# ERNIE-M: Enhanced Multilingual Representation by Aligning Cross-lingual Semantics with Monolingual Corpora

**Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun,**
**Hao Tian, Hua Wu, Haifeng Wang**
Baidu Inc., China
{ouyangxuan, wangshuohuan, pangchao04, sunyu02}@baidu.com
{tianhao, wu_hua, wanghaifeng}@baidu.com

## Abstract

Recent studies have demonstrated that pre-trained cross-lingual models achieve impressive performance in downstream cross-lingual tasks. This improvement benefits from learning a large amount of monolingual and parallel corpora. Although it is generally acknowledged that parallel corpora are critical for improving the model performance, existing methods are often constrained by the size of parallel corpora, especially for low-resource languages. In this paper, we propose ERNIE-M, a new training method that encourages the model to align the representation of multiple languages with monolingual corpora, to overcome the constraint that the parallel corpus size places on the model performance. Our key insight is to integrate back-translation into the pre-training process. We generate pseudo-parallel sentence pairs on a monolingual corpus to enable the learning of semantic alignments between different languages, thereby enhancing the semantic modeling of cross-lingual models. Experimental results show that ERNIE-M outperforms existing cross-lingual models and delivers new state-of-the-art results in various cross-lingual downstream tasks.[1]

## 1 Introduction

Recent studies have demonstrated that the pre-training of cross-lingual language models can significantly improve their performance in cross-lingual natural language processing tasks (Devlin et al., 2018; Lample and Conneau, 2019; Conneau et al., 2019; Liu et al., 2020). Existing pre-training methods include multilingual masked language modeling (MMLM; Devlin et al. 2018) and translation language modeling (TLM; Lample and Conneau 2019), of which the key point is to learn a shared language-invariant feature space among multiple languages. MMLM implicitly models the semantic representation of each language in a unified feature space by learning them separately. TLM is an extension of MMLM that is trained with a parallel corpus and captures semantic alignment by learning a pair of parallel sentences simultaneously. This study shows that the use of parallel corpora can significantly improve the performance in downstream cross-lingual understanding and generation tasks. However, the sizes of parallel corpora are limited (Tran et al., 2020), restricting the performance of the cross-lingual language model.

To overcome the constraint of the parallel corpus size on the model performance, we propose ERNIE-M, a novel cross-lingual pre-training method to learn semantic alignment among multiple languages on monolingual corpora. Specifically, we propose cross-attention masked language modeling (CAMLM) to improve the cross-lingual transferability of the model on parallel corpora, and it trains the model to predict the tokens of one language by using another language. Then, we utilize the transferability learned from parallel corpora to enhance multilingual representation. We propose back-translation masked language modeling (BTMLM) to train the model, and this helps the model to learn sentence alignment from monolingual corpora. In BTMLM, a part of the tokens in the input monolingual sentences is predicted into the tokens of another language. We then concatenate the predicted tokens and the input sentences as pseudo-parallel sentences to train the model. In this way, the model can learn sentence alignment with only monolingual corpora and overcome the constraint of the parallel corpus size while improving the model performance.

ERNIE-M is implemented on the basis of XLM-R (Conneau et al., 2019), and we evaluate its performance on five widely used cross-lingual benchmarks: XNLI (Conneau et al., 2018) for cross-lingual natural language inference, MLQA (Lewis

[1] Code and models are available at https://github.com/PaddlePaddle/ERNIE

*Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 27–38
November 7–11, 2021. ©2021 Association for Computational Linguistics

et al., 2019) for cross-lingual question answering, CoNLL (Sang and De Meulder, 2003) for named entity recognition, cross-lingual paraphrase adversaries from word scrambling (PAWS-X) (Hu et al., 2020) for cross-lingual paraphrase identification, and Tatoeba (Hu et al., 2020) for cross-lingual retrieval. The experimental results demonstrate that ERNIE-M outperforms existing cross-lingual models and achieves new state-of-the-art (SoTA) results.

## 2 Related Work

### 2.1 Multilingual Language Models

Existing multilingual language models can be classified into two main categories: (1) discriminative models; (2) generative models.

In the first category, a multilingual bidirectional encoder representation from transformers (mBERT; Devlin et al. 2018) is pre-trained using MMLM on a monolingual corpus, which learns a shared language-invariant feature space among multiple languages. The evaluation results show that the mBERT achieves significant performance in downstream tasks (Wu and Dredze, 2019). XLM (Lample and Conneau, 2019) is extended on the basis of mBERT using TLM, which enables the model to learn cross-lingual token alignment from parallel corpora. XLM-R (Conneau et al., 2019) demonstrates the effects of models when trained on a large-scale corpus. It used 2.5T data extracted from Common Crawl (Wenzek et al., 2019) that involves 100 languages for MMLM training. The results show that a large-scale training corpus can significantly improve the performance of the cross-lingual model. Unicoder (Huang et al., 2019) achieves gains on downstream tasks by employing a multitask learning framework to learn cross-lingual semantic representations with monolingual and parallel corpora. ALM (Yang et al., 2020) improves the model's transferability by enabling the model to learn cross-lingual code-switch sentences. INFOXLM (Chi et al., 2020b) adds a contrastive learning task for cross-lingual model training. HICTL (Wei et al., 2020) learns cross-lingual semantic representation from multiple facets (at word-levels and sentence-levels) to improve the performance of cross-lingual models. VECO (Luo et al., 2020) presents a variable encoder-decoder framework to unify the understanding and generation tasks and achieves significant improvement in both downstream tasks.

The second category includes MASS (Song et al., 2019), mBART (Liu et al., 2020), XNLG (Chi et al., 2020a) and mT5 (Xue et al., 2020). MASS (Vaswani et al., 2017) proposed a training objective for restore the input sentences in which successive token fragments are masked which improved the model's performance on machine translation. Similar to MASS, mBART pre-trains a denoised sequence-to-sequence model and uses an autoregressive task to train the model. XNLG focuses on multilingual question generation and abstractive summarization and updates the parameters of the encoder and decoder through auto-encoding and autoregressive tasks. mT5 uses the same model structure and pre-training method as T5 (Raffel et al., 2019), and extends the parameters of the cross-lingual model to 13B, significantly improving the performance of the cross-language downstream tasks.

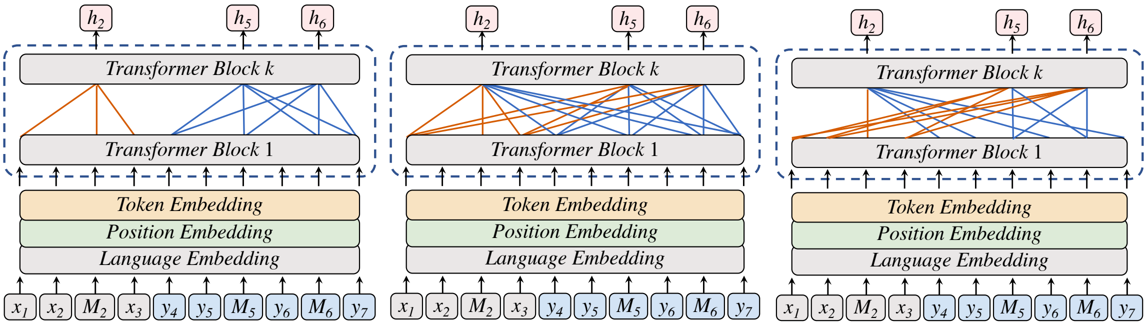
### 2.2 Back Translation and Non-Autoregressive Neural Machine Translation

Back translation (BT) is an effective neural-network-based machine translation method proposed by Sennrich et al. (2015). It can significantly improve the performance of both supervised and unsupervised machine translation via augment the parallel training corpus (Lample et al., 2017; Edunov et al., 2018). BT has been found to particularly useful when the parallel corpus is sparse (Karakanta et al., 2018). Predicting the token of the target language in one batch can also improve the speed of non-auto regressive machine translation (NAT; Gu et al. 2017; Wang et al. 2019a). Our work is inspired by NAT and BT. We generate the tokens of another language in batches and then use these in pre-training to help sentence alignment learning.

