Improving Pretrained Cross-Lingual Language Models via Self-Labeled Word Alignment

Zewen Chi[†][*], **Li Dong**[‡], **Bo Zheng**[‡][*], **Shaohan Huang**[‡]
Xian-Ling Mao[†], **Heyan Huang**[†], **Furu Wei**[‡]
[†]Beijing Institute of Technology
[‡]Microsoft Research
{czw,maoxl,hhy63}@bit.edu.cn
{lidong1,v-zhebo,shaohanh,fuwei}@microsoft.com

Abstract

The cross-lingual language models are typically pretrained with masked language modeling on multilingual text or parallel sentences. In this paper, we introduce denoising word alignment as a new cross-lingual pre-training task. Specifically, the model first self-labels word alignments for parallel sentences. Then we randomly mask tokens in a bitext pair. Given a masked token, the model uses a pointer network to predict the aligned token in the other language. We alternately perform the above two steps in an expectation-maximization manner. Experimental results show that our method improves cross-lingual transferability on various datasets, especially on the token-level tasks, such as question answering, and structured prediction. Moreover, the model can serve as a pretrained word aligner, which achieves reasonably low error rates on the alignment benchmarks. The code and pretrained parameters are available at github.com/CZWin32768/XLM-Align.

1 Introduction

Despite the current advances in NLP, most applications and resources are still English-centric, making non-English users hard to access. Therefore, it is essential to build cross-lingual transferable models that can learn from the training data in high-resource languages and generalize on low-resource languages. Recently, pretrained cross-lingual language models have shown their effectiveness for cross-lingual transfer. By pre-training on monolingual text and parallel sentences, the models provide significant improvements on a wide range of cross-lingual end tasks (Conneau and Lample, 2019; Conneau et al., 2020; Liu et al., 2020; Chi et al., 2021b).

Cross-lingual language model pre-training is typically achieved by learning various pretext tasks on monolingual and parallel corpora. By simply learning masked language modeling (MLM; Devlin et al. 2019) on monolingual text of multiple languages, the models surprisingly achieve competitive results on cross-lingual tasks (Wu and Dredze, 2019; K et al., 2020). Besides, several pretext tasks are proposed to utilize parallel corpora to learn better sentence-level cross-lingual representations (Conneau and Lample, 2019; Chi et al., 2021b; Hu et al., 2020a). For example, the translation language modeling (TLM; Conneau and Lample 2019) task performs MLM on the concatenated parallel sentences, which implicitly enhances cross-lingual transferability. However, most pretext tasks either learn alignment at the sentence level or implicitly encourage cross-lingual alignment, leaving explicit fine-grained alignment task not fully explored.

In this paper, we introduce a new cross-lingual pre-training task, named as *denoising word alignment*. Rather than relying on external word aligners trained on parallel corpora (Cao et al., 2020; Zhao et al., 2020; Wu and Dredze, 2020), we utilize self-labeled alignments in our task. During pre-training, we alternately self-label word alignments and conduct the denoising word alignment task in an expectation-maximization manner. Specifically, the model first self-labels word alignments for a translation pair. Then we randomly mask tokens in the bitext sentence, which is used as the perturbed input for denosing word alignment. For each masked token, the model learns a pointer network to predict the self-labeled alignments in the other language. We repeat the above two steps to iteratively boost the bitext alignment knowledge for cross-lingual pre-training.

We conduct extensive experiments on a wide range of cross-lingual understanding tasks. Experimental results show that our model outperforms the baseline models on various datasets, particularly on the token-level tasks such as question answer-

---

[*]Contribution during internship at Microsoft Research.

3418

ing and structured prediction. Moreover, our model can also serve as a multilingual word aligner, which achieves reasonable low error rates on the bitext alignment benchmarks.

Our contributions are summarized as follows:

- We present a cross-lingual pre-training paradigm that alternately self-labels and predicts word alignments.

- We introduce a pre-training task, denoising word alignment, which predicts word alignments from perturbed translation pairs.

- We propose a word alignment algorithm that formulates the word alignment problem as optimal transport.

- We demonstrate that our explicit alignment objective is effective for cross-lingual transfer.

2 Related Work

Cross-lingual LM pre-training   Pretrained with masked language modeling (MLM; Devlin et al. 2019) on monolingual text, multilingual BERT (mBERT; Devlin et al. 2019) and XLM-R (Conneau et al., 2020) produce promising results on cross-lingual transfer benchmarks (Hu et al., 2020b). mT5 (Xue et al., 2020) learns a multilingual version of T5 (Raffel et al., 2020) with text-to-text tasks. In addition to monolingual text, several methods utilize parallel corpora to improve cross-lingual transferability. XLM (Conneau and Lample, 2019) presents the translation language modeling (TLM) task that performs MLM on concatenated translation pairs. ALM (Yang et al., 2020) introduces code-switched sequences into cross-lingual LM pre-training. Unicoder (Huang et al., 2019) employs three cross-lingual tasks to learn mappings among languages. From an information-theoretic perspective, InfoXLM (Chi et al., 2021b) proposes the cross-lingual contrastive learning task to align sentence-level representations. Additionally, AM-BER (Hu et al., 2020a) introduces an alignment objective that minimizes the distance between the forward and backward attention matrices. More recently, Ernie-M (Ouyang et al., 2020) presents the back-translation masked language modeling task that generates pseudo parallel sentence pairs for learning TLM, which provides better utilization of monolingual corpus. VECO (Luo et al., 2020) pretrains a unified cross-lingual language model for both NLU and NLG. mT6 (Chi et al., 2021a)

improves the multilingual text-to-text transformer with translation pairs.

Notably, Word-aligned BERT models (Cao et al., 2020; Zhao et al., 2020) finetune mBERT by an explicit alignment objective that minimizes the distance between aligned tokens. Wu and Dredze (2020) exploit contrastive learning to improve the explicit alignment objectives. However, Wu and Dredze (2020) show that these explicit alignment objectives do not improve cross-lingual representations under a more extensive evaluation. Moreover, these models are restricted to stay close to their original pretrained values, which is not applicable for large-scale pre-training. On the contrary, we demonstrate that employing our explicit alignment objective in large-scale pre-training can provide consistent improvements over baseline models.

