aligned}
$$

where $[;]$ is the operator of concatenation, and $\mathcal{M}_e, \mathcal{M}_f$ stand for the randomly selected masked positions for $\boldsymbol{e}$ and $\boldsymbol{f}$, respectively. This loss function is identical to the translation language modeling loss (TLM; Conneau and Lample 2019). The discriminator D learns to distinguish real input tokens from the corrupted translation pair. The corrupted translation pair $(\boldsymbol{e}^{\text{corrupt}}, \boldsymbol{f}^{\text{corrupt}})$ is constructed by replacing tokens with the tokens sampled from G with the concatenated translation pair as input. Formally, $\boldsymbol{e}^{\text{corrupt}}$ is constructed by

$$
\begin{cases}
e_i^{\text{corrupt}} \sim p_G(e_i | [\boldsymbol{e}; \boldsymbol{f}]^{\text{masked}}), & i \in \mathcal{M}_e \\
e_i^{\text{corrupt}} = e_i, & i \notin \mathcal{M}_e
\end{cases}
\quad (4)
$$

The same operation is also used to construct $\boldsymbol{f}^{\text{corrupt}}$. Then, the loss function of the discriminator D can be written as

$$
\mathcal{L}_D(\boldsymbol{e}, \boldsymbol{f}; \boldsymbol{\theta}_D) = - \sum_{i=1}^{n_e+n_f} \log p_D(r_i | [\boldsymbol{e}; \boldsymbol{f}]^{\text{corrupt}}) \quad (5)
$$

where r_i represents the label of whether the i-th input token is the original one or the replaced one. The final loss function of the translation replaced token detection task is $\mathcal{L}_G + \lambda \mathcal{L}_D$.

3.2 Pre-training XLM-E

The XLM-E model is jointly pretrained with the masked language modeling, translation language modeling, multilingual replaced token detection and the translation replaced token detection tasks. The overall training objective is to minimize

$$
\begin{aligned}
\mathcal{L} \; = \; & \mathcal{L}_{\text{MLM}}(\boldsymbol{x}; \theta_G) + \mathcal{L}_{\text{TLM}}(\boldsymbol{e}, \boldsymbol{f}; \theta_G) \\
& + \lambda \mathcal{L}_{\text{MRTD}}(\boldsymbol{x}; \theta_D) + \lambda \mathcal{L}_{\text{TRTD}}(\boldsymbol{e}, \boldsymbol{f}; \theta_D)
\end{aligned}
$$

over large scale multilingual corpus $\mathcal{X} = \{\boldsymbol{x}\}$ and parallel corpus $\mathcal{P} = \{(\boldsymbol{e}, \boldsymbol{f})\}$. We jointly pretrain the generator and the discriminator from scratch. Following Clark et al. (2020b), we make the generator smaller to improve the pre-training efficiency.

3.3 Gated Relative Position Bias

We propose to use gated relative position bias in the self-attention mechanism. Given input tokens $\{x_i\}_{i=1}^{|x|}$, let $\{\mathbf{h}_i\}_{i=1}^{|x|}$ denote their hidden states in Transformer. The self-attention outputs $\{\tilde{\mathbf{h}}_i\}_{i=1}^{|x|}$ are computed via:

$$\mathbf{q}_i, \mathbf{k}_i, \mathbf{v}_i = \mathbf{h}_i \mathbf{W}^Q, \mathbf{h}_i \mathbf{W}^K, \mathbf{h}_i \mathbf{W}^V \tag{6}$$

$$a_{ij} \propto \exp\{\frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d_k}} + r_{i-j}\} \tag{7}$$

$$\tilde{\mathbf{h}}_i = \sum_{j=1}^{|x|} a_{ij} \mathbf{v}_i \tag{8}$$

where r_{i-j} represents gated relative position bias, each $\mathbf{h}_i$ is linearly projected to a triple of query, key and value using parameter matrices $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d_h \times d_k}$, respectively.