## 3 Methodology

In this section, we first introduce the general workflow of ERNIE-M and then present the details of the model training.

**Cross-lingual Semantic Alignment.** The key idea of ERNIE-M is to utilize the transferability learned from parallel corpora to enhance the model's learning of large-scale monolingual corpora, and thus enhance the multilingual semantic representation. Based on this idea, we propose two pre-training objectives, cross-attention masked language modeling (CAMLM) and back-translation



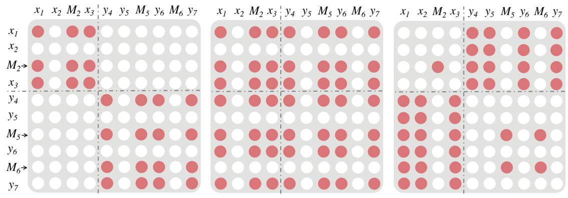
(a) MMLM (b) TLM (c) CAMLM

Figure 1: Overview of MMLM, TLM and CAMLM training. The input sentences in sub-figure (a) are monolingual sentences; $x$ and $y$ represent monolingual input sentences in different languages. The input sentences in sub-figures (b) and (c) are parallel sentences; $x$ and $y$ denote the source and target sentences of the parallel sentences, respectively. $h$ indicates the token predicted by the model.

masked language modeling (BTMLM). CAMLM is to align the cross-lingual semantic representation on parallel corpora. Then, the transferability learned from parallel corpora is utilized to enhance the multilingual representation. Specifically, we train the ERNIE-M by using BTMLM, enabling the model to align the semantics of multiple languages from monolingual corpora and improve the multilingual representation of the model. The MMLM and TLM are used by default because of the strong performance shown in Lample and Conneau 2019. We combine MMLM, TLM with CAMLM, BTMLM to train ERNIE-M. In the following sections, we will introduce the details of each objective.

**Cross-attention Masked Language Modeling.** To learn the alignment of cross-lingual semantic representations in parallel corpora, we propose a new pre-training objective, CAMLM. We denote a parallel sentence pair as <source sentence, target sentence>. In CAMLM, we learn the multilingual semantic representation by restoring the MASK token in the input sentences. When the model restores the MASK token in the source sentence, the model can only rely on the semantics of the target sentence, which means that the model has to learn how to represent the source language with the semantics of the target sentence and thus align the semantics of multiple languages.

Figure 1 (b) and (c) show the differences between TLM (Lample and Conneau, 2019) and CAMLM. TLM learns the semantic alignment between languages with both the source and target sentences while CAMLM only relies on one side of the sentence to restore the MASK token. The advantage of CAMLM is that it avoids the information leakage that the model can attend to a pair of input sentences at the same time, which makes learning of BTMLM possible. The self-attention matrix of the example in Figure 1 is shown in Figure 2. For TLM, the prediction of the MASK token relies on the input sentence pair. When the model learns CAMLM, the model can only predict the MASK token based on the sentence of its corresponding parallel sentence and the MASK symbol of this sentence, which provides the position and language information. Thus, the probability of the MASK token $M_2$ is $p(x_2|M_2, y_4, y_5, y_6, y_7)$, $p(y_5|x_1, x_2, x_3, M_5)$ for $M_5$, and $p(y_6|x_1, x_2, x_3, M_6)$ for $M_6$ in CAMLM.



(a) MMLM (b) TLM (c) CAMLM

Figure 2: Self-attention mask matrix in MMLM, TLM and CAMLM. We use different self-attention masks for different pre-training objectives.

Given the input in a bilingual corpus $X_{src} = \{x_1, x_2, \cdots, x_s\}$, and its corresponding MASK position, $M_{src} = \{m_1, m_2, \cdots, m_{ms}\}$, the target sentence is $X_{tgt} = \{x_{s+1}, x_{s+2}, \cdots, x_{s+t}\}$, and its corresponding MASK position is $M_{tgt} = \{m_{ms+1}, m_{ms+2}, \cdots, m_{ms+mt}\}$. In TLM, the model can attend to the tokens in the source and target sentences, so the probability of masked tokens is $\prod_{m \in M} p(x_m|X_{/M})$, where $M = M_{src} \cup M_{tgt}$.

$X/_M$ denotes all input tokens $x$ in $X$ except $x$ in $M$, where $X = X_{src} \cup X_{tgt}$. $x_m$ denotes the token with position $m$. In CAMLM, the probability of the MASK token in the source sentence is $\prod_{m \in M_{src}} p(x_m | X/_{M \cup X_{src}})$, which means that when predicting the MASK tokens in the source sentence, we only focus on the target sentence. As for the target sentence, the probability of the MASK token is $\prod_{m \in M_{tgt}} p(x_m | X/_{M \cup X_{tgt}})$, which means that the MASK tokens in the target sentence will be predicted based only on the source sentence. Therefore, the model must learn to use the corresponding sentence to predict and learn the alignment across multiple languages. The pre-training loss of CAMLM in the source/target sentence is

$$\mathcal{L}_{CAMLM(src)} = -\sum_{x \in D_B} log \prod_{m \in M_{src}} p(x_m | X/_{M \cup X_{src}})$$

$$\mathcal{L}_{CAMLM(tgt)} = -\sum_{x \in D_B} log \prod_{m \in M_{tgt}} p(x_m | X/_{M \cup X_{tgt}})$$

where $D_B$ is the bilingual training corpus. The CAMLM loss is

$$\mathcal{L}_{CAMLM} = \mathcal{L}_{CAMLM(src)} + \mathcal{L}_{CAMLM(tgt)}$$

**Back-translation Masked Language Modeling.** To overcome the constraint that the parallel corpus size places on the model performance, we propose a novel pre-training objective inspired by NAT (Gu et al., 2017; Wang et al., 2019a) and BT methods called BTMLM to align cross-lingual semantics with the monolingual corpus. We use BTMLM to train our model, which builds on the transferability learned through CAMLM, generating pseudo-parallel sentences from the monolingual sentences and the generated pseudo-parallel sentences are then used as the input of the model to align the cross-lingual semantics, thus enhancing the multilingual representation. The training process for BTMLM is shown in Figure 3.

The learning process for the BTMLM is divided into two stages. Stage 1 involves the generation of pseudo-parallel tokens from monolingual corpora. Specifically, we fill in several placeholder MASK at the end of the monolingual sentence to indicate the location and the language we want to generate, and let the model generate its corresponding parallel language token based on the original monolingual sentence and the corresponding position of the pseudo-token. In this way, we generate the tokens of another language from the monolingual sentence, which will be used in learning cross-lingual semantic alignment for multiple languages.