Word alignment   The IBM models (Brown et al., 1993) are statistical models for modeling the translation process that can extract word alignments between sentence pairs. A large number of word alignment models are based on the IBM models (Och and Ney, 2003; Mermer and Saraçlar, 2011; Dyer et al., 2013; Östling and Tiedemann, 2016). Recent studies have shown that word alignments can be extracted from neural machine translation models (Ghader and Monz, 2017; Koehn and Knowles, 2017; Li et al., 2019) or from pretrained cross-lingual LMs (Jalili Sabet et al., 2020; Nagata et al., 2020).

3 Method

Figure 1 illustrates an overview of our method for pre-training our cross-lingual LM, which is called XLM-ALIGN. XLM-ALIGN is pretrained in an expectation-maximization manner with two alternating steps, which are word alignment self-labeling and denoising word alignment. We first formulate word alignment as an optimal transport problem, and self-label word alignments of the input translation pair on-the-fly. Then, we update the model parameters with the denoising word alignment task, where the model uses a pointer network (Vinyals et al., 2015) to predict the aligned tokens from the perturbed translation pair.

3.1 Word Alignment Self-Labeling

The goal of word alignment self-labeling is to estimate the word alignments of the input translation pair on-the-fly, given the current XLM-ALIGN model. Given a source sentence

(a) Word alignment self-labeling
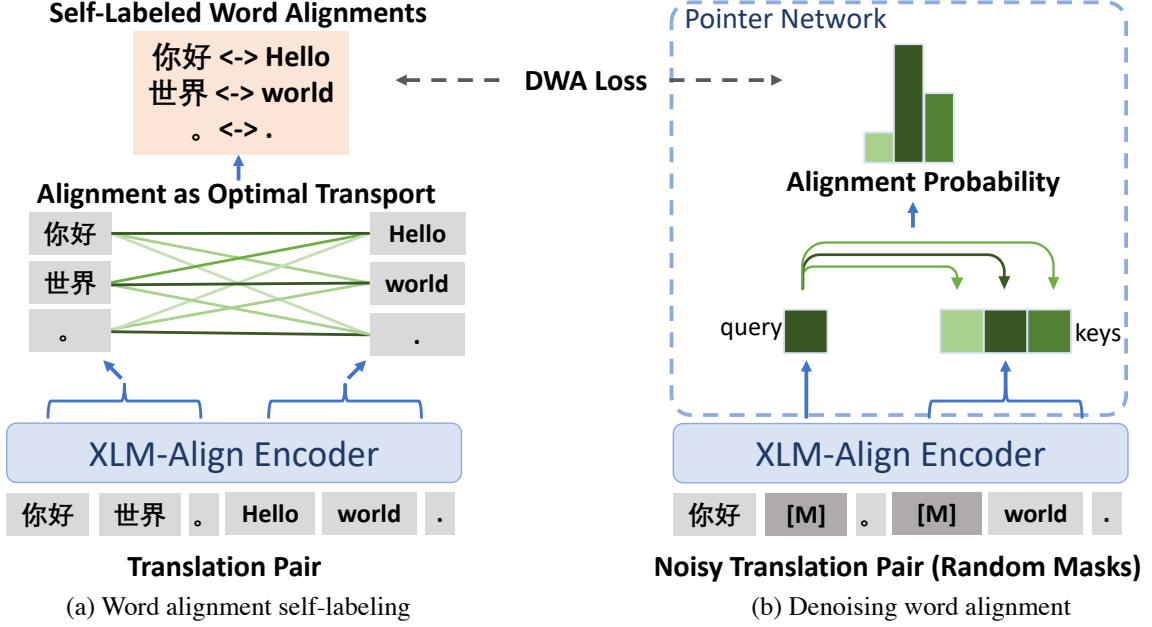(b) Denoising word alignment

Figure 1: An overview of our method. XLM-ALIGN is pretrained in an expectation-maximization manner with two alternating steps. (a) Word alignment self-labeling: we formulate word alignment as an optimal transport problem, and self-labels word alignments of the input translation pair on-the-fly; (b) Denoising word alignment: we update the model parameters with the denoising word alignment task, where the model uses a pointer network to predict the aligned tokens from the perturbed translation pair.

$\mathcal{S} = s_1 \dots s_i \dots s_n$ and a target sentence $\mathcal{T} = t_1 \dots t_j \dots t_m$, we model the word alignment between $\mathcal{S}$ and $\mathcal{T}$ as a doubly stochastic matrix $A \in \mathbb{R}_+^{n \times m}$ such that the rows and the columns all sum to 1, where A_{ij} stands for the probability of the alignment between s_i and t_j. The rows and the columns of A represent probability distributions of the forward alignment and the backward alignment, respectively. To measure the similarity between two tokens from $\mathcal{S}$ and $\mathcal{T}$, we define a metric function f_{sim} by using cross-lingual representations produced by XLM-ALIGN:

$$f_{\text{sim}}(s_i, t_j) = -\log \max(\epsilon, \boldsymbol{h}_i^\top \boldsymbol{h}_j) \quad (1)$$

where ϵ is a constant to avoid negative values in the log function, and $\boldsymbol{h}_i$ is the hidden vector of the i-th token by encoding the concatenated sequence of $\mathcal{S}$ and $\mathcal{T}$ with XLM-ALIGN. Empirically, the metric function produces a high similarity score if the two input tokens are semantically similar.

