VECO: Variable and Flexible Cross-lingual Pre-training for Language Understanding and Generation

Fuli Luo[*], **Wei Wang**[*]**, Jiahao Liu, Yijia Liu, Bin Bi, Songfang Huang, Fei Huang, Luo Si**
Alibaba Group
{lfl259702,hebian.ww,glacier.ljh,yanshan.lyj}@alibaba-inc.com
{b.bi,songfang.hsf,f.huang,luo.si}@alibaba-inc.com

Abstract

Existing work in multilingual pretraining has demonstrated the potential of cross-lingual transferability by training a unified Transformer encoder for multiple languages. However, much of this work only relies on the shared vocabulary and bilingual contexts to encourage the correlation across languages, which is loose and implicit for aligning the contextual representations between languages. In this paper, we plug a cross-attention module into the Transformer encoder to explicitly build the interdependence between languages. It can effectively avoid the degeneration of predicting masked words only conditioned on the context in its own language. More importantly, when fine-tuning on downstream tasks, the cross-attention module can be plugged in or out on-demand, thus naturally benefiting a wider range of cross-lingual tasks, from language understanding to generation.

As a result, the proposed cross-lingual model delivers new state-of-the-art results on various cross-lingual understanding tasks of the XTREME benchmark, covering text classification, sequence labeling, question answering, and sentence retrieval. For cross-lingual generation tasks, it also outperforms all existing cross-lingual models and state-of-the-art Transformer variants on WMT14 English-to-German and English-to-French translation datasets, with gains of up to 1∼2 BLEU. [1]

1 Introduction

Cross-lingual pre-trained models like mBERT (Devlin et al., 2019), XLM (Lample and Conneau, 2019) and XLM-R (Conneau et al., 2019) that target providing contextualized representations for the inputs across languages, have shown large poten-

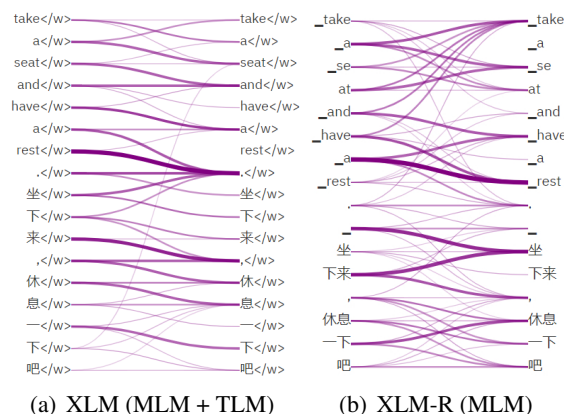(a) XLM (MLM + TLM)   (b) XLM-R (MLM)

Figure 1: The attention scores of XLM and XLM-R with the input of a pair of parallel sentences: *Take a seat and have a rest* in English and its translated Chinese sentence. The darker line denotes a higher score. We can found that there are only a few attention patterns across English and Chinese subwords.

tial on a variety of cross-lingual understanding and generation tasks.

Behind the great success, two major factors play the role of aligning the contextual representations between languages: 1) build the shared vocabulary across languages through subword tokenization, which supports the simple extension of masked language modeling (MLM) from English corpus to multilingual corpus; 2) capture the alignment in parallel data via concatenating two sentences as input, called translation language modeling (TLM). However, both of these two mechanisms rely on the ***self-attention*** module (query=key/value) of the Transformer encoder to *implicitly* enhance the interdependence between languages, which may lead to few attention patterns across languages. Taking Figure 1 as an example, even though inputting a pair of parallel sentences, both models *only* attend to the English context to build the representation of English tokens, while ignoring the se-
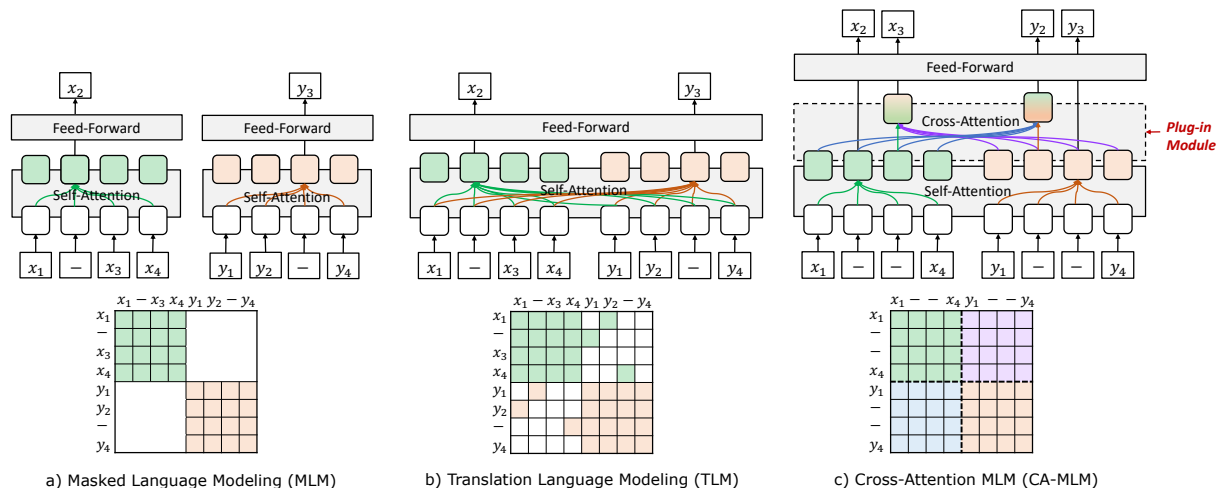