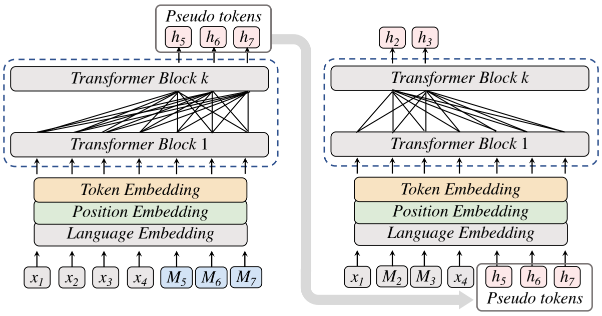


Figure 3: Overview of BTMLM training; the left figure represents the first stage of BTMLM, predicting the pseudo-tokens. The right figure represents the second stage of the BTMLM, making predictions based on the predicted pseudo-tokens and original sentences.

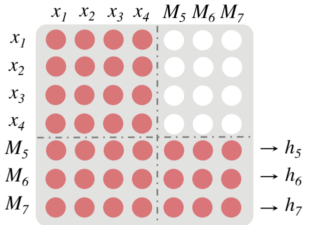


Figure 4: Self-attention matrix of BTMLM Stage 1.

The self-attention matrix for generating pseudo-tokens in Figure 3 is shown in Figure 4. In the pseudo-token generating process, the model can only attend to the source sentence and the placeholder MASK tokens, which indicate the language and position we want to predict by using language embedding and position embedding. The probability of mask token $M_5$ is $p(y_5|x_1, x_2, x_3, x_4, M_5)$, $p(y_6|x_1, x_2, x_3, x_4, M_6)$ for $M_6$ and $p(y_7|x_1, x_2, x_3, x_4, M_7)$ for $M_7$.

Stage 2 uses the pseudo-tokens generated in Stage 1 to learn the cross-lingual semantics alignment. The process in Stage 2 is shown in the right-hand diagram of Figure 3. In the training process of Stage 2, the input of the model is the concatenation of the monolingual sentences and the generated pseudo-parallel tokens, and the learning objective is to restore the MASK tokens based on the original sentences and the generated pseudo-parallel tokens. Because the model can rely not only on the input monolingual sentence but also the generated pseudo-tokens in the process of inference MASK tokens, the model can explicitly learn the alignment of the cross-lingual semantic representation from the monolingual sentences.

The learning process of the BTMLM can be interpreted as follows: given the input in monolingual corpora $X = \{x_1, x_2, \cdots, x_s\}$, the positions of masked tokens $M = \{m_1, m_2, \cdots, m_m\}$ and the position of the pseudo-token to be predicted, $M_{pseudo} = \{m_{s+1}, m_{s+2}, \cdots, m_{s+p}\}$, we first generate pseudo-tokens $P = \{h_{s+1}, h_{s+2}, \cdots, h_{s+p}\}$, as described earlier; we then concatenate the generated pseudo-token with input monolingual sentence as a new parallel sentence pair and use it to train our model. Thus, the probability of the masked tokens in BTMLM is $\prod_{m \in M} p(x_m|X/_M, P)$, where $X/_M$ denotes all input tokens $x$ in $X$ except $x$ in $M$. The pre-training loss of BTMLM is

$$\mathcal{L}_{BTMLM} = - \sum_{x \in D_M} log \prod_{m \in M} p(x_m|X/_M, P)$$

where $D_M$ is the monolingual training corpus.

# 4 Experiments

We consider five cross-lingual evaluation benchmarks: XNLI for cross-lingual natural language inference, MLQA for cross-lingual question answering, CoNLL for cross-lingual named entity recognition, PAWS-X for cross-lingual paraphrase identification, and Tatoeba for cross-lingual retrieval. Next, we first describe the data and pre-training details and then compare the ERNIE-M with the existing state-of-the-art models.

## 4.1 Data and Model

ERNIE-M is trained with monolingual and parallel corpora that involved 96 languages. For the monolingual corpus, we extract it from CC-100 according to Wenzek et al. (2019); Conneau et al. (2019). For the bilingual corpus, we use the same corpus as INFOXLM (Chi et al., 2020b), including MultiUN (Ziemski et al., 2016), IIT Bombay (Kunchukuttan et al., 2017), OPUS (Tiedemann, 2012), and WikiMatrix (Schwenk et al., 2019)

We use a transformer-encoder (Vaswani et al., 2017) as the backbone of the model. For the ERNIE-M$_{\text{BASE}}$ model, we adopt a structure with 12 layers, 768 hidden units, 12 heads. For ERNIE-M$_{\text{LARGE}}$ model , we adopt a structure with 24 layers, 1024 hidden units, 16 heads. The activation function used is GeLU (Hendrycks and Gimpel, 2016). Following Chi et al. 2020b and Luo et al. 2020, we initialize the parameters of ERNIE-M with XLM-R. We use the Adam optimizer (Kingma and Ba, 2014) to train ERNIE-M; the learning rate is scheduled with a linear decay with 10K warm-up steps, and the peak learning rate is $2e-4$ for the base model and $1e-4$ for the large model. We conduct the pre-training experiments using 64 Nvidia V100-32GB GPUs with 2048 batch size and 512 max length.

## 4.2 Experimental Evaluation

**Cross-lingual Natural Language Inference.** The cross-lingual natural language inference (XNLI; Conneau et al. 2018) task is a multilingual language inference task. The goal of XNLI is to determine the relationship between the two input sentences. We evaluate ERNIE-M in (1) cross-lingual transfer (Conneau et al., 2018) setting: fine-tune the model with an English training set and evaluate the foreign language XNLI test and (2) translate-train-all (Huang et al., 2019) setting: fine-tune the model on the concatenation of all other languages and evaluate on each language test set.

Table 1 shows the results of ERNIE-M in XNLI task. The result shows that ERNIE-M outperforms all baseline models including XLM (Lample and Conneau, 2019), Unicoder (Huang et al., 2019), XLM-R (Conneau et al., 2019), INFOXLM (Chi et al., 2020b) and VECO (Luo et al., 2020) on both the evaluation settings on XNLI. The final scores on the test set are averaged over five runs with different random seeds. On cross-lingual transfer setting, ERNIE-M achieves 77.3 average accuracy, outperforming INFOXLM by 1.1, ERNIE-M$_{\text{LARGE}}$ achieves 82.0 accuracy, outperforming INFOXLM$_{\text{LARGE}}$ by 0.6. ERNIE-M also yields outstanding performance in low-resource languages, including 69.5 in Swahili (sw) and 68.8 in Urdu (ur). In the case of translate-train-all, ERNIE-M improves the performance and reaches an accuracy of 80.6, outperforming INFOXLM by 0.9, ERNIE-M$_{\text{LARGE}}$ achieves 84.2 accuracy, a new SoTA for XNLI, outperforming XLM-R$_{\text{LARGE}}$ by 0.6.