The word alignment problem is formulated as finding A that maximizes the sentence similarity between $\mathcal{S}$ and $\mathcal{T}$:

$$\max_A \sum_{i=1}^n \sum_{j=1}^m A_{ij} f_{\text{sim}}(s_i, t_j) \quad (2)$$

We can find that Eq. (2) is identical to the regularized optimal transport problem (Peyré et al., 2019),

if we add an entropic regularization to A:

$$\max_A \sum_{i=1}^n \sum_{j=1}^m A_{ij} f_{\text{sim}}(s_i, t_j) - \mu A_{ij} \log A_{ij} \quad (3)$$

Eq. (3) has a unique solution A^* such that

$$A^* = \text{diag}(\boldsymbol{u}) K \text{diag}(\boldsymbol{v}) \quad (4)$$

$$K_{ij} = e^{f_{\text{sim}}(s_i, t_j)/\mu} \quad (5)$$

where $\boldsymbol{u} \in \mathbb{R}_+^n, \boldsymbol{v} \in \mathbb{R}_+^m, K \in \mathbb{R}_+^{n \times m}$. According to Sinkhorn's algorithm (Peyré et al., 2019), the variables $\boldsymbol{u}$ and $\boldsymbol{v}$ can be calculated by the following iterations:

$$\boldsymbol{u}^{t+1} = \frac{\mathbf{1}_n}{K \boldsymbol{v}^t}, \quad \boldsymbol{v}^{t+1} = \frac{\mathbf{1}_m}{K^\top \boldsymbol{u}^{t+1}} \quad (6)$$

where $\boldsymbol{v}^t$ can be initialized by $\boldsymbol{v}^{t=0} = \mathbf{1}_m$.

With the solved stochastic matrix A^*, we can produce the forward word alignments $\overrightarrow{\mathcal{A}}$ by applying argmax over rows:

$$\overrightarrow{\mathcal{A}} = \{(i, j) \mid j = \arg \max_k A_{ik}^*\} \quad (7)$$

Similarly, the backward word alignments $\overleftarrow{\mathcal{A}}$ can be computed by applying argmax over columns. To obtain high-precision alignment labels, we adopt an iterative alignment filtering operation. We initialize

3420

the alignment labels $\mathcal{A}$ as $\emptyset$. In each iteration, we follow the procedure of Itermax (Jalili Sabet et al., 2020) that first computes $\overrightarrow{\mathcal{A}}$ and $\overleftarrow{\mathcal{A}}$ by Eq. (7). Then, the alignment labels are updated by:

$$\mathcal{A} \leftarrow \mathcal{A} \cup (\overrightarrow{\mathcal{A}} \cap \overleftarrow{\mathcal{A}}) \qquad (8)$$

Finally, A^* is updated by:

$$A^*_{ij} \leftarrow \begin{cases} 0, & (i,j) \in \mathcal{A} \\ \alpha A^*_{ij}, & \exists k \ (i,k) \in \mathcal{A} \vee (k,j) \in \mathcal{A} \\ A^*_{ij}, & \text{others} \end{cases}$$

$$(9)$$

where α is a discount factor. After several iterations, we obtain the final self-labeled word alignments $\mathcal{A}$.

3.2 Denoising Word Alignment

After self-labeling word alignments, we update the model parameters with the denoising word alignment (DWA) task. The goal of DWA is to predict the word alignments from the perturbed version of the input translation pair.

Consider the perturbed version of the input translation pair $(\mathcal{S}^*, \mathcal{T}^*)$ constructed by randomly replacing the tokens with masks. We first encode the translation pair into hidden vectors $\boldsymbol{h}^*$ with the XLM-ALIGN encoder:

$$\boldsymbol{h}^*_1 \ldots \boldsymbol{h}^*_{n+m} = \text{encoder}([\mathcal{S}^*, \mathcal{T}^*]) \qquad (10)$$

where $[\mathcal{S}^*, \mathcal{T}^*]$ is the concatenated sequence of $\mathcal{S}^*$ and $\mathcal{T}^*$ with the length of $n + m$. Then, we build a pointer network upon the XLM-ALIGN encoder that predicts the word alignments. Specifically, for the i-th source token, we use $\boldsymbol{h}^*_i$ as the query vector and $\boldsymbol{h}^*_{n+1}, \ldots, \boldsymbol{h}^*_{n+m}$ as the key vectors. Given the query and key vectors, the forward alignment probability $\boldsymbol{a}_i$ is computed by the scaled dot-product attention (Vaswani et al., 2017):

$$\boldsymbol{a}_i = \text{softmax}(\frac{\boldsymbol{q}_i^\top K}{\sqrt{d_h}}) \qquad (11)$$

$$\boldsymbol{q}_i = \text{linear}(\boldsymbol{h}^*_i) \qquad (12)$$

$$K = \text{linear}([\boldsymbol{h}^*_{n+1} \ldots \boldsymbol{h}^*_{n+m}]) \qquad (13)$$

where d_h is the dimension of the hidden vectors. Similarly, the backward alignment probability can be computed by above equations if we use target tokens as the query vectors and $\boldsymbol{h}^*_1 \ldots \boldsymbol{h}^*_n$ as key vectors. Notice that we only consider the self-labeled

and masked positions as queries. Formally, we use the following query positions in the pointer network:

$$\mathcal{P} = \{i | (i, \cdot) \in \mathcal{A} \vee (\cdot, i) \in \mathcal{A}\} \cap \mathcal{M} \qquad (14)$$

where $\mathcal{M}$ is the set of masked positions. The training objective is to minimize the cross-entropy between the alignment probabilities and the self-labeled word alignments:

$$\mathcal{L}_{\text{DWA}} = \sum_{i \in \mathcal{P}} \text{CE}(\boldsymbol{a}_i, \mathcal{A}(i)) \qquad (15)$$

where $\text{CE}(\cdot, \cdot)$ stands for the cross-entropy loss, and $\mathcal{A}(i)$ is the self-labeled aligned position of the i-th token.

