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.

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 pre-train 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 \( x = \{x_i\}_{i=1}^n \) containing \( n \) tokens. MLM first randomly selects a subset \( M \subseteq \{1, \ldots, n\} \) as the positions to be masked, and construct the masked sentence \( x^{\text{masked}} \) by replacing tokens in \( M \) with \([\text{MASK}]\). Then, the generator predicts the probability distributions of the masked tokens \( p_G(x|x^{\text{masked}}) \). The loss function of the generator \( G \) is:

\[
\mathcal{L}_G(x; \theta_G) = -\sum_{i \in M} \log p_G(x_i|x^{\text{masked}}). \tag{1}
\]

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

\[
\begin{align*}
   x^{\text{corrupt}}_i &\sim p_G(x_i|x^{\text{masked}}), & i \in M \\
   x^{\text{corrupt}}_i &= x_i, & i \notin M
\end{align*}
\]  

(2)

Then, the discriminator predicts whether \( x^{\text{corrupt}}_i \) is original or sampled from the generator. The loss function of the discriminator \( D \) is

\[
\mathcal{L}_D(x; \theta_D) = -\sum_{i=1}^{n} \log p_D(z_i|x^{\text{corrupt}}) \tag{3}
\]

where \( z_i \) represents the label of whether \( x^{\text{corrupt}}_i \) 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, XLM-E 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

There are two pre-training tasks, including 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
In this study, the 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 \((e, f)\).

We construct the input sequence by concatenating the translation pair as a single sentence. The loss function of the generator \(G\) is:

\[
\mathcal{L}_G(e, f; \theta_G) = - \sum_{i \in \mathcal{M}_e} \log p_G(e_i | [e; f]^{\text{masked}}) - \sum_{i \in \mathcal{M}_f} \log p_G(f_i | [e; f]^{\text{masked}})
\]

where \([;]\) is the operator of concatenation, and \(\mathcal{M}_e, \mathcal{M}_f\) stand for the randomly selected masked positions for \(e\) and \(f\), 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 \((e^{\text{corrupt}}, f^{\text{corrupt}})\) is constructed by replacing tokens with the tokens sampled from \(G\) with the concatenated translation pair as input. Formally, \(e^{\text{corrupt}}\) is constructed by

\[
\left\{
\begin{array}{ll}
    e^{\text{corrupt}}_i \sim p_G(e_i | [e; f]^{\text{masked}}), & i \in \mathcal{M}_e \\
    e^{\text{corrupt}}_i = e_i, & i \notin \mathcal{M}_e
\end{array}
\right.
\]

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

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

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

\[
\mathcal{L} = \mathcal{L}_{\text{MLM}}(x; \theta_G) + \mathcal{L}_{\text{TLM}}(e, f; \theta_G) + \lambda \mathcal{L}_{\text{MRTD}}(x; \theta_D) + \lambda \mathcal{L}_{\text{TRTD}}(e, f; \theta_D)
\]

over large scale multilingual corpus \(\mathcal{X} = \{x\}\) and parallel corpus \(\mathcal{P} = \{(e, f)\}\). 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 \( \{h_i\}_{i=1}^{x} \) denote their hidden states in Transformer. The self-attention outputs \( \{h_i\}_{i=1}^{x} \) are computed via:

\[
\begin{align*}
q_i, k_i, v_i &= h_iW^Q, h_iW^K, h_iW^V \quad (6) \\
q_{ij} &\propto \exp\left(\frac{\mathbf{q}_i \cdot \mathbf{k}_j}{\sqrt{d_k}} + r_{i-j}\right) \quad (7) \\
\hat{h}_i &= \sum_{j=1}^{|x|} a_{ij}v_i \quad (8)
\end{align*}
\]

where \( r_{i-j} \) represents gated relative position bias, each \( h_i \) is linearly projected to a triple of query, key and value using parameter matrices \( W^Q, W^K, W^V \in \mathbb{R}_{d_h \times d_k} \), 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:

\[
\begin{align*}
g^{\text{update}}, g^{\text{reset}} &= \sigma(q_i \cdot u), \sigma(q_i \cdot v) \\
\tilde{r}_{i-j} &= wq^{\text{reset}}d_{i-j} \\
r_{i-j} &= d_{i-j} + g^{\text{update}}d_{i-j} + (1 - g^{\text{update}})\tilde{r}_{i-j}
\end{align*}
\]

where \( d_{i-j} \) is learnable relative position bias, the vectors \( u, v \in \mathbb{R}_{d_k} \) are parameters, \( \sigma \) 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

\[
p_j = \frac{m_j^\alpha}{\sum_{k=1}^N m_k^\alpha}
\]

The exponent \( \alpha \) controls the distribution such that a lower \( \alpha \) 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). We use the same vocabulary with XLM-R (Conneau et al., 2020) that consists of 250K subwords tokenized by SentencePiece (Kudo and Richardson, 2018).