Inspired by the gating mechanism of Gated Recurrent Unit (GRU; Cho et al. 2014), we compute gated relative position bias r_{i-j} via:

$$g^{(\text{update})}, g^{(\text{reset})} = \sigma(\mathbf{q}_i \cdot \mathbf{u}), \sigma(\mathbf{q}_i \cdot \mathbf{v})$$

$$\tilde{r}_{i-j} = w g^{(\text{reset})} d_{i-j}$$

$$r_{i-j} = d_{i-j} + g^{(\text{update})} d_{i-j} + (1 - g^{(\text{update})}) \tilde{r}_{i-j}$$

where d_{i-j} is learnable relative position bias, the vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{d_k}$ are parameters, σ is a sigmoid function, and w is a learnable value.

Compared with relative position bias (Parikh et al., 2016; Raffel et al., 2020; Bao et al., 2020), the proposed gates take the content into consideration, which adaptively adjusts the relative position bias by conditioning on input tokens. Intuitively, the same distance between two tokens tends to play different roles in different languages.

4 Experiments

4.1 Setup

Data We use the CC-100 (Conneau et al., 2020) dataset for the replaced token detection task. CC-100 contains texts in 100 languages collected from the CommonCrawl dump. We use parallel corpora for the translation replaced token detection task, including translation pairs in 100 languages collected from MultiUN (Ziemski et al., 2016), IIT Bombay (Kunchukuttan et al., 2018), OPUS (Tiedemann, 2012), WikiMatrix (Schwenk et al., 2019), and CCAligned (El-Kishky et al., 2020).

Following XLM (Conneau and Lample, 2019), we sample multilingual sentences to balance the language distribution. Formally, consider the pre-training corpora in N languages with m_j examples for the j-th language. The probability of using an example in the j-th language is

$$p_j = \frac{m_j^\alpha}{\sum_{k=1}^{N} m_k^\alpha} \tag{9}$$

The exponent α controls the distribution such that a lower α increases the probability of sampling examples from a low-resource language. In this paper, we set $\alpha = 0.7$.

Model We use a Base-size 12-layer Transformer (Vaswani et al., 2017) as the discriminator, with hidden size of 768, and FFN hidden size of $3,072$. The generator is a 4-layer Transformer using the same hidden size as the discriminator (Meng et al., 2021). See Appendix A for more details of model hyperparameters.

Training We jointly pretrain the generator and the discriminator of XLM-E from scratch, using the Adam (Kingma and Ba, 2015) optimizer for 125K training steps. We use dynamic batching of approximately 1M tokens for each pre-training task. We set λ, the weight for the discriminator objective to 50. The whole pre-training procedure takes about 1.7 days on 64 Nvidia A100 GPU cards. See Appendix B for more details of pre-training hyperparameters.

4.2 Cross-lingual Understanding

We evaluate XLM-E on the XTREME (Hu et al., 2020b) benchmark, which is a multilingual multi-task benchmark for evaluating cross-lingual understanding. The XTREME benchmark contains seven cross-lingual understanding tasks, namely part-of-speech tagging on the Universal Dependencies v2.5 (Zeman et al., 2019), NER named entity recognition on the Wikiann (Pan et al., 2017; Rahimi et al., 2019) dataset, cross-lingual natural language inference on XNLI (Conneau et al., 2018), cross-lingual paraphrase adversaries from word scrambling (PAWS-X; Yang et al. 2019a), and cross-lingual question answering on MLQA (Lewis et al., 2020), XQuAD (Artetxe et al., 2020), and TyDiQA-GoldP (Clark et al., 2020a).