Algorithm 1 Pre-training XLM-ALIGN

Input: Multilingual corpus $\mathcal{D}_\text{m}$, parallel corpus $\mathcal{D}_\text{p}$, learning rate τ
Output: XLM-ALIGN parameters $\boldsymbol{\theta}$
1: Initialize $\boldsymbol{\theta}$ with cold-start pre-training
2: **while** not converged **do**
3: $\quad \mathcal{X} \sim \mathcal{D}_\text{m}, \ (\mathcal{S}, \mathcal{T}) \sim \mathcal{D}_\text{p}$
4: $\quad \mathcal{A} \leftarrow f_\text{self-labeling}(\mathcal{S}, \mathcal{T}; \boldsymbol{\theta})$
5: $\quad \boldsymbol{g} \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}_\text{MLM}(\mathcal{X}) + \nabla_{\boldsymbol{\theta}} \mathcal{L}_\text{TLM}(\mathcal{S}, \mathcal{T}) + \nabla_{\boldsymbol{\theta}} \mathcal{L}_\text{DWA}(\mathcal{S}, \mathcal{T}, \mathcal{A})$
6: $\quad \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \tau \boldsymbol{g}$

3.3 Pre-training XLM-ALIGN

We illustrate the pre-training procedure of XLM-ALIGN in Algorithm 1. In addition to DWA, we also include MLM and TLM for pre-training XLM-ALIGN, which implicitly encourage the cross-lingual alignment. The overall loss function is defined as:

$$\mathcal{L}_\text{MLM}(\mathcal{X}) + \mathcal{L}_\text{TLM}(\mathcal{S}, \mathcal{T}) + \mathcal{L}_\text{DWA}(\mathcal{S}, \mathcal{T}, \mathcal{A})$$

In each iteration, we first sample monolingual text $\mathcal{X}$, and parallel text $(\mathcal{S}, \mathcal{T})$. Then, we self-label word alignments and update the model parameters by learning pretext tasks. Notice that the model parameters are initialized by a cold-start pre-training to avoid producing low-quality alignment labels. The cold-start pre-training can be accomplished by using a pretrained LM as the model initialization.

4 Experiments

4.1 Pre-training

Following previous cross-lingual pretrained models (Conneau and Lample, 2019; Conneau et al.,

Model	Structured Prediction		Question Answering			Sentence Classification		Avg
	POS	NER	XQuAD	MLQA	TyDiQA	XNLI	PAWS-X	
Metrics	F1	F1	F1 / EM	F1 / EM	F1 / EM	Acc.	Acc.	
MBERT*	70.3	62.2	64.5 / 49.4	61.4 / 44.2	59.7 / 43.9	65.4	81.9	63.1
XLM*	70.1	61.2	59.8 / 44.3	48.5 / 32.6	43.6 / 29.1	69.1	80.9	58.6
MT5$_{base}$	-	56.6	67.0 / 49.0	64.6 / 45.0	58.1 / 42.8	75.4	**87.4**	-
XLM-R$_{base}$	75.6	61.8	71.9 / 56.4	65.1 / 47.2	55.4 / 38.3	75.0	84.9	66.4
XLM-ALIGN	**76.0**	**63.7**	**74.7 / 59.0**	**68.1 / 49.8**	**62.1 / 44.8**	**76.2**	86.8	**68.9**

Table 1: Evaluation results on XTREME structured prediction, question answering, and sentence classification tasks. We adopt 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-ALIGN and XLM-R$_{base}$ are averaged over five runs.

2020; Chi et al., 2021b), we use raw sentences from the Wikipedia dump and CCNet (Wenzek et al., 2019) for MLM, including 94 languages. For TLM and DWA, we use parallel corpora from MultiUN (Ziemski et al., 2016), IIT Bombay (Kunchukuttan et al., 2018), OPUS (Tiedemann, 2012), and WikiMatrix (Schwenk et al., 2019), including 14 English-centric language pairs. We pretrain a Transformer with 12 layers and the hidden size of 768, where the parameters are initialized with XLM-R (Conneau et al., 2020). The model is optimized with the Adam optimizer (Kingma and Ba, 2015) for 150K steps with batch size of $2,048$. Notice that TLM and DWA share the same forward procedure for encoding the perturbed sentence pair. The pre-training of XLM-ALIGN takes about six days with two Nvidia DGX-2 stations. More details of the training data and the hyperparameters are in supplementary document.

4.2 XTREME Benchmark

XTREME is a multilingual benchmark for evaluating cross-lingual generalization. We evaluate our model on 7 cross-lingual downstream tasks included by XTREME, which can be grouped into 3 categories: (1) Structured prediction: part-of-speech tagging on the Universal Dependencies v2.5 (Zeman et al., 2019), and named entity recognition on the WikiAnn (Pan et al., 2017; Rahimi et al., 2019) dataset; (2) Question answering: cross-lingual question answering on MLQA (Lewis et al., 2020) and XQuAD (Artetxe et al., 2020), and gold passage of typologically diverse question answering (TyDiQA-GoldP; Clark et al. 2020); (3) Sentence classification: cross-lingual natural language inference (XNLI; Conneau et al. 2018), and cross-lingual paraphrase adversaries from word scrambling (PAWS-X; Yang et al. 2019).

Baselines We use the following pretrained cross-lingual LMs as baselines. (1) Multilingual BERT (MBERT; Devlin et al. 2019) is pretrained with masked language modeling (MLM) and next sentence prediction on Wikipedia of 104 languages; (2) XLM (Conneau and Lample, 2019) is jointly pretrained with MLM on 100 languages and translation language modeling (TLM) on 14 language pairs; (3) MT5 (Xue et al., 2020) is the multilingual version of T5 pretrained with text-to-text tasks; (4) XLM-R (Conneau et al., 2020) is pretrained with MLM on large-scale CC-100 dataset with long training steps.

Fine-tuning Following Hu et al. (2020b), we adopt the zero-shot transfer setting for evaluation, where the models are only fine-tuned on English training data but evaluated on all target languages. Besides, we only use one model for evaluation on all target languages, rather than selecting different models for each language. The detailed fine-tuning hyperparameters can be found in supplementary document.