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 \( \lambda \), 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 A 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
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. Results with “*” are taken from (Hu et al., 2020b). Results of XLM-E and XLM-R \texttt{base} are averaged over five runs.

| Model               | Structured Prediction | Question Answering | Classification | Avg  |
|---------------------|-----------------------|--------------------|----------------|------|
|                     | POS F1                | NER F1 / F1 / EM  | XQuAD F1 / EM  | MLQA F1 / EM  | TyDiQA F1 / EM  | XNLI Acc. | PAWS-X Acc. | Avg  |
| Pre-training on multilingual corpus |                       |                    |                |                  |                  |           |               |      |
| \texttt{MBERT}*     | 70.3                  | 62.2               | 64.5 / 49.4    | 61.4 / 42.2     | 59.7 / 43.9     | 65.4       | 81.9         | 63.1 |
| \texttt{MT5}_base   | -                     | 55.7               | 67.0 / 49.0    | 64.6 / 45.0     | 57.2 / 41.2     | \textbf{75.4} | 86.4         | -   |
| XLM-R\texttt{base} | \textbf{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                  | \textbf{62.7}     | \textbf{74.3} / 58.2 | \textbf{67.8} / 49.7 | 57.8 / 40.6 | 75.1       | \textbf{87.1} | \textbf{67.6} |

| Pre-training on both multilingual corpus and parallel corpus |                       |                    |                |                  |                  |           |               |      |
| XLM*               | 70.1                  | 61.2               | 59.8 / 44.3    | 48.5 / 32.6     | 43.6 / 29.1     | 69.1       | 80.9         | 58.6 |
| \texttt{INFOXLM}_\texttt{base} | -                   | -                  | - / -         | 68.1 / 49.6     | - / -          | 76.5       | -            | -    |
| XLM-\texttt{ALIGN} | \textbf{76.0}         | \textbf{63.7}     | 74.7 / 59.0    | \textbf{64.8} / 49.8 | 62.1 / 44.8 | 76.2       | 86.8         | 68.9 |
| XLM-E              | 75.6                  | 63.5               | \textbf{76.2} / 60.2 | \textbf{68.3} / 49.8 | \textbf{62.4} / 45.7 | \textbf{76.6} | \textbf{88.3} | \textbf{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. Results with “*” are taken from (Hu et al., 2020b). Results of XLM-E and XLM-R \texttt{base} are averaged over five runs.

\textbf{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 pre-trained models into two categories, i.e., the models pretrained on multilingual text and parallel corpora, i.e., XLM (Conneau and Lample, 2019), \texttt{INFOXLM} (Chi et al., 2021b), and XLM-\texttt{ALIGN} (Chi et al., 2021c).

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 2 compares the XTREME average score, the number of parameters, and the pre-training computation cost. Notice that \texttt{INFOXLM}_\texttt{base} and XLM-\texttt{ALIGN} are continue-trained from XLM-R\texttt{base}, so the total training FLOPs are accumulated over XLM-R\texttt{base}. Table 2 shows that XLM-E substantially reduces the computation cost for cross-lingual language model pre-training. Compared to XLM-R\texttt{base} and XLM-\texttt{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 mult-
Table 3: 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.

| Model       | Tatoeba-14 | Tatoeba-36 |
|-------------|------------|------------|
|             | en→xx     | xx→en     | en→xx     | xx→en     |
| XLM-R_base  | 59.5       | 57.6       | 55.5       | 53.4       |
| InfoXLM_base| 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       |

Cross-lingual corpora, XLM-E (w/o TRTD) also outperforms XLM-R_base 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.

4.4 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 3 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_base. XLM-E performs slightly worse than InfoXLM_base. We believe the cross-lingual contrast (Chi et al., 2021b) task explicitly learns the sentence representations, which makes InfoXLM_base more effective for the cross-lingual sentence retrieval task.

For the word-level alignment, we use the word alignment datasets from EuroParl¹, WPT2003², and WPT2005³, containing 1,244 translation pairs annotated with golden alignments. The predicted alignments are evaluated by alignment error rate (AER; Och and Ney 2003):

$$AER = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}$$

Table 4: Alignment error rate scores (lower is better) for the word alignment task on four English-centric 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.

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

Table 4 shows that 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 substantially lower computation costs than the other cross-lingual pre-trained language models to achieve such low AER scores. See the detailed training efficiency analysis in Section 4.3. 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.