Baselines We compare our XLM-E model with the cross-lingual language models pretrained with multilingual text, i.e., Multilingual BERT (mBERT; Devlin et al. 2019), mT5 (Xue et al., 2021), and XLM-R (Conneau et al., 2020), or

Model	Structured Prediction		Question Answering			Classification		Avg
	POS	NER	XQuAD	MLQA	TyDiQA	XNLI	PAWS-X	
Metrics	F1	F1	F1 / EM	F1 / EM	F1 / EM	Acc.	Acc.	
Pre-training on multilingual corpus								
mBERT (Hu et al., 2020b)	70.3	62.2	64.5 / 49.4	61.4 / 44.2	59.7 / 43.9	65.4	81.9	63.1
mT5 (Xue et al., 2021)	-	55.7	67.0 / 49.0	64.6 / 45.0	57.2 / 41.2	**75.4**	86.4	-
XLM-R	**75.6**	61.8	71.9 / 56.4	65.1 / 47.2	55.4 / 38.3	75.0	84.9	66.4
XLM-E (w/o TRTD)	74.2	**62.7**	**74.3 / 58.2**	**67.8 / 49.7**	**57.8 / 40.6**	75.1	**87.1**	**67.6**
Pre-training on both multilingual corpus and parallel corpus								
XLM (Hu et al., 2020b)	70.1	61.2	59.8 / 44.3	48.5 / 32.6	43.6 / 29.1	69.1	80.9	58.6
InfoXLM (Chi et al., 2021b)	-	-	- / -	68.1 / 49.6	- / -	76.5	-	-
XLM-Align (Chi et al., 2021c)	**76.0**	**63.7**	74.7 / 59.0	68.1 / **49.8**	62.1 / 44.8	76.2	86.8	68.9
XLM-E	75.6	63.5	**76.2 / 60.2**	**68.3 / 49.8**	**62.4 / 45.7**	**76.6**	**88.3**	**69.3**

Table 1: Evaluation results on XTREME cross-lingual understanding tasks. We consider the cross-lingual transfer setting, where models are only fine-tuned on the English training data but evaluated on all target languages. The compared models are all in Base size. Results of XLM-E and XLM-R are averaged over five runs.

pretrained with both multilingual text and parallel corpora, i.e., XLM (Conneau and Lample, 2019), InfoXLM (Chi et al., 2021b), and XLM-Align (Chi et al., 2021c). The compared models are all in Base size. In what follows, models are considered as in Base size by default.

Results We use the cross-lingual transfer setting for the evaluation on XTREME (Hu et al., 2020b), where the models are first fine-tuned with the English training data and then evaluated on the target languages. In Table 1, we report the accuracy, F1, or Exact-Match (EM) scores on the XTREME cross-lingual understanding tasks. The results are averaged over all target languages and five runs with different random seeds. We divide the pretrained models into two categories, i.e., the models pretrained on multilingual corpora, and the models pretrained on both multilingual corpora and parallel corpora. For the first setting, we pretrain XLM-E with only the multilingual replaced token detection task. From the results, it can be observed that XLM-E outperforms previous models on both settings, achieving the averaged scores of 67.6 and 69.3, respectively. Compared to XLM-R, XLM-E (w/o TRTD) produces an absolute 1.2 improvement on average over the seven tasks. For the second setting, compared to XLM-Align, XLM-E produces an absolute 0.4 improvement on average. XLM-E performs better on the question answering tasks and sentence classification tasks while preserving reasonable high F1 scores on structured prediction tasks. Despite the effectiveness of XLM-E, our model requires substantially lower computation cost than XLM-R and XLM-Align. A detailed efficiency analysis in presented in Section 4.5.

Model	XNLI	MLQA
XLM (reimplementation)	73.4	66.2 / 47.8
−TLM	70.6	64.0 / 46.0
XLM-E	**76.6**	**68.3 / 49.8**
−TRTD	75.1	67.8 / 49.7
−TRTD−Gated relative position bias	75.2	67.4 / 49.2

Table 2: Ablation studies of XLM-E. We studies the effects of the main components of XLM-E, and compare the models with XLM under the same pre-training setup, including training steps, learning rate, etc.