**Named Entity Recognition.** For the named-entity-recognition task, we evaluate ERNIE-M on the CoNLL-2002 and CoNLL-2003 datasets (Sang and De Meulder, 2003), which is a cross-lingual named-entity-recognition task including English, Dutch, Spanish and German. We consider ERNIE-M in the following setting: (1) fine-tune on the

| Model | en | fr | es | de | el | bg | ru | tr | ar | vi | th | zh | hi | sw | ur | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Fine-tune cross-lingual model on English training set (Cross-lingual Transfer)* | | | | | | | | | | | | | | | | |
| XLM (Lample and Conneau, 2019) | 85.0 | 78.7 | 78.9 | 77.8 | 76.6 | 77.4 | 75.3 | 72.5 | 73.1 | 76.1 | 73.2 | 76.5 | 69.6 | 68.4 | 67.3 | 75.1 |
| Unicoder (Huang et al., 2019) | 85.1 | 79.0 | 79.4 | 77.8 | 77.2 | 77.2 | 76.3 | 72.8 | 73.5 | 76.4 | 73.6 | 76.2 | 69.4 | 69.7 | 66.7 | 75.4 |
| XLM-R (Conneau et al., 2019) | 85.8 | 79.7 | 80.7 | 78.7 | 77.5 | 79.6 | 78.1 | 74.2 | 73.8 | 76.5 | 74.6 | 76.7 | 72.4 | 66.5 | 68.3 | 76.2 |
| INFOXLM (Chi et al., 2020b) | **86.4** | **80.6** | 80.8 | 78.9 | 77.8 | 78.9 | 77.6 | 75.6 | 74.0 | 77.0 | 73.7 | 76.7 | 72.0 | 66.4 | 67.1 | 76.2 |
| ERNIE-M | 85.5 | 80.1 | **81.2** | **79.2** | **79.1** | **80.4** | **78.1** | **76.8** | **76.3** | **78.3** | **75.8** | **77.4** | **72.9** | **69.5** | **68.8** | **77.3** |
| XLM-R$_{\text{LARGE}}$ (Conneau et al., 2019) | 89.1 | 84.1 | 85.1 | 83.9 | 82.9 | 84.0 | 81.2 | 79.6 | 79.8 | 80.8 | 78.1 | 80.2 | 76.9 | 73.9 | 73.8 | 80.9 |
| INFOXLM$_{\text{LARGE}}$ (Chi et al., 2020b) | **89.7** | 84.5 | 85.5 | 84.1 | 83.4 | 84.2 | 81.3 | 80.9 | 80.4 | 80.8 | 78.9 | 80.9 | 77.9 | 74.8 | 73.7 | 81.4 |
| VECO$_{\text{LARGE}}$ (Luo et al., 2020) | 88.2 | 79.2 | 83.1 | 82.9 | 81.2 | 84.2 | **82.8** | 76.2 | 80.3 | 74.3 | 77.0 | 78.4 | 71.3 | **80.4** | **79.1** | 79.9 |
| ERNIE-M$_{\text{LARGE}}$ | 89.3 | **85.1** | **85.7** | **84.4** | **83.7** | **84.5** | 82.0 | **81.2** | **81.2** | **81.9** | **79.2** | **81.0** | **78.6** | 76.2 | 75.4 | **82.0** |
| *Fine-tune cross-lingual model on all training sets (Translate-Train-All)* | | | | | | | | | | | | | | | | |
| XLM (Lample and Conneau, 2019) | 85.0 | 80.8 | 81.3 | 80.3 | 79.1 | 80.9 | 78.3 | 75.6 | 77.6 | 78.5 | 76.0 | 79.5 | 72.9 | 72.8 | 68.5 | 77.8 |
| Unicoder (Huang et al., 2019) | 85.6 | 81.1 | 82.3 | 80.9 | 79.5 | 81.4 | 79.7 | 76.8 | 78.2 | 77.9 | 77.1 | 80.5 | 73.4 | 73.8 | 69.6 | 78.5 |
| XLM-R (Conneau et al., 2019) | 85.4 | 81.4 | 82.2 | 80.3 | 80.4 | 81.3 | 79.7 | 78.6 | 77.3 | 79.7 | 77.9 | 80.2 | 76.1 | 73.1 | 73.0 | 79.1 |
| INFOXLM (Chi et al., 2020b) | 86.1 | 82.0 | 82.8 | 81.8 | 80.9 | 82.0 | 80.2 | 79.0 | 78.8 | 80.5 | 78.3 | 80.5 | 77.4 | 73.0 | 71.6 | 79.7 |
| ERNIE-M | **86.2** | **82.5** | **83.8** | **82.6** | **82.4** | **83.4** | **80.2** | **80.6** | **80.5** | **81.1** | **79.2** | **80.5** | **77.7** | **75.0** | **73.3** | **80.6** |
| XLM-R$_{\text{LARGE}}$ (Conneau et al., 2019) | 89.1 | 85.1 | 86.6 | 85.7 | 85.3 | 85.9 | 83.5 | 83.2 | 83.1 | 83.7 | 81.5 | **83.7** | **81.6** | 78.0 | 78.1 | 83.6 |
| VECO$_{\text{LARGE}}$ (Luo et al., 2020) | 88.9 | 82.4 | 86.0 | 84.7 | 85.3 | 86.2 | **85.8** | 80.1 | 83.0 | 77.2 | 80.9 | 82.8 | 75.3 | **83.1** | **83.0** | 83.0 |
| ERNIE-M$_{\text{LARGE}}$ | **89.5** | **86.5** | **86.9** | **86.1** | **86.0** | **86.8** | 84.1 | **83.8** | **84.1** | **84.5** | **82.1** | 83.5 | 81.1 | 79.4 | 77.9 | **84.2** |

Table 1: Evaluation results on XNLI cross-lingual natural language inference. We report the accuracy on each of the 15 XNLI languages and the average accuracy. Our ERNIE-M results are based on five runs with different random seeds.

| Model | en | nl | es | de | Avg |
|---|---|---|---|---|---|
| *Fine-tune on English dataset* | | | | | |
| mBERT$^{*}$ | 91.97 | 77.57 | 74.96 | 69.56 | 78.52 |
| XLM-R$^{\dagger}$ | 92.25 | **78.08** | 76.53 | **69.60** | 79.11 |
| ERNIE-M | **92.78** | 78.01 | **79.37** | 68.08 | **79.56** |
| XLM-R$^{\dagger}_{\text{LARGE}}$ | 92.92 | 80.80 | 78.64 | 71.40 | 80.94 |
| ERNIE-M$_{\text{LARGE}}$ | **93.28** | **81.45** | **78.83** | **72.99** | **81.64** |
| *Fine-tune on all dataset* | | | | | |
| XLM-R$^{\dagger}$ | 91.08 | 89.09 | 87.28 | 83.17 | 87.66 |
| ERNIE-M | **93.04** | **91.73** | **88.33** | **84.20** | **89.32** |
| XLM-R$^{\dagger}_{\text{LARGE}}$ | 92.00 | 91.60 | **89.52** | 84.60 | 89.43 |
| ERNIE-M$_{\text{LARGE}}$ | **94.01** | **93.81** | 89.23 | **86.20** | **90.81** |

Table 2: Evaluation results on CoNLL named entity recognition. The results of ERNIE-M are averaged over five runs. Results with "†" and "∗" are from (Conneau et al., 2019), and (Wu and Dredze, 2019), respectively.

English dataset and evaluate on each cross-lingual dataset to evaluate cross-lingual transfer and (2) fine-tune on all training datasets to evaluate cross-lingual learning. For each setting, we reported the F1 score for each language.

Table 2 shows the results of ERNIE-M, XLM-R, and mBERT on CoNLL-2002 and CoNLL-2003. The results of XLM-R and mBERT are reported from Conneau et al. (2019). ERNIE-M model yields SoTA performance on both settings and outperforms XLM-R by 0.45 F1 when trained on English and 0.70 F1 when trained on all languages in the base model. Similar to the performance in the XNLI task, ERNIE-M shows better performance on low-resource languages. For large models and fine-tune in all languages setting, ERNIE-M is 2.21 F1 higher than SoTA in Dutch (nl) and 1.6 F1 higher than SoTA in German (de).