Results In Table 1, we present the evaluation results on XTREME structured prediction, question answering, and sentence classification tasks. It can be observed that our XLM-ALIGN obtains the best average score over all the baseline models, improving the previous score from 66.4 to 68.9. It demonstrates that our model learns more transferable representations for the cross-lingual tasks, which is beneficial for building more accessible multilingual NLP applications. It is worth mentioning that our method brings noticeable improvements on the question answering and the structured prediction tasks. Compared with XLM-R$_{base}$, XLM-ALIGN provides 6.7% and 1.9% F1 improvements on TyDiQA and NER. The improvements show that the

Alignment Method	Pretrained Model	Alignment Error Rate ↓				Avg
		en-de	en-fr	en-hi	en-ro	
fast_align (Dyer et al., 2013)	-	32.14	19.46	59.90	-	-
SimAlign - Argmax (Jalili Sabet et al., 2020)	XLM-R	19.	7.	39.	29.	24.
SimAlign - Itermax (Jalili Sabet et al., 2020)	XLM-R	20.	9.	39.	28.	24.
SimAlign - Itermax (reimplementation)	XLM-R	20.15	10.05	38.72	27.41	24.08
Ours - Optimal Transport (Section 3.1)	XLM-R	17.74	7.54	37.79	27.49	22.64
SimAlign (reimplementation)	XLM-ALIGN	18.93	10.33	**33.84**	27.09	22.55
Ours - Optimal Transport (Section 3.1)	XLM-ALIGN	**16.63**	**6.61**	33.98	**26.97**	**21.05**

Table 2: Evaluation results for word alignment on four English-centric language pairs. We report the alignment error rate scores (lower is better). For both SimAlign (Jalili Sabet et al., 2020) and our optimal-transport alignment method, we use the hidden vectors from the 8-th layer produced by XLM-R$_{base}$ or XLM-ALIGN. "(reimplementation)" is our reimplementation of SimAlign-Itermax.
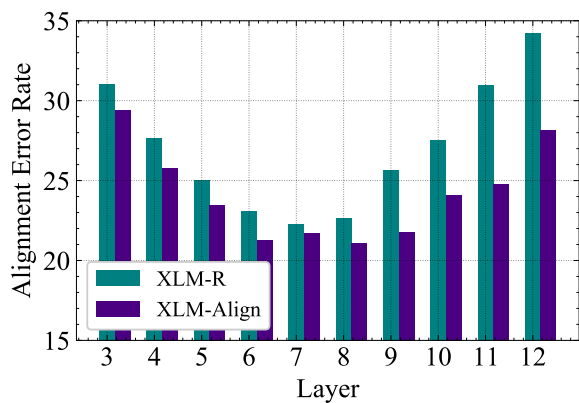

Figure 2: Evaluation results on word alignment across different layers. We illustrate the averaged AER scores on the test sets of four language pairs. The results of the first two layers are not included due to the high AER.

pretrained XLM-ALIGN benefits from the explicit word alignment objective, particularly on the structured prediction and question answering tasks that require token-level cross-lingual transfer. In terms of sentence classification tasks, XLM-ALIGN also consistently outperforms XLM-R$_{base}$.

4.3 Word Alignment

Word alignment is the task of finding corresponding word pairs in a parallel sentence. We conduct evaluations with golden alignments of four language pairs from EuroParl[1], WPT2003[2], and WPT2005[3], containing 1,244 annotated sentence pairs in total. We use alignment error rate (AER; Och and Ney

---

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

2003) as the evaluation metrics.

Results We first explore whether our word alignment self-labeling method is effective for generating high-quality alignment labels. Thus, we compare our method with (1) fast_align (Dyer et al., 2013), a widely-used implementation of IBM Model 2 (Och and Ney, 2003); (2) SimAlign (Jalili Sabet et al., 2020), state-of-the-art unsupervised word alignment method. For a fair comparison, we use the same pretrained LM and hidden layer as in SimAlign to produce sentence representations. In specific, we take the hidden vectors from the 8-th layer of XLM-R$_{base}$ or XLM-ALIGN, and obtain the alignments following the procedure as described in Section 3.1. Since the produced alignments are subword-level, we convert the alignments into word-level by the following rule that "if two subwords are aligned, the words they belong to are also aligned".

As shown in Table 2, we report the AER scores on the four language pairs. It can be observed that our optimal-transport method outperforms fast_align and SimAlign, demonstrating that our method can produce high-quality alignment labels, which is helpful for the DWA task. Moreover, our method consistently outperforms SimAlign when using hidden vectors from both XLM-R$_{base}$ and XLM-ALIGN.

Then, we compare our XLM-ALIGN with XLM-R$_{base}$ on the word alignment task. Empirically, a lower AER indicates that the model learns better cross-lingual representations. From Table 2, XLM-ALIGN obtains the best AER results over all the four language pairs, reducing the averaged AER from 22.64 to 21.05. Besides, un-

Models	XNLI	POS	NER	MLQA	Avg
XLM-R*	74.6	75.7	61.6	65.7	69.4
XLM-ALIGN	**75.2**	75.6	**62.6**	**66.7**	**70.0**
−DWA	75.1	75.2	62.0	65.8	69.5
−TLM	74.4	**76.0**	60.4	66.0	69.2

Table 3: Ablation studies on the components of XLM-ALIGN. XLM-R* stands for continue-training XLM-R$_{base}$ with MLM for fair comparisons. Results are averaged over five runs.

Layer	XNLI	POS	NER	MLQA	Avg
Layer-8	75.1	75.3	61.9	66.7	69.8
Layer-10	75.2	75.6	**62.6**	66.7	70.0
Layer-12	**75.2**	**75.8**	62.3	**67.0**	**70.1**

Table 4: Results of XLM-ALIGN with different layers used for word alignment self-labeling during pre-training. Results are averaged over five runs.

Layer	XNLI	POS	NER	MLQA	Avg
Layer-8	**75.4**	75.3	61.7	66.2	69.7
Layer-10	75.1	75.6	**62.5**	66.3	69.9
Layer-12	75.2	**75.8**	62.3	**67.0**	**70.1**

Table 5: Results of XLM-ALIGN with different layers used for denoising word alignment during pre-training. Results are averaged over five runs.

der both SimAlign and our optimal-transport method, XLM-ALIGN provides consistent reduction of AER, demonstrating the effectiveness of our method for learning fine-grained cross-lingual representations.