4.3 Ablation Studies

For a deeper insight to XLM-E, we conduct ablation experiments where we first remove the TRTD task and then remove the gated relative position bias. Besides, we reimplement XLM that is pretrained with the same pre-training setup with XLM-E, i.e., using the same training steps, learning rate, etc. Table 2 shows the ablation results on XNLI and MLQA. Removing TRTD weakens the performance of XLM-E on both downstream tasks. On this basis, the results on MLQA further decline when removing the gated relative position bias. This demonstrates that XLM-E benefits from both TRTD and the gated relative position bias during pre-training. Besides, XLM-E substantially outperform XLM on both tasks. Notice that when removing the two components from XLM-E, our model only requires a multilingual corpus, but still achieves better performance than XLM, which uses an additional parallel corpus.

Model	Size	Params	XNLI	MLQA
XLM-E	Base	279M	76.6	68.3 / 49.8
XLM-E	Large	840M	81.3	72.7 / 54.2
XLM-E	XL	2.2B	**83.7**	**76.2 / 57.9**
XLM-R	XL	3.5B	82.3	73.4 / 55.3
mT5	XL	3.7B	82.9	73.5 / 54.5

Table 3: Results of scaling-up the model size.

Model	XTREME	Params	FLOPs
mBERT	63.1	167M	6.4e19
XLM-R	66.4	279M	9.6e21
InfoXLM*	-	279M	9.6e21 + 1.7e20
XLM-Align*	68.9	279M	9.6e21 + 9.6e19
XLM-E	69.3	279M	9.5e19
−TRTD	67.6	279M	6.3e19

Table 4: Comparison of the pre-training costs. The models with '*' are continue-trained from XLM-R rather than pre-training from scratch.

4.4 Scaling-up Results

Scaling-up model size has shown to improve performance on cross-lingual downstream tasks (Xue et al., 2021; Goyal et al., 2021). We study the scalability of XLM-E by pre-training XLM-E models using larger model sizes. We consider two larger model sizes in our experiments, namely Large and XL. Detailed model hyperparameters can be found in Appendix A. As present in Table 3, XLM-E$_{\text{XL}}$ achieves the best performance while using significantly fewer parameters than its counterparts. Besides, scaling-up the XLM-E model size consistently improves the results, demonstrating the effectiveness of XLM-E for large-scale pre-training.

4.5 Training Efficiency

We present a comparison of the pre-training resources, to explore whether XLM-E provides a more compute-efficient and sample-efficient way for pre-training cross-lingual language models. Table 4 compares the XTREME average score, the number of parameters, and the pre-training computation cost. Notice that InfoXLM and XLM-Align are continue-trained from XLM-R, so the total training FLOPs are accumulated over XLM-R.

Table 4 shows that XLM-E substantially reduces the computation cost for cross-lingual language model pre-training. Compared to XLM-R and XLM-Align that use at least 9.6e21 training FLOPs, XLM-E only uses 9.5e19 training FLOPs in total while even achieving better XTREME performance than the two baseline models. For the setting of pre-training with only multilingual corpora, XLM-E (w/o TRTD) also outperforms XLM-R using 6.3e19 FLOPs in total. This demonstrates the compute-effectiveness of XLM-E, i.e., XLM-E as a stronger cross-lingual language model requires substantially less computation resource.

Model	Tatoeba-14 en → xx	Tatoeba-14 xx → en	Tatoeba-36 en → xx	Tatoeba-36 xx → en
XLM-R	59.5	57.6	55.5	53.4
InfoXLM	**80.6**	**77.8**	**68.6**	**67.3**
XLM-E	74.4	72.3	65.0	62.3
−TRTD	55.8	55.1	46.4	44.6

Table 5: Average accuracy@1 scores for Tatoeba cross-lingual sentence retrieval. The models are evaluated under two settings with 14 and 36 of the parallel corpora for evaluation, respectively.

4.6 Cross-lingual Alignment

To explore whether discriminative pre-training improves the resulting cross-lingual representations, we evaluate our model on the sentence-level and word-level alignment tasks, i.e., cross-lingual sentence retrieval and word alignment.