**Cross-lingual Question Answering.** For the question answering task, we use a multilingual question answering (MLQA) dataset to evaluate ERNIE-M. MLQA has the same format as SQuAD v1.1 (Rajpurkar et al., 2016) and is a multilingual language question answering task composed of seven languages. We fine-tune ERNIE-M by training on English data and evaluating on seven cross-lingual datasets. The fine-tune method is the same as in Lewis et al. (2019), which concatenates the question-passage pair as the input.

Table 3 presents a comparison of ERNIE-M and several baseline models on MLQA. We report the F1 and extract match (EM) scores based on the average over five runs. The performance of ERNIE-M in MLQA is significantly better than the previous models, and it achieves a SoTA score. We outperform INFOXLM 0.8 in F1 and 0.5 in EM.

**Cross-lingual Paraphrase Identification.** For cross-lingual paraphrase identification task, we use the PAWS-X (Hu et al., 2020) dataset to evaluate our model. The goal of PAWS-X was to determine whether two sentences were paraphrases. We evaluate ERNIE-M on both the cross-lingual transfer setting and translate-train-all setting.

Table 4 shows a comparison of ERNIE-M and various baseline models on PAWS-X. We report the accuracy score on each language test set based on the average over five runs. The results show that

| Model | en | es | de | ar | hi | vi | zh | Avg |
|---|---|---|---|---|---|---|---|---|
| mBERT (Lewis et al., 2019) | 77.7 / 65.2 | 64.3 / 46.6 | 57.9 / 44.3 | 45.7 / 29.8 | 43.8 / 29.7 | 57.1 / 38.6 | 57.5 / 37.3 | 57.7 / 41.6 |
| XLM (Lewis et al., 2019) | 74.9 / 62.4 | 68.0 / 49.8 | 62.2 / 47.6 | 54.8 / 36.3 | 48.8 / 27.3 | 61.4 / 41.8 | 61.1 / 39.6 | 61.6 / 43.5 |
| XLM-R (Conneau et al., 2019) | 77.1 / 64.6 | 67.4 / 49.6 | 60.9 / 46.7 | 54.9 / 36.6 | 59.4 / 42.9 | 64.5 / 44.7 | 61.8 / 39.3 | 63.7 / 46.3 |
| INFOXLM (Chi et al., 2020b) | 81.3 / 68.2 | 69.9 / 51.9 | 64.2 / 49.6 | 60.1 / 40.9 | 65.0 / 47.5 | 70.0 / 48.6 | 64.7 / **41.2** | 67.9 / 49.7 |
| ERNIE-M | **81.6 / 68.5** | **70.9 / 52.6** | **65.8 / 50.7** | **61.8 / 41.9** | **65.4 / 47.5** | **70.0 / 49.2** | **65.6** / 41.0 | **68.7 / 50.2** |
| XLM-R$_{\text{LARGE}}$ (Conneau et al., 2019) | 80.6 / 67.8 | 74.1 / 56.0 | 68.5 / 53.6 | 63.1 / 43.5 | 62.9 / 51.6 | 71.3 / 50.9 | 68.0 / 45.4 | 70.7 / 52.7 |
| INFOXLM$_{\text{LARGE}}$ (Chi et al., 2020b) | **84.5 / 71.6** | **75.1 / 57.3** | **71.2 / 56.2** | **67.6 / 47.6** | 72.5 / 54.2 | **75.2 / 54.1** | 69.2 / 45.4 | 73.6 / 55.2 |
| ERNIE-M$_{\text{LARGE}}$ | 84.4 / 71.5 | 74.8 / 56.6 | 70.8 / 55.9 | 67.4 / 47.2 | **72.6 / 54.7** | 75.0 / 53.7 | **71.1 / 47.5** | **73.7 / 55.3** |

Table 3: Evaluation results on MLQA cross-lingual question answering. We report the F1 and exact match (EM) scores. The results of ERNIE-M are averaged over five runs.

| Model | en | de | es | fr | ja | ko | zh | Avg |
|---|---|---|---|---|---|---|---|---|
| *Cross-lingual Transfer* | | | | | | | | |
| mBERT† | 94.0 | 85.7 | 87.4 | 87.0 | 73.0 | 69.6 | 77.0 | 81.9 |
| XLM† | 94.0 | 85.9 | 88.3 | 87.4 | 69.3 | 64.8 | 76.5 | 80.9 |
| MMTE† | 93.1 | 85.1 | 87.2 | 86.9 | 72.0 | 69.2 | 75.9 | 81.3 |
| XLM-R$^{\dagger}_{\text{LARGE}}$ | 94.7 | 89.7 | 90.1 | 90.4 | 78.7 | 79.0 | 82.3 | 86.4 |
| VECO$^{*}_{\text{LARGE}}$ | **96.2** | 91.3 | 91.4 | 92.0 | 81.8 | 82.9 | 85.1 | 88.7 |
| ERNIE-M$_{\text{LARGE}}$ | 96.0 | **91.9** | **91.4** | **92.2** | **83.9** | **84.5** | **86.9** | **89.5** |
| *Translate-Train-All* | | | | | | | | |
| VECO$^{*}_{\text{LARGE}}$ | 96.4 | 93.0 | 93.0 | 93.5 | 87.2 | 86.8 | 87.9 | 91.1 |
| ERNIE-M$_{\text{LARGE}}$ | **96.5** | **93.5** | **93.3** | **93.8** | **87.9** | **88.4** | **89.2** | **91.8** |

Table 4: Evaluation results on PAWS-X. The results of ERNIE-M are averaged over five runs. Results with "†" and "$*$" are from (Hu et al., 2020) and (Luo et al., 2020), respectively.

ERNIE-M outperforms all baseline models on most languages and achieves a new SoTA.

**Cross-lingual Sentence Retrieval.** The goal of the cross-lingual sentence retrieval task was to extract parallel sentences from bilingual corpora. We used a subset of the Tatoeba (Hu et al., 2020) dataset, which contains 36 language pairs to evaluate ERNIE-M. Following Luo et al. 2020, we used the averaged representation in the middle layer of the best XNLI model to evaluate the retrieval task.

Table 5 shows the results of ERNIE-M in the retrieval task; XLM-R results are reported from Luo et al. 2020. ERNIE-M achieves a score of 87.9 in the Tatoeba dataset, outperforming VECO 1.0 and obtaining new SoTA results.

| Model | Avg |
|---|---|
| XLM-R$_{\text{LARGE}}$ (Luo et al., 2020) | 75.2 |
| VECO$_{\text{LARGE}}$ (Luo et al., 2020) | 86.9 |
| ERNIE-M$_{\text{LARGE}}$ | **87.9** |
| ERNIE-M$^{\dagger}_{\text{LARGE}}$ | 93.3 |

Table 5: Evaluation results on Tatoeba. "†" indicates the results after fine-tuning.

To further evaluate the performance of ERNIE-M in retrieval task, we use hardest negative binary cross-entropy loss (Wang et al., 2019b; Faghri et al., 2017) to fine-tune ERNIE-M with the same bilingual corpus in pre-training. Figure 5 shows the details of accuracy on each language in Tatoeba. After fine-tuning, ERNIE-M shows a significant improvement in all languages, with the average accuracy in all languages increasing from 87.9 to 93.3.