Figure 2: A schematic comparison of cross-lingual pre-training tasks and their attention matrices. When predicting the masked words of different languages: a) MLM can only attend to the context in its own language; b) TLM implicitly attend to a part of words across languages (as shown in Figure 1). However, c) the proposed CA-MLM can: (1) not only attend to the context in its own language to predict words x_2 and y_3, (2) but also can firstly attend to its own context and then explicitly attend to all words across languages to predict words x_3 and y_2 via a plug-in cross-attention module.

mantically related Chinese tokens. That is, the self-attention module captures little communication across languages, which is crucial for learning universal cross-lingual representations.

Based on the above observation, we propose to plug a ***cross-attention*** module (query!=key/value) into the Transformer encoder and design a cross-attention MLM task to *explicitly* capture the interdependence between languages. As illustrated in Figure 2 (c), the cross-attention module takes the representation of x as query and y as key/value (purple lines) to build the representations of x in the next layer, thus explicitly aligning the representations across languages (purple attention matrices). It can effectively avoid the degeneration of predicting masked words only conditioned on the context in its own language. Moreover, what distinguishes our work from pre-training an encoder-decoder model (Liu et al., 2020b) is that we also keep the good nature (i.e., bidirectional contextual modeling) of the original encoder by *unplugging* the cross-attention from the model to predicting the masked words (e.g., x_2 and y_3).

Furthermore, when fine-tuning on various downstream tasks, we can choose either *plug-in* or *plug-out* the cross-attention module on-demand, thus making it suitable for both cross-lingual language understanding (NLU) and generation tasks (NLG). For cross-lingual NLU tasks, if plugging the cross-attention module out, we can adopt the same fine-

tuning methods as an encoder-only model like XLM. However, we find that plugging the cross-attention module in fine-tuning can better utilize the bilingual context to boost the performance. For cross-lingual NLG like machine translation (MT), the cross attention is already jointly pre-trained with the whole network. Therefore, the parameters of the decoder do not need to be re-adjusted substantially in the following tuning process, thus fundamentally solving the main drawback of utilizing pre-trained encoders like XLM for initializing encoder-decoder models.

We call our approach VECO for "Variable and Flexible Cross-lingual Pre-training". We validate VECO on a variety of representative cross-lingual understanding and generation benchmarks. Regrading cross-lingual understanding tasks, we conduct experiments on the XTREME benchmark consisting of 9 cross-lingual tasks, including text classification, sequence labeling, question answering, and sentence retrieval. VECO ranks first at the XTREME leaderboard [2] at the submission deadline. Regrading cross-lingual generation tasks, we validate VECO on the widely used WMT14 English-German and English-French machine translation benchmarks. VECO obtains 44.5 and 31.7 BLEU scores, consistently outperforming existing cross-lingual pre-training approaches and state-of-the-art Transformer variants by around 1~2 BLEU.

---

[2] https://sites.research.google/xtreme

2 Pre-training of VECO

2.1 Overview of VECO

VECO extends from a multi-layer Transformer encoder and plugs a cross-attention module in each layer. Given a pair of input $(\boldsymbol{x}, \boldsymbol{y})$ and its corrupted version $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ via randomly masking part of its tokens, the model builds two types of contextualized vector representation for each token:

- One suit of contextual representations $\mathbf{H}$, denoted as green blocks and yellow blocks in Figure 2 (c), are only build on self-attention module (i.e., unpluging the cross-attention module) in each layer.

- Another suit of contextual representations $\mathbf{S}$, denoted as mixed color blocks in Figure 2 (c), are build on both the self-attention and cross-attention modules [3].