We use the Tatoeba (Artetxe and Schwenk, 2019) dataset for the cross-lingual sentence retrieval task, the goal of which is to find translation pairs from the corpora in different languages. Tatoeba consists of English-centric parallel corpora covering 122 languages. Following Chi et al. (2021b) and Hu et al. (2020b), we consider two settings where we use 14 and 36 of the parallel corpora for evaluation, respectively. The sentence representations are obtained by average pooling over hidden vectors from a middle layer. Specifically, we use layer-7 for XLM-R and layer-9 for XLM-E. Then, the translation pairs are induced by the nearest neighbor search using the cosine similarity. Table 5 shows the average accuracy@1 scores under the two settings of Tatoeba for both the xx → en and en → xx directions. XLM-E achieves 74.4 and 72.3 accuracy scores for Tatoeba-14, and 65.0 and 62.3 accuracy scores for Tatoeba-36, providing notable improvement over XLM-R. XLM-E performs slightly worse than InfoXLM. We believe the cross-lingual contrast (Chi et al., 2021b) task explicitly learns the sentence representations, which makes InfoXLM more effective for the cross-lingual sentence retrieval task.

For the word-level alignment, we use the word

Model	Alignment Error Rate ↓ en-de	en-fr	en-hi	en-ro	Avg
fast_align	32.14	19.46	59.90	-	-
XLM-R	17.74	7.54	37.79	27.49	22.64
XLM-ALIGN	16.63	6.61	33.98	26.97	21.05
XLM-E	**16.49**	**6.19**	**30.20**	**24.41**	**19.32**
−TRTD	17.87	6.29	35.02	30.22	22.35

Table 6: Alignment error rate scores (lower is better) for the word alignment task on four language pairs. Results of the baseline models are from Chi et al. (2021c). We use the optimal transport method to obtain the resulting word alignments, where the sentence representations are from the 9-th layer of XLM-E.

alignment datasets from EuroParl[1], WPT2003[2], and WPT2005[3], containing 1,244 translation pairs annotated with golden alignments. The predicted alignments are evaluated by alignment error rate (AER; Och and Ney 2003):

$$\text{AER} = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \quad (10)$$

where A, S, and P stand for the predicted alignments, the annotated sure alignments, and the annotated possible alignments, respectively. In Table 6 we compare XLM-E with baseline models, i.e., fast_align (Dyer et al., 2013), XLM-R, and XLM-ALIGN. The resulting word alignments are obtained by the optimal transport method (Chi et al., 2021c), where the sentence representations are from the 9-th layer of XLM-E. Over the four language pairs, XLM-E achieves lower AER scores than the baseline models, reducing the average AER from 21.05 to 19.32. It is worth mentioning that our model requires substantial lower computation costs than the other cross-lingual pretrained language models to achieve such low AER scores. See the detailed training efficiency analysis in Section 4.5. It is worth mentioning that XLM-E shows notable improvements over XLM-E (w/o TRTD) on both tasks, demonstrating that the translation replaced token detection task is effective for cross-lingual alignment.

4.7 Universal Layer Across Languages

We evaluate the word-level and sentence-level representations over different layers to explore whether the XLM-E tasks encourage universal representations.

[1] www-i6.informatik.rwth-aachen.de/goldAlignment/
[2] web.eecs.umich.edu/~mihalcea/wpt/
[3] web.eecs.umich.edu/~mihalcea/wpt05/

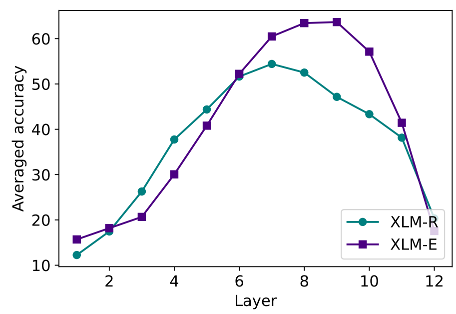


Figure 3: Evaluation results on Tatoeba cross-lingual sentence retrieval over different layers. For each layer, the accuracy score is averaged over all the 36 language pairs in both the xx → en and en → xx directions.