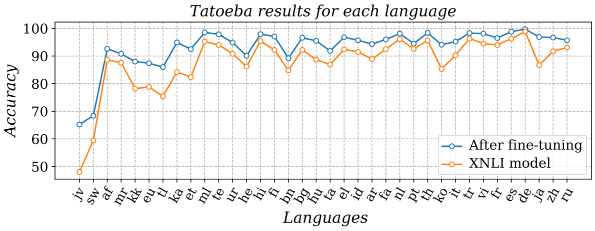


Figure 5: Tatoeba results for each language. The languages are sorted according to their size in the pre-trained corpus from smallest to largest. Fine-tuning can significantly improve the accuracy of different language families in the cross-lingual retrieval task.

## 4.3 Ablation Study

To understand the effect of aligning semantic representations of multiple languages in the training process of ERNIE-M, we conducted an ablation study as reported in Table 6. $exp_0$ was directly fine-tuning XLM-R model on the XNLI and the CoNLL. We trained (1) only MMLM on the monolingual corpus, and the purpose of $exp_1$ was to measure how much performance gain could be achieved by continuing training based on the XLM-R model, (2) MMLM on the monolingual corpus, and TLM on the bilingual corpus, (3) MMLM on the monolingual corpus and CAMLM on the bilingual corpus, (4) MMLM and BTMLM on the monolingual corpus and CAMLM on the bilingual corpus and (5) full strategy of ERNIE-M. We use the base model structure for our experiments, and to speed up the experiments, we use the XLM-R$_{\text{BASE}}$ model to initialize the parameters of ERNIE-M, all of which run 50,000 steps with the same hyperparameters with a batch size of 2048, and the score reported in the downstream task is the average score of five runs.

Comparing $exp_0$ and $exp_1$, we can observer that there is no gain in the performance of the cross-lingual model by continuing pre-training XLM-

| Index | Monolingual | Bilingual | XNLI | CoNLL |
|---|---|---|---|---|
| $exp_0$ | / | / | 75.7 | 79.2 |
| $exp_1$ | MMLM | / | 75.8 | 79.2 |
| $exp_2$ | MMLM | TLM | 76.3 | 78.3 |
| $exp_3$ | MMLM | CAMLM | 76.1 | 79.5 |
| $exp_4$ | MMLM + BTMLM | CAMLM | 76.6 | 79.6 |
| $exp_5$ | MMLM + BTMLM | CAMLM + TLM | **76.9** | **79.6** |

Table 6: Ablation study on each task in ERNIE-M.

| Model | MLQA | XNLI | Avg |
|---|---|---|---|
| mBERT | 23.3 | 16.9 | 20.1 |
| XLM-R | 17.6 | 10.4 | 14.0 |
| INFOXLM | 15.7 | 10.9 | 13.3 |
| ERNIE-M | **15.0** | **8.8** | **11.9** |

Table 7: Cross-lingual transfer gap score, smaller gap indicates better transferability.

R model. Comparing $exp_2$ $exp_3$ $exp_4$ with $exp_1$, we find that the learning of cross-lingual semantic alignment on parallel corpora is helpful for the performance of the model. Experiments that use the bilingual corpus for training show a significant improvement in XNLI. However, there are a surprised result that the using of TLM objective hurt the performance of NER task as $exp_1$ and $exp_2$ shows. Comparing $exp_2$ with $exp_4$, we find that our proposed BTMLM and CAMLM training objective are better for capturing the alignment of cross-lingual semantics. The training model with CAMLM and BTMLM objective results in a 0.3 improvement on XNLI and a 1.3 improvement on CoNLL compared to the training model with TLM. Comparing $exp_3$ to $exp_4$, we find that there is a 0.5 improvement on XNLI and 0.1 improvement on CoNLL after the model learns BTMLM. This demonstrates that our proposed BTMLM can learn cross-lingual semantic alignment and improve the performance of our model.

To further analyze the effect of our strategy, we trained the small-sized ERNIE-M model from scratch. Table 8 shows the results of XNLI and CoNLL. Both XNLI and CoNLL results are the average of each languages. We observe that, ERNIE-M$_{\text{SMALL}}$ can outperform XLM-R$_{\text{SMALL}}$ by 4.4 in XNLI and 6.6 in CoNLL. It suggests that our models can benefit from align cross-lingual semantic representation.

Table 7 shows the gap scores for English and other languages in the downstream task. This gap score is the difference between the English testset and the average performance on the testset in other languages. So, a smaller gap score represents a better transferability of the model. We can notice that the gap scores of ERNIE-M are smaller compared to XLM-R and INFOXLM in both the XNLI and MLQA tasks, which indicates a better transferability of ERNIE-M.

| Model | CoNLL | XNLI |
|---|---|---|
| XLM-R | 63.2 | 55.7 |
| XLM-R + TLM | 65.6 | 67.3 |
| XLM-R + CAMLM | 66.4 | 66.9 |
| XLM-R + CAMLM + BTMLM | 69.5 | 68.9 |
| ERNIE-M* | 69.7 | 69.9 |
| ERNIE-M | **69.8** | **70.1** |

Table 8: XNLI and CoNLL accuracy under the cross-lingual transfer setting. All the models are small-sized trained from scratch. The small-sized model has the same hyperparameter as base model except that the number of layers is 6. ERNIE-M* is the result in downstream tasks with the same computational overhead as XLM-R. All the models have the same training steps except ERNIE-M*.

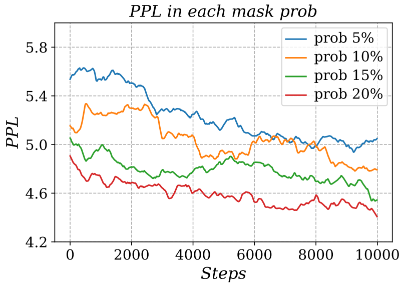


Figure 6: PPL in BTMLM training with different mask prob, prob means the proportion of pseudo-tokens generated in BTMLM Stage 1.

To measure the computation cost of ERNIE-M, we trained ERNIE-M and XLM-R (MMLM + TLM) from scratch. The result shows that the training speed of ERNIE-M is 1.075x compared with XLM-R, so the overall computational of ERNIE-M is 1.075x compared with XLM-R. With the same computational overhead, the performance of ERNIE-M is 69.9 in XNLI and 69.7 in CoNLL, while XLM-R's performance is 67.3 in XNLI and 65.6 in CoNLL. The results demonstrate that ERNIE-M performs better than XLM-R even with the same computational overhead.

In addition, we explored the effect of the number of generated pseudo-parallel tokens on the convergence of the model. In particular, we compare the impact on the convergence speed of the model when generating a 5%, 10%, 15%, and 20% proportion of pseudo-tokens. As shown in Figure 6, we can find that the perplexity (PPL) of the model decreases as the proportion of generated tokens increases, which indicates that the generated pseudo-parallel tokens are helpful for model convergence.

## 5 Conclusion

To overcome the constraint that the parallel corpus size places on the cross-lingual models performance, we propose a new cross-lingual model, ERNIE-M, which is trained using both monolingual and parallel corpora. The contribution of ERNIE-M is to propose two training objectives. The first objective is to enhance the multilingual representation on parallel corpora by applying CAMLM, and the second objective is to help the model to align cross-lingual semantic representations from a monolingual corpus by using BTMLM. Experiments show that ERNIE-M achieves SoTA results in various downstream tasks on the XNLI, MLQA, CoNLL, PAWS-X, and Tatoeba datasets.