We also compare XLM-ALIGN with XLM-R$_{base}$ using the hidden vectors from the 3-th layer to the 12-th layer. We illustrate the averaged AER scores in Figure 2. Notice that the results on the first two layers are not presented in the figure because of the high AER. It can be observed that XLM-ALIGN consistently improves the results over XLM-R$_{base}$ across these layers. Moreover, it shows a parabolic trend across the layers of XLM-R$_{base}$, which is consistent with the results in (Jalili Sabet et al., 2020). In contrast to XLM-R$_{base}$, XLM-ALIGN alleviates this trend and greatly reduces AER in the last few layers. We believe this property of XLM-ALIGN brings better cross-lingual transferability on the end tasks.

5 Analysis

In this section, we conduct comprehensive ablation studies for a better understanding of our XLM-ALIGN. To reduce the computational cost, we reduce the batch size to 256, and pretrain models with 50K steps in the following experiments.

5.1 Ablation Studies

We perform ablation studies to understand the components of XLM-ALIGN, by removing the denoising word alignment loss (−DWA), the TLM loss (−TLM), or removing both (XLM-R*), which is identical to continue-training XLM-R$_{base}$ with MLM. We evaluate the models on XNLI, POS, NER, and MLQA, and present the results in Table 3. Comparing −TLM with −DWA, we find that DWA is more effective for POS and MLQA, while TLM performs better on XNLI and NER. Comparing −TLM with XLM-R*, it shows that directly learning DWA slightly harms the perfor-

mance. However, jointly learning DWA with TLM provides remarkable improvements over −DWA, especially on the question answering and the structure prediction tasks that requires token-level cross-lingual transfer. This indicates that TLM potentially improves the quality of self-labeled word alignments, making DWA more effective for cross-lingual transfer.

5.2 Word Alignment Self-Labeling Layer

It has been shown that the word alignment performance has a parabolic trend across the layers of mBERT and XLM-R (Jalili Sabet et al., 2020). It indicates that the middle layers produce higher-quality word alignments than the bottom and the top layers. To explore which layer produces better alignment labels for pre-training, we pretrain three variants of XLM-ALIGN, where we use the hidden vectors from three different layers for word alignment self-labeling. We use the 8-th, 10-th, and 12-th layers for word alignment self-labeling during the pre-training. We present the evaluation results in Table 4. Surprisingly, although Layer-8 produces higher-quality alignment labels at the beginning of the pre-training, using the alignment labels from the 12-th layer learns a more transferable XLM-ALIGN model for cross-lingual end tasks.

5.3 Denoising Word Alignment Layer

Beyond the self-labeling layer, we also investigate which layer is better for learning the denoising word alignment task. Recent studies have shown

Filtering	XNLI	POS	NER	MLQA	Avg
Enable	**75.2**	**75.6**	**62.6**	**66.7**	**70.0**
Disable	74.2	75.3	61.6	65.3	69.1

Table 6: Effects of alignment filtering in word alignment self-labeling. Results are averaged over five runs.

Position	XNLI	POS	NER	MLQA	Avg
masked	75.2	75.6	**62.6**	**66.7**	**70.0**
unmasked	**75.5**	75.5	62.0	66.5	69.8
all-aligned	75.3	**75.9**	61.6	**66.7**	69.9
no-query	75.1	75.2	62.0	65.8	69.5

Table 7: Effects of the query positions in the pointer network for denoising word alignment. Results are averaged over five runs.

that it is beneficial to learn sentence-level cross-lingual alignment at a middle layer (Chi et al., 2021b). Therefore, we pretrain XLM-ALIGN models by using three different layers for DWA, that is, using the hidden vectors of middle layers as the input of the pointer network. We compare the evaluation results of the three models in Table 5. It can be found that learning DWA at Layer-8 improves XNLI while learning DWA at higher layers produces better performance on the other three tasks. It suggests that, compared with sentence-level pretext tasks that prefers middle layers, the DWA task should be applied at top layers.

5.4 Effects of Alignment Filtering

Although our self-labeling method produces high-quality alignment labels, the alignment filtering operation can potentially make some of the tokens unaligned, which reduces the example efficiency. Thus, we explore whether the alignment filtering is beneficial for pre-training XLM-ALIGN. To this end, we pretrain an XLM-ALIGN model without alignment filtering. In specific, we use the union set of the forward and backward alignments as the self-labeled alignments so that all tokens are aligned at least once. The forward and backward alignments are obtained by applying the argmax function over rows and columns of A^*, respectively. Empirically, the alignment filtering operation generates high-precision yet fewer labels, while removing the filtering promises more labels but introduces low-confident labels. In Table 6, we compare the results of the models with or without alignment filtering. It can be observed that the alignment filtering operation improves the performance on the end tasks. This demonstrates that it is necessary to use high-precision labels for learning the denoising word alignment task. On the contrary, using perturbed alignment labels in pre-training harms the performance on the end tasks.

5.5 Effects of DWA Query Positions

In the denoising word alignment task, we always use the hidden vectors of the masked positions

as the query vectors in the pointer network. To explore the impact of the DWA query positions, we compare three different query positions in Table 7: (1) *masked*: only using the masked tokens as queries; (2) *unmasked*: randomly using 15% of the unmasked tokens as queries; (3) *all-aligned*: for each self-labeled aligned pair, randomly using one of the two tokens as a query. Also, we include the *no-query* baseline that does not use any queries, which is identical to removing DWA. It can be observed that using all the three query positions improves the performance over the *no-query* baseline. Moreover, using the masked positions as queries achieves better results than the other two positions, demonstrating the effectiveness of the masked query positions.

6 Discussion

In this paper, we introduce denoising word alignment as a new cross-lingual pre-training task. By alternately self-labeling and predicting word alignments, our XLM-ALIGN model learns transferable cross-lingual representations. Experimental results show that our method improves the cross-lingual transferability on a wide range of tasks, particularly on the token-level tasks such as question answering and structured prediction.