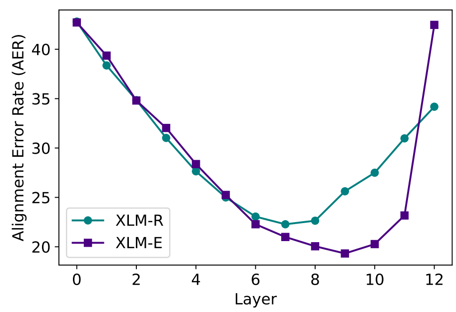


Figure 4: Evaluation results of cross-lingual word alignment over different layers. Layer-0 stands for the embedding layer.

As shown in Figure 3, we illustrate the accuracy@1 scores of XLM-E and XLM-R on Tatoeba cross-lingual sentence retrieval, using sentence representations from different layers. For each layer, the final accuracy score is averaged over all the 36 language pairs in both the xx → en and en → xx directions. From the figure, it can be observed that XLM-E achieves notably higher averaged accuracy scores than XLM-R for the top layers. The results of XLM-E also show a parabolic trend across layers, i.e., the accuracy continuously increases before a specific layer and then continuously drops. This trend is also found in other cross-lingual language models such as XLM-R and XLM-Align (Jalili Sabet et al., 2020; Chi et al., 2021c). Different from XLM-R that achieves the highest accuracy of 54.42 at layer-7, XLM-E pushes it to layer-9, achieving an accuracy of 63.66. At layer-10, XLM-R only obtains an accuracy of 43.34 while XLM-E holds the accuracy score as high as 57.14.

Figure 4 shows the averaged alignment error rate (AER) scores of XLM-E and XLM-R on the word alignment task. We use the hidden vectors from

Model	XQuAD	MLQA	TyDiQA	XNLI	PAWS-X
MBERT	25.0	27.5	22.2	16.5	14.1
XLM-R	15.9	20.3	15.2	10.4	11.4
INFOXLM	-	18.8	-	**10.3**	-
XLM-ALIGN	**14.6**	18.7	**10.6**	11.2	9.7
XLM-E	14.9	19.2	13.1	11.2	**8.8**
−TRTD	16.3	**18.6**	16.3	11.5	9.6

Table 7: The cross-lingual transfer gap scores on the XTREME tasks. A lower transfer gap score indicates better cross-lingual transferability. We use the EM scores to compute the gap scores for the QA tasks.

different layers to perform word alignment, where layer-0 stands for the embedding layer. The final AER scores are averaged over the four test sets in different languages. Figure 4 shows a similar trend to that in Figure 3, where XLM-E not only provides substantial performance improvements over XLM-R, but also pushes the best-performance layer to a higher layer, i.e., the model obtains the best performance at layer-9 rather than a lower layer such as layer-7.

On both tasks, XLM-E shows good performance for the top layers, even though both XLM-E and XLM-R use the Transformer (Vaswani et al., 2017) architecture. Compared to the masked language modeling task that encourages the top layers to be language-specific, discriminative pre-training makes XLM-E producing better-aligned text representations at the top layers. It indicates that the cross-lingual discriminative pre-training encourages universal representations inside the model.

4.8 Cross-lingual Transfer Gap

We analyze the cross-lingual transfer gap (Hu et al., 2020b) of the pretrained cross-lingual language models. The transfer gap score is the difference between performance on the English test set and the average performance on the test set in other languages. This score suggests how much end task knowledge has not been transferred to other languages after fine-tuning. A lower gap score indicates better cross-lingual transferability. Table 7 compares the cross-lingual transfer gap scores on five of the XTREME tasks. We notice that XLM-E obtains the lowest gap score only on PAWS-X. Nonetheless, it still achieves reasonably low gap scores on the other tasks with such low computation cost, demonstrating the cross-lingual transferability of XLM-E. We believe that it is more difficult to achieve the same low gap scores when the model obtains better performance.