## References

Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. 2020a. Cross-lingual natural language generation via pre-training. In *AAAI*, pages 7570–7577.

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

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

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2017. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

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*.

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. *arXiv preprint arXiv:1909.00964*.

Alina Karakanta, Jon Dehdari, and Josef van Genabith. 2018. Neural machine translation for low-resource languages without parallel corpora. *Machine Translation*, 32(1):167–189.

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

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2017. The iit bombay english-hindi parallel corpus. *arXiv preprint arXiv:1710.02855*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*.

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

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*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

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*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218.

Chau Tran, Yuqing Tang, Xian Li, and Jiatao Gu. 2020. Cross-lingual retrieval for iterative self-supervised training. *Advances in Neural Information Processing Systems*, 33.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008.

Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019a. Non-autoregressive machine translation with auxiliary regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5377–5384.

Zihao Wang, Xihui Liu, Hongsheng Li, Lu Sheng, Junjie Yan, Xiaogang Wang, and Jing Shao. 2019b. Camp: Cross-modal adaptive message passing for text-image retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5764–5773.

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*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. *arXiv preprint arXiv:1904.09077*.

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 *AAAI*, pages 9386–9393.

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'16)*, pages 3530–3534.

# A Appendix

## A.1 Pre-training Data

We follow (Wenzek et al., 2019) to reconstruct CC-100 data for ERNIE-M training. The monolingual training corpus contains 96 languages, as shown in Table 9. Note that several languages have the same ISO code, e.g., zh represents both Simplified Chinese and Traditional Chinese; ur represents both Urdu and Urdu Romanized. Table 10 shows the statistics of the parallel data in each language.

| Code | Size (GB) | Code | Size (GB) | Code | Size (GB) |
|---|---|---|---|---|---|
| af | 0.1 | hi | 4.2 | or | 0.3 |
| am | 0.3 | hr | 1.0 | pa | 0.6 |
| ar | 12.5 | hu | 6.9 | pl | 20.2 |
| as | 0.1 | hy | 0.6 | ps | 0.3 |
| az | 0.6 | id | 11.7 | pt | 27.4 |
| be | 0.4 | is | 0.4 | ro | 7.5 |
| bg | 5.6 | it | 32.9 | ru | 215.6 |
| bn | 4.6 | ja | 78.1 | sa | 0.1 |
| br | 0.1 | jv | 0.1 | sd | 0.1 |
| bs | 0.1 | ka | 0.9 | si | 1.1 |
| ca | 2.1 | kk | 0.5 | sk | 9.5 |
| cs | 10.5 | km | 0.2 | sl | 4.3 |
| cy | 0.3 | kn | 0.2 | so | 0.1 |
| da | 4.8 | ko | 29.4 | sq | 2.0 |
| de | 71.0 | ku | 0.1 | sr | 5.5 |
| el | 10.5 | ky | 0.4 | su | 0.1 |
| en | 512.5 | la | 0.2 | sv | 42.1 |
| eo | 0.4 | lo | 0.2 | sw | 0.2 |
| es | 62.6 | lt | 1.7 | ta | 6.9 |
| et | 1.0 | lv | 0.9 | te | 2.0 |
| eu | 0.7 | mg | 0.1 | th | 29.1 |
| fa | 14.8 | mk | 0.5 | tl | 0.8 |
| fi | 4.3 | ml | 1.2 | tr | 43.3 |
| fr | 61.5 | mn | 0.3 | ug | 0.1 |
| fy | 0.1 | mr | 0.4 | uk | 11.1 |
| ga | 0.2 | ms | 0.5 | ur | 2.2 |
| gd | 0.1 | my | 0.4 | uz | 0.1 |
| gl | 1.0 | ne | 0.5 | vi | 52.0 |
| gu | 0.2 | nl | 17.8 | yi | 0.2 |
| he | 3.3 | no | 3.8 | zh | 96.0 |

Table 9: Statistics of CC-100 used for ERNIE-M pre-training.

| ISO Code | Size (GB) | ISO Code | Size (GB) |
|---|---|---|---|
| ar | 9.8 | ru | 8.3 |
| bg | 2.2 | sw | 0.1 |
| de | 10.7 | th | 3.3 |
| el | 4.0 | tr | 1.1 |
| es | 8.8 | ur | 0.7 |
| fr | 13.7 | vi | 0.8 |
| hi | 0.3 | zh | 5.0 |

Table 10: Statistics of parallel data used for ERNIE-M pre-training.

## A.2 Hyperparameters for Pre-training

Table 11 lists the hyperparameters for pre-training. We use the XLM-R model to initialize the parameters of base and large model, for the small model, we train it from scratch. The vocab of ERNIE-M is the same as that of XLM-R.

| Hyperparameters | SMALL | BASE | LARGE |
|---|---|---|---|
| Layers | 6 | 12 | 24 |
| Hidden size | 768 | 768 | 1024 |
| FFN inner hidden size | 3,072 | 3,072 | 4,096 |
| FFN dropout | 0.1 | 0.1 | 0.1 |
| Attention heads | 12 | 12 | 16 |
| Attention dropout | 0.1 | 0.1 | 0.1 |
| Embedding size | 768 | 768 | 1024 |
| Training steps | 240K | 150K | 200K |
| Batch size | 1,024 | 2,048 | 2,048 |
| Learning rate | 3e-4 | 2e-4 | 1e-4 |
| Learning rate schedule | Linear | Linear | Linear |
| Adam $\varepsilon$ | 1e-6 | 1e-6 | 1e-6 |
| Adam $\beta_1$ | 0.98 | 0.98 | 0.98 |
| Adam $\beta_2$ | 0.999 | 0.999 | 0.999 |
| Weight decay | 0.01 | 0.01 | 0.01 |
| Warmup steps | 10,000 | 10,000 | 10,000 |

Table 11: Hyperparameters used for pre-training.

## A.3 Hyperparameters for Fine-tuning

Tables 12 and 13 list the fine-tuning parameters on XNLI, MLQA, CoNLL and PAWS-X. For each task, we select the model with the best performance on the validation set, and the test set score is the average of five runs with different random seeds. Tables 14 list the fine-tuning parameters on Tatoeba.

## A.4 Results for 15 languages model

To better evaluate the performance of ERNIE-M, we train the ERNIE-M-15 model for 15 languages. The languages of training corpora is the same as that of HICTL (Wei et al., 2020). We evaluate ERNIE-M-15 on the XNLI dataset. Table 15 shows the results of 15 languages models. The ERNIE-M-15 model outperforms the current best 15-language cross-lingual model on the XNLI task, achieving a score of 77.5 in the cross-lingual transfer setting, outperforming HICTL 0.2 and a score of 80.7 in the translate-train-all setting, outperforming HICTL 0.7.