Despite the effectiveness for learning cross-lingual transferable representations, our method also has the limitation that requires a cold-start pre-training to prevent the model from producing low-quality alignment labels. In our experiments, we also try to pretrain XLM-ALIGN from scratch, i.e., without cold-start pre-training. However, the DWA task does not work very well due to the low-quality of self-labeled alignments. Thus, we recommend continue-training XLM-ALIGN on the basis of other pretrained cross-lingual language models. For future work, we would like to research on removing this restriction so that the model can learn word alignments from scratch.

7 Ethical Considerations

Despite the current advances in NLP, most NLP research works and applications are English-centric, making none-English users hard to access to NLP-related services. Our method aims to pretrain cross-lingual language models that transfer supervision signals from high-resource languages to low-resource languages, which makes the NLP services and applications more accessible for low-resource-language speakers. Furthermore, our method can build multilingual models that serve on different languages at the same time, reducing the computational resources for building multilingual models separately for each language.

Acknowledgements

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.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

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

Zewen Chi, Li Dong, Shuming Ma, Shaohan Huang Xian-Ling Mao, Heyan Huang, and Furu Wei. 2021a. mt6: Multilingual pretrained text-to-text transformer with translation pairs. *arXiv preprint arXiv:2104.08692*.

Zewen Chi, Li Dong, Furu Wei, 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.

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

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.

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.

Hamidreza Ghader and Christof Monz. 2017. What does attention in neural machine translation pay attention to? In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 30–39, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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

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

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pretraining with multiple cross-lingual tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2485–2494, Hong Kong, China. Association for Computational Linguistics.

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

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.

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

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, Miyazaki, Japan. European Language Resources Association.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303.

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*.

Coşkun Mermer and Murat Saraçlar. 2011. Bayesian word alignment for statistical machine translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 182–187.

Masaaki Nagata, Katsuki Chousa, and Masaaki Nishino. 2020. A supervised word alignment method based on cross-language span prediction using multilingual BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 555–565, Online. Association for Computational Linguistics.

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

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with markov chain monte carlo. *The Prague Bulletin of Mathematical Linguistics*, 106(1):125–146.

Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie-m: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. *arXiv preprint arXiv:2012.15674*.

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

Gabriel Peyré, Marco Cuturi, et al. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

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

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

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

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

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

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28, pages 2692–2700.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzman, 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. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Do explicit alignments robustly improve multilingual encoders? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4471–4482, Online. Association for Computational Linguistics.

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. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

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

Wei Zhao, Steffen Eger, Johannes Bjerva, and Isabelle Augenstein. 2020. Inducing language-agnostic multilingual representations. *arXiv preprint arXiv:2008.09112*.

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

A Pre-Training Data

We use raw sentences from the Wikipedia dump and CCNet[4] as monolingual corpora. The CCNet corpus we use is reconstructed following (Conneau et al., 2020) to reproduce the CC-100 corpus. The resulting corpus contains 94 languages. Table 8 and Table 9 report the language codes and data size of CCNet and Wikipedia dump. Notice that several languages share the same ISO language codes, e.g., zh represents both Simplified Chinese and Traditional Chinese. Besides, Table 10 shows the statistics of our parallel corpora.

Code	Size (GB)	Code	Size (GB)	Code	Size (GB)
af	0.2	hr	1.4	pa	0.8
am	0.4	hu	9.5	pl	28.6
ar	16.1	hy	0.7	ps	0.4
as	0.1	id	17.2	pt	39.4
az	0.8	is	0.5	ro	11.0
ba	0.2	it	47.2	ru	253.3
be	0.5	ja	86.8	sa	0.2
bg	7.0	ka	1.0	sd	0.2
bn	5.5	kk	0.6	si	1.3
ca	3.0	km	0.2	sk	13.6
ckb	0.6	kn	0.3	sl	6.2
cs	14.9	ko	40.0	sq	3.0
cy	0.4	ky	0.5	sr	7.2
da	6.9	la	0.3	sv	60.4
de	99.0	lo	0.2	sw	0.3
el	13.1	lt	2.3	ta	7.9
en	731.6	lv	1.3	te	2.3
eo	0.5	mk	0.6	tg	0.7
es	85.6	ml	1.3	th	33.0
et	1.4	mn	0.4	tl	1.2
eu	1.0	mr	0.5	tr	56.4
fa	19.0	ms	0.7	tt	0.6
fi	5.9	mt	0.2	ug	0.2
fr	89.9	my	0.4	uk	13.4
ga	0.2	ne	0.6	ur	3.0
gl	1.5	nl	25.9	uz	0.1
gu	0.3	nn	0.4	vi	74.5
he	4.4	no	5.5	yi	0.3
hi	5.0	or	0.3	zh	96.8

Table 8: The statistics of CCNet used for pre-training.

B Hyperparameters for Pre-Training

As shown in Table 11, we present the hyperparameters for pre-training XLM-ALIGN. We use the same vocabulary with XLM-R (Conneau et al., 2020).

---

[4] https://github.com/facebookresearch/cc_net

Code	Size (GB)	Code	Size (GB)	Code	Size (GB)
af	0.12	hr	0.28	pa	0.10
am	0.01	hu	0.80	pl	1.55
ar	1.29	hy	0.60	ps	0.04
as	0.04	id	0.52	pt	1.50
az	0.24	is	0.05	ro	0.42
ba	0.13	it	2.70	ru	5.63
be	0.31	ja	2.65	sa	0.04
bg	0.62	ka	0.37	sd	0.02
bn	0.41	kk	0.29	si	0.09
ca	1.10	km	0.12	sk	0.21
ckb	0.00	kn	0.25	sl	0.21
cs	0.81	ko	0.56	sq	0.11
cy	0.06	ky	0.10	sr	0.74
da	0.33	la	0.05	sv	1.70
de	5.43	lo	0.01	sw	0.03
el	0.73	lt	0.19	ta	0.46
en	12.58	lv	0.12	te	0.45
eo	0.25	mk	0.34	tg	0.04
es	3.38	ml	0.28	th	0.52
et	0.23	mn	0.05	tl	0.04
eu	0.24	mr	0.10	tr	0.43
fa	0.66	ms	0.20	tt	0.09
fi	0.68	mt	0.01	ug	0.03
fr	4.00	my	0.15	uk	2.43
ga	0.03	ne	0.06	ur	0.13
gl	0.27	nl	1.38	uz	0.06
gu	0.09	nn	0.13	vi	0.76
he	1.11	no	0.54	yi	0.02
hi	0.38	or	0.04	zh	1.08

Table 9: The statistics of Wikipedia dump used for pre-training.