5 Related Work

Learning self-supervised tasks on large-scale multilingual texts has proven to be effective for pre-training cross-lingual language models. Masked language modeling (MLM; Devlin et al. 2019) is typically used to learn cross-lingual encoders such as multilingual BERT (mBERT; Devlin et al. 2019) and XLM-R (Conneau et al., 2020). The cross-lingual language models can be further improved by introducing external pre-training tasks using parallel corpora. XLM (Conneau and Lample, 2019) introduces the translation language modeling (TLM) task that predicts masked tokens from concatenated translation pairs. ALM (Yang et al., 2020) utilizes translation pairs to construct code-switched sequences as input. InfoXLM (Chi et al., 2021b) considers an input translation pair as cross-lingual views of the same meaning, and proposes a cross-lingual contrastive learning task. Several pre-training tasks utilize the token-level alignments in parallel data to improve cross-lingual language models (Cao et al., 2020; Zhao et al., 2021; Hu et al., 2020a; Chi et al., 2021c).

In addition, parallel data are also employed for cross-lingual sequence-to-sequence pre-training. XNLG (Chi et al., 2020) presents cross-lingual masked language modeling and cross-lingual auto-encoding for cross-lingual natural language generation, and achieves the cross-lingual transfer for NLG tasks. VECO (Luo et al., 2020) utilizes cross-attention MLM to pretrain a variable cross-lingual language model for both NLU and NLG. mT6 (Chi et al., 2021a) improves mT5 (Xue et al., 2021) by learning the translation span corruption task on parallel data. ΔLM (Ma et al., 2021) proposes to align pretrained multilingual encoders to improve cross-lingual sequence-to-sequence pre-training.

6 Conclusion

We introduce XLM-E, a cross-lingual language model pretrained by ELECTRA-style tasks. Specifically, we present two pre-training tasks, i.e., multilingual replaced token detection, and translation replaced token detection. XLM-E outperforms baseline models on cross-lingual understanding tasks although using much less computation cost. In addition to improved performance and computational efficiency, we also show that XLM-E obtains the cross-lingual transferability with a reasonably low transfer gap.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

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*, 7(0):597–610.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. 2020. UniLMv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, pages 7006–7016.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *International Conference on Learning Representations*.

Zewen Chi, Li Dong, Shuming Ma, Shaohan Huang, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2021a. mT6: Multilingual pretrained text-to-text transformer with translation pairs. *arXiv preprint arXiv:2104.08692*.

Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. 2020. Cross-lingual natural language generation via pre-training. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7570–7577. AAAI Press.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021b. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

Zewen Chi, Li Dong, Bo Zheng, Shaohan Huang, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2021c. Improving pretrained cross-lingual language models via self-labeled word alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3418–3430, Online. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020a. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020b. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

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*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7057–7067. Curran Associates, Inc.

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

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13063–13075. Curran Associates, Inc.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. 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, pages 644–648.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. 2021. Larger-scale transformers for multilingual masked language modeling. *arXiv preprint arXiv:2105.00572*.

Junjie Hu, Melvin Johnson, Orhan Firat, Aditya Siddhant, and Graham Neubig. 2020a. Explicit alignment objectives for multilingual bidirectional encoders. *arXiv preprint arXiv:2010.07972*.

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

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643, Online. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2019. SpanBERT: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, San Diego, CA.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

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*, Miyazaki, Japan. European Language Resources Association.

Patrick Lewis, Barlas Oguz, 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*, pages 7315–7330, Online. Association for Computational Linguistics.

Fuli Luo, Wei Wang, Jiahao Liu, Yijia Liu, Bin Bi, Songfang Huang, Fei Huang, and Luo Si. 2020. VECO: Variable encoder-decoder pre-training for cross-lingual understanding and generation. *arXiv preprint arXiv:2010.16046*.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. DeltaLM: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *arXiv preprint arXiv:2106.13736*.