4.5 Universal Layer Across Languages

We evaluate the word-level and sentence-level representations over different layers to explore

¹www-i6.informatik.rwth-aachen.de/goldAlignment/
²web.eecs.umich.edu/~mihalcea/wpt/
³web.eecs.umich.edu/~mihalcea/wpt05/
whether the XLM-E tasks encourage universal representations.

As shown in Figure 3, we illustrate the accuracy@1 scores of XLM-E and XLM-R_base 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_base 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_base 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_base 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_base on Tatoeba cross-lingual sentence retrieval, using sentence representations from different layers. For each layer, the final AER 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 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_base 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.

### 4.6 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. We use the EM scores to compute the gap scores for the QA tasks.

| Model      | XQuAD | MLQA | TyDiQA | XNLI | PAWS-X |
|------------|-------|------|--------|------|--------|
| MBERT      | 25.0  | 27.5 | 22.2   | 16.5 | 14.1   |
| XLM-R_base | 15.9  | 20.3 | 15.2   | 10.4 | 11.4   |
| INFOXLM_base | -     | 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 5: 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.
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). Masked sequence to sequence pre-training (MASS; Song et al. 2019), denoising auto-encoding, and span corruption (Raffel et al., 2020) are designed for learning cross-lingual sequence-to-sequence models, i.e., MASS, mBART (Liu et al., 2020), and mT5 (Xue 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 the cross-lingual contrastive learning task that aims to maximize the InfoNCE (Oord et al., 2018) lower bound of the mutual information of the two views. Contrastive learning is also used in Hictl (Wei et al., 2021) and post-pretrained multilingual BERT (Pan et al., 2020). 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., 2020; 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 by learning the translation span corruption task on parallel data. ∆LM (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. For future work, we would like to scale up XLM-E to larger model size.

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, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2020. UniLMv2: Pseudo-masked language models for unified language model pre-training. arXiv preprint arXiv:2002.12804.

Steven Cao, Nikita Kitayev, 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. arXiv preprint arXiv:2106.06381.

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.

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 multi-task benchmark for evaluating cross-lingual generalization. arXiv preprint arXiv:2003.11080.

Masoud Jafari 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
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.

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 cross-lingual understanding and generation. arXiv preprint arXiv:2010.16046.

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.

Lingxue Jiao. 2019. CAST: Cross-lingual self-training for cross-lingual named entity recognition. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 151–164, Florence, Italy. Association for Computational Linguistics.

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. 2020. mT5: A massively multilingual pre-trained text-to-text transformer. arXiv preprint arXiv:2010.11934.

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: System Demonstrations, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
A Hyperparameters for Pre-Training

As shown in Table 6, we present the hyperparameters for pre-training XLM-E, where $D$ and $G$ stand for the discriminator and the generator, respectively. 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          |
|----------------------------|----------------|
| Layers of $D$              | 12             |
| Hidden size of $D$         | 768            |
| FFN inner hidden size of $D$ | 3,072         |
| Attention heads of $D$     | 12             |
| Layers of $G$              | 4              |
| Hidden size of $G$         | 768            |
| FFN inner hidden size of $G$ | 3,072         |
| Attention heads of $G$     | 12             |
| Training steps             | 125K           |
| Batch tokens per task      | 1M             |
| Adam $\epsilon$            | 1e-6           |
| Adam $\beta$               | (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 6: Hyperparameters used for pre-training XLM-E, where $D$ means discriminator and $G$ means generator.

B Hyperparameters for Fine-Tuning

In Table 7, we report the hyperparameters for fine-tuning XLM-E on the XTREME end tasks.
| Task    | Batch size | Learning rate | LR schedule | Warmup | Weight decay | Epochs |
|---------|------------|---------------|-------------|--------|--------------|--------|
| POS     | 8,16,32    | 1,2,3e-5      | Linear      | 10%    | 0            | 10     |
| NER     | 8          | 5,...,9e-6    | Linear      | 10%    | 0            | 10     |
| XQuAD   | 32         | 2,3,4e-5      | Linear      | 10%    | 0            | 10     |
| MLQA    | 32         | 2,3,4e-5      | Linear      | 10%    | 0            | 10     |
| TyDiQA  | 32         | 5,...,8e-6    | Linear      | 10%    | 0            | 10     |
| XNLI    | 32         | 8,9,10,20e-6  | Linear      | 10%    | 0            | 10     |
| PAWS-X  | 32         | 8,9,10,20e-6  | Linear      | 10%    | 0            | 10     |

Table 7: Hyperparameters used for fine-tuning on the XTREME end tasks.