ISO Code	Size (GB)	ISO Code	Size (GB)
en-ar	5.88	en-ru	7.72
en-bg	0.49	en-sw	0.06
en-de	4.21	en-th	0.47
en-el	2.28	en-tr	0.34
en-es	7.09	en-ur	0.39
en-fr	7.63	en-vi	0.86
en-hi	0.62	en-zh	4.02

Table 10: Parallel data used for pre-training.

C Hyperparameters for Fine-Tuning

In Table 12, we present the hyperparameters for fine-tuning XLM-R$_{base}$ and XLM-ALIGN on the XTREME end tasks. For each task, the hyperparameters are searched on the joint validation set of all languages.

D Detailed Results on XTREME

We present the detailed results of XLM-ALIGN on XTREME in Table 13-19.

Hyperparameters	Value
Layers	12
Hidden size	768
FFN inner hidden size	3,072
Attention heads	12
Training steps	150K
Batch size	2,048
Adam ϵ	1e-6
Adam β	(0.9, 0.98)
Learning rate	2e-4
Learning rate schedule	Linear
Warmup steps	10,000
Gradient clipping	1.0
Weight decay	0.01
Self-labeling layer	10
Entropic regularization μ	1.0
Sinkhorn iterations	2
Alignment filtering iterations	2
Alignment filtering α	0.9

Table 11: Hyperparameters used for pre-training XLM-ALIGN.

	POS	NER	XQuAD	MLQA	TyDiQA	XNLI	PAWS-X
Batch size	{8,16,32}	8	32	32	32	32	32
Learning rate	{1,2,3}e-5	{5,...,9}e-6	{2,3,4}e-5	{2,3,4}e-5	{2,3}e-5	{5,...,8}e-6	{1,2}e-5
LR schedule	Linear	Linear	Linear	Linear	Linear	Linear	Linear
Warmup	10%	10%	10%	10%	10%	12,500 steps	10%
Weight decay	0	0	0	0	0	0	0
Epochs	10	10	4	{2,3,4}	{5,10,15,20}	10	10

Table 12: Hyperparameters used for fine-tuning XLM-R$_{base}$ and XLM-ALIGN on the XTREME end tasks.

Model	af	ar	bg	de	el	en	es	et	eu	fa	fi	fr	he	hi	hu	id	it
XLM-ALIGN	88.5	69.1	88.8	88.8	85.8	95.9	88.5	84.9	68.3	70.9	84.8	88.1	79.6	71.6	83.3	72.3	89.4

Model	ja	kk	ko	mr	nl	pt	ru	ta	te	th	tl	tr	ur	vi	yo	zh	Avg
XLM-ALIGN	51.1	75.3	53.8	80.3	89.3	87.6	88.9	62.3	85.9	60.2	90.1	74.8	63.3	55.9	24.2	67.9	76.0

Table 13: Results on part-of-speech tagging.

Model	ar	he	vi	id	jv	ms	tl	eu	ml	ta	te	af	nl	en	de	el	bn	hi	mr	ur
XLM-ALIGN	57.7	54.3	72.5	49.7	56.9	68.3	72.0	53.1	68.6	58.0	54.6	76.3	82.1	84.2	77.9	76.4	73.1	69.2	64.9	65.8

Model	fa	fr	it	pt	es	bg	ru	ja	ka	ko	th	sw	yo	my	zh	kk	tr	et	fi	hu	Avg
XLM-ALIGN	53.2	79.0	79.4	78.8	73.8	78.9	66.2	23.0	70.6	56.6	2.2	69.3	43.8	56.5	28.3	49.2	77.5	73.3	77.0	77.0	63.7

Table 14: Results on WikiAnn named entity recognition.

Model	en	es	de	el	ru	tr	ar	vi	th	zh	hi	Avg
XLM-ALIGN	85.7 / 74.6	70.3 / 52.5	76.6 / 60.3	75.5 / 56.8	79.4 / 60.8	71.8 / 54.7	75.4 / 59.4	72.1 / 61.0	70.9 / 55.5	76.7 / 56.9	67.3 / 56.8	74.7 / 59.0

Table 15: Results on XQuAD question answering.

Model	en	es	de	ar	hi	vi	zh	Avg
XLM-ALIGN	81.5 / 68.3	70.3 / 52.2	64.5 / 49.8	60.7 / 41.2	65.2 / 47.5	69.8 / 48.9	64.4 / 40.4	68.1 / 49.8

Table 16: Results on MLQA question answering.

Model	en	ar	bn	fi	id	ko	ru	sw	te	Avg
XLM-ALIGN	69.4 / 56.2	68.7 / 49.4	56.0 / 38.9	64.2 / 47.2	73.9 / 57.9	53.0 / 40.4	62.3 / 38.0	60.1 / 42.8	51.0 / 31.9	62.1 / 44.8

Table 17: Results on TyDiQA question answering.

Model	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	Avg
XLM-ALIGN	86.7	80.6	81.0	78.8	77.4	78.8	77.4	75.2	73.9	76.9	73.8	77.0	71.9	67.1	66.6	76.2

Table 18: Results on XNLI natural language inference.

Model	en	fr	de	es	ja	ko	zh	Avg
XLM-ALIGN	95.1	89.3	90.5	90.7	79.1	79.5	83.2	86.8

Table 19: Results on PAWS-X cross-lingual paraphrase adversaries.