Yu Meng, Chenyan Xiong, Payal Bajaj, Saurabh Tiwary, Paul Bennett, Jiawei Han, and Xia Song. 2021. COCO-LM: Correcting and contrasting text sequences for language model pretraining. *arXiv preprint arXiv:2102.08473*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from wikipedia. *arXiv preprint arXiv:1907.05791*.

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

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*, pages 5998–6008. Curran Associates, Inc.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Jian Yang, Shuming Ma, Dongdong Zhang, Shuangzhi Wu, Zhoujun Li, and Ming Zhou. 2020. Alternating language modeling for cross-lingual pre-training. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019a. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. 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)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, and et al. 2019. Universal dependencies 2.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Wei Zhao, Steffen Eger, Johannes Bjerva, and Isabelle Augenstein. 2021. Inducing language-agnostic multilingual representations. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 229–240, Online. Association for Computational Linguistics.

Bo Zheng, Li Dong, Shaohan Huang, Saksham Singhal, Wanxiang Che, Ting Liu, Xia Song, and Furu Wei. 2021. Allocating large vocabulary capacity for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3203–3215, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1. 0. In *LREC*, pages 3530–3534.

Appendix

A Model Hyperparameters

Table 8 and Table 9 shows the model hyperparameters of XLM-E in the sizes of Base, Large, and XL. For the Base-size model, we use the same vocabulary with XLM-R (Conneau et al., 2020) that consists of 250K subwords tokenized by SentencePiece (Kudo and Richardson, 2018). For the models in Large size and XL size, we use VoCap (Zheng et al., 2021) to allocate a 500K vocabulary for models in Large size and XL size.

Hyperparameters	Base	Large	XL
Layers	4	6	8
Hidden size	768	1,024	1,536
FFN inner hidden size	3,072	4,096	6,144
Attention heads	12	16	24

Table 8: Model hyperparameters of XLM-E generators in different sizes.

Hyperparameters	Base	Large	XL
Layers	12	24	48
Hidden size	768	1,024	1,536
FFN inner hidden size	3,072	4,096	6,144
Attention heads	12	16	24

Table 9: Model hyperparameters of XLM-E discriminators in different sizes.

B Hyperparameters for Pre-Training

As shown in Table 10, we present the hyperparameters for pre-training XLM-E. We use the batch size of 1M tokens for each pre-training task. In multilingual replaced token detection, a batch is constructed by 2,048 length-512 input sequences, while the input length is dynamically set as the length of the original translation pairs in translation replaced token detection.

Hyperparameters	Value
Training steps	125K
Batch tokens per task	1M
Adam ϵ	1e-6
Adam β	(0.9, 0.98)
Learning rate	5e-4
Learning rate schedule	Linear
Warmup steps	10,000
Gradient clipping	2.0
Weight decay	0.01

Table 10: Hyperparameters used for pre-training XLM-E.

C Hyperparameters for Fine-Tuning

In Table 11, we report the hyperparameters for fine-tuning XLM-E on the XTREME end tasks.

	POS	NER	XQuAD	MLQA	TyDiQA	XNLI	PAWS-X
Batch size	{8,16,32}	8	32	32	32	32	32
Learning rate	{1,2,3}e-5	{5,...,9}e-6	{2,3,4}e-5	{2,3,4}e-5	{2,3,4}e-5	{5,...,8}e-6	{8,9,10,20}e-6
LR schedule	Linear	Linear	Linear	Linear	Linear	Linear	Linear
Warmup	10%	10%	10%	10%	10%	12,500 steps	10%
Weight decay	0	0	0	0	0	0	0
Epochs	10	10	4	{2,3,4}	{10,20,40}	10	10

Table 11: Hyperparameters used for fine-tuning on the XTREME end tasks.