## A.5 Results for Cross-lingual Retrieval

Table 16 shows the details of accuracy on each language in the cross-lingual retrieval task. For a fair comparison with VECO, we use the averaged representation in the middle layer of best XNLI model for cross-lingual retrieval task. ERNIE-M outperforms VECO in most languages and achieves state-of-the-art results. We also proposed a new method for cross-lingual retrieval. We use hardest negative binary cross-entropy loss (Wang et al., 2019b; Faghri et al., 2017) to fine-tune ERNIE-M with the same bilingual corpora in pre-training. Table 16 report the results after fine-tuning, the average accuracy of Tatoeba improve from 87.9 to 93.3.

| **Hyperparameters** | **XNLI** | **XNLI*** | **MLQA** | **CoNLL** | **CoNLL*** |
|---|---|---|---|---|---|
| Batch size | 32 | 128 | 32 | 8 | 8 |
| Learning rate | 5e-5 | 5e-5 | 3e-4 | 4e-4 | 3e-4 |
| Layerwise LR decay | 0.8 | 0.8 | 0.8 | 0.8 | 0.8 |
| LR schedule | Linear | Linear | Linear | Linear | Linear |
| Warmup faction | 10% | 10% | 10% | 10% | 10% |
| Weight decay | 0 | 0 | 0 | 0.01 | 0.01 |
| Epoch | 5 | 2 | 2 | 10 | 10 |

Table 12: Hyperparameters used for ERNIE-M$_{\text{SMALL}}$ and ERNIE-M$_{\text{BASE}}$ fine-tuning; parameters with "*" are in the translate-train-all setting, and those without "*" are in the cross-lingual setting.

| **Hyperparameters** | **XNLI** | **XNLI*** | **MLQA** | **CoNLL** | **CoNLL*** | **PAWS-X** | **PAWS-X*** |
|---|---|---|---|---|---|---|---|
| Batch size | 32 | 128 | 32 | 8 | 8 | 64 | 64 |
| Learning rate | 5e-5 | 5e-5 | 8e-5 | 4e-4 | 3e-4 | 5e-5 | 7e-5 |
| Layerwise LR decay | 0.8 | 0.8 | 0.9 | 0.8 | 0.8 | 0.9 | 0.9 |
| LR schedule | Linear | Linear | Linear | Linear | Linear | Linear | Linear |
| Warmup faction | 10% | 10% | 10% | 10% | 10% | 10% | 10% |
| Weight decay | 0 | 0 | 0 | 0.01 | 0.01 | 0.01 | 0.01 |
| Epoch | 5 | 1 | 2 | 10 | 10 | 10 | 2 |

Table 13: Hyperparameters used for ERNIE-M$_{\text{LARGE}}$ fine-tuning; parameters with "*" are in the translate-train-all setting, and those without "*" are in the cross-lingual setting.

| **Hyperparameters** | **LARGE** |
|---|---|
| Training steps | 200K |
| Batch size | 32 |
| Learning rate | 5e-5 |
| Learning rate schedule | Linear |
| Weight decay | 0.0 |
| Warmup faction | 10% |

Table 14: Hyperparameters used for ERNIE-M$_{\text{LARGE}}$ fine-tuneing in Tatoeba.

| **Model** | **en** | **fr** | **es** | **de** | **el** | **bg** | **ru** | **tr** | **ar** | **vi** | **th** | **zh** | **hi** | **sw** | **ur** | **Avg** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Fine-tune cross-lingual model on English training set (Cross-lingual Transfer)* | | | | | | | | | | | | | | | | |
| XLM (Lample and Conneau, 2019) | 85.0 | 78.7 | 78.9 | 77.8 | 76.6 | 77.4 | 75.3 | 72.5 | 73.1 | 76.1 | 73.2 | 76.5 | 69.6 | 68.4 | 67.3 | 75.1 |
| HICTL (Wei et al., 2020) | **86.3** | 80.5 | 81.3 | 79.5 | 78.9 | 80.6 | **79.0** | 75.4 | 74.8 | 77.4 | 75.7 | **77.6** | **73.1** | **69.9** | **69.7** | 77.3 |
| ERNIE-M-15 | 85.9 | **80.5** | **81.3** | **79.8** | **79.3** | **80.7** | 78.7 | **76.8** | **76.8** | **78.0** | **76.1** | 77.4 | 72.9 | 68.9 | 68.9 | **77.5** |
| *Fine-tune cross-lingual model on all training sets (Translate-Train-All)* | | | | | | | | | | | | | | | | |
| XLM (Lample and Conneau, 2019) | 85.0 | 80.8 | 81.3 | 80.3 | 79.1 | 80.9 | 78.3 | 75.6 | 77.6 | 78.5 | 76.0 | 79.5 | 72.9 | 72.8 | 68.5 | 77.8 |
| HICTL (Wei et al., 2020) | **86.5** | 82.3 | 83.2 | 80.8 | 81.6 | 82.2 | **81.3** | 80.5 | 78.1 | 80.4 | 78.6 | **80.7** | 76.7 | 73.8 | **73.9** | 80.0 |
| ERNIE-M-15 | 86.4 | **82.4** | **83.5** | **82.7** | **83.1** | **83.2** | 81.0 | **80.6** | **80.5** | **80.9** | **79.2** | 80.6 | 77.7 | **75.8** | 72.8 | **80.7** |

Table 15: Evaluation results on XNLI cross-lingual natural language inference for 15 languages model.

| **Model** | **af** | **ar** | **bg** | **bn** | **de** | **el** | **es** | **et** | **eu** | **fa** | **fi** | **fr** | **he** | **hi** | **hu** | **id** | **it** | **ja** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VECO$_{\text{LARGE}}$ (Luo et al., 2020) | 80.9 | 85.1 | 91.3 | 78.1 | 98.5 | 89.5 | **97.4** | **94.8** | **79.8** | **93.1** | **95.4** | 93.7 | 85.8 | 94.2 | **93.8** | **93.0** | **92.2** | **92.8** |
| ERNIE-M$_{\text{LARGE}}$ | **88.6** | **88.9** | **92.2** | **84.8** | **98.8** | **92.4** | 96.3 | 82.4 | 78.8 | 92.5 | 92.2 | **94.0** | **86.2** | **95.4** | 88.7 | 91.5 | 90.3 | 86.7 |
| ERNIE-M$^{\dagger}_{\text{LARGE}}$ | 92.6 | 94.3 | 96.6 | 89.2 | 99.7 | 96.8 | 98.8 | 92.5 | 87.4 | 96.0 | 97.1 | 96.5 | 90.1 | 97.9 | 95.5 | 95.7 | 95.2 | 96.9 |
| **Model** | **jv** | **ka** | **kk** | **ko** | **ml** | **mr** | **nl** | **pt** | **ru** | **sw** | **ta** | **te** | **th** | **tl** | **tr** | **ur** | **vi** | **zh** |
| VECO$_{\text{LARGE}}$ (Luo et al., 2020) | 35.1 | 83.0 | 74.1 | **88.7** | 94.8 | 82.5 | 95.9 | **94.6** | 92.2 | **69.7** | 82.4 | 91.0 | 94.7 | 73.0 | 95.2 | 63.8 | **95.1** | **93.9** |
| ERNIE-M$_{\text{LARGE}}$ | **48.0** | **84.2** | **78.2** | 85.3 | **95.2** | **87.6** | **96.1** | 92.6 | **93.1** | 59.4 | **86.9** | **94.0** | **95.6** | **75.4** | **96.3** | **90.8** | 94.5 | 91.7 |
| ERNIE-M$^{\dagger}_{\text{LARGE}}$ | 65.2 | 94.9 | 88.0 | 94.1 | 98.5 | 90.8 | 98.1 | 94.5 | 95.7 | 68.4 | 91.8 | 97.9 | 98.4 | 86.0 | 98.3 | 94.9 | 98.1 | 96.7 |

Table 16: Tatoeba results for each language. "†" indicates the results after fine-tuning