The model is trained to predict the masked tokens via two corresponding representations, conditioning on both its own context and paired context, respectively. Take predicting the masked words in sequence $\boldsymbol{x}$ as an example, the training objective is the cross-entropy of the gold distribution and predicted distribution $P(\boldsymbol{x}|\hat{\boldsymbol{x}})$ and $P(\boldsymbol{x}|\hat{\boldsymbol{y}}, \hat{\boldsymbol{x}})$ computed via the above two suits of contextual representations. Thus, the training objective of cross-attention masked language modeling (CA-MLM) can be formulated as

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = \\
-\log P(\boldsymbol{x}|\hat{\boldsymbol{x}}; \theta_{\mathbf{s}}) - \log P(\boldsymbol{x}|\hat{\boldsymbol{y}}, \hat{\boldsymbol{x}}; \theta_{\mathbf{s}}, \theta_{\mathbf{c}}) \quad (1) \\
-\log P(\boldsymbol{y}|\hat{\boldsymbol{y}}; \theta_{\mathbf{s}}) - \log P(\boldsymbol{y}|\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}; \theta_{\mathbf{s}}, \theta_{\mathbf{c}})
\end{aligned}
$$

where $\theta_{\mathbf{s}}$ and $\theta_{\mathbf{c}}$ are the parameters of self-attention and cross-attention modules.

2.2 Architecture

The backbone network of VECO is composed of a stack of N Transformer layers. Each layer has three modules: a required self-attention module, a plug-and-play cross-attention module, and a required feed-forward linear module. Both self-attention and cross-attention modules are based on the multi-head attention (Vaswani et al., 2017). An attention function can be described as mapping a query ($\mathbf{Q}$) and a set of key-value ($\mathbf{K}$-$\mathbf{V}$) pairs to an output.

---

[3]For simplicity of illustration, we only show the mixed representations $\mathbf{S}$ of $\boldsymbol{x}_3$ and $\boldsymbol{y}_2$ in Figure 2 (c).

For the self-attention module, all the queries, keys and values are the same representations from the previous layer. Specifically, for the l-th Transformer layer, the output of a self-attention head $\mathbf{A}_l^s$ is computed via:

$$
\mathbf{Q} = \mathbf{H}^{l-1}\mathbf{W}_l^Q \quad (2)
$$

$$
\mathbf{K} = \mathbf{H}^{l-1}\mathbf{W}_l^K \quad (3)
$$

$$
\mathbf{V} = \mathbf{H}^{l-1}\mathbf{W}_l^V \quad (4)
$$

$$
\mathbf{A}_l^s = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\mathsf{T}}{\sqrt{d_k}})\mathbf{V} \quad (5)
$$

where $\mathbf{H}^{l-1}$ are the previous layer's outputs, $\mathbf{W}_l^Q, \mathbf{W}_l^K, \mathbf{W}_l^V$ are the parameter matrices of self-attention modules.

For the cross-attention module, the queries come from the previous layer, and the keys and values come from the last layer's representations of paired input. Specifically, for the l-th layer, the output of a cross-attention head $\mathbf{A}_l^c$ is computed via:

$$
\mathbf{Q} = \mathbf{S}^{l-1}\mathbf{U}_l^Q \quad (6)
$$

$$
\mathbf{K} = \mathbf{H}^L\mathbf{U}_l^K \quad (7)
$$

$$
\mathbf{V} = \mathbf{H}^L\mathbf{U}_l^V \quad (8)
$$

$$
\mathbf{A}_l^c = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\mathsf{T}}{\sqrt{d_k}})\mathbf{V} \quad (9)
$$

where $\mathbf{S}^{l-1}$ are the previous layer's outputs, $\mathbf{U}_l^Q, \mathbf{U}_l^K, \mathbf{U}_l^V$ are the parameter matrices of cross-attention modules.

Finally, the output $\mathbf{H}^L$ of the last layer is used to recover the masked tokens of $\boldsymbol{x}$, conditioning on its own context.

$$
P(\boldsymbol{x}|\hat{\boldsymbol{x}}) = \text{softmax}(f(\mathbf{H}_x^L)) \quad (10)
$$

$$
P(\boldsymbol{y}|\hat{\boldsymbol{y}}) = \text{softmax}(f(\mathbf{H}_y^L)) \quad (11)
$$

where f is the feed-forward network that maps the output vectors into the dictionary. $\mathbf{H}_x^L$ and $\mathbf{H}_y^L$ are computed via Eq 2~5 when $\mathbf{H}_x^0$ and $\mathbf{H}_y^0$ are the word embeddings of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively.

Meanwhile, $\mathbf{S}^L$, conditioning on the context of the paired sequence $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$, is used to predict the masked tokens of $\boldsymbol{y}$.

$$
P(\boldsymbol{x}|\hat{\boldsymbol{y}}, \hat{\boldsymbol{x}}) = \text{softmax}(f(\mathbf{S}_x^L)) \quad (12)
$$

$$
P(\boldsymbol{y}|\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) = \text{softmax}(f(\mathbf{S}_y^L)) \quad (13)
$$

where $\mathbf{S}_x^L$ and $\mathbf{S}_y^L$ are computed via Eq 6~9 with the corresponding word embeddings and $\mathbf{H}^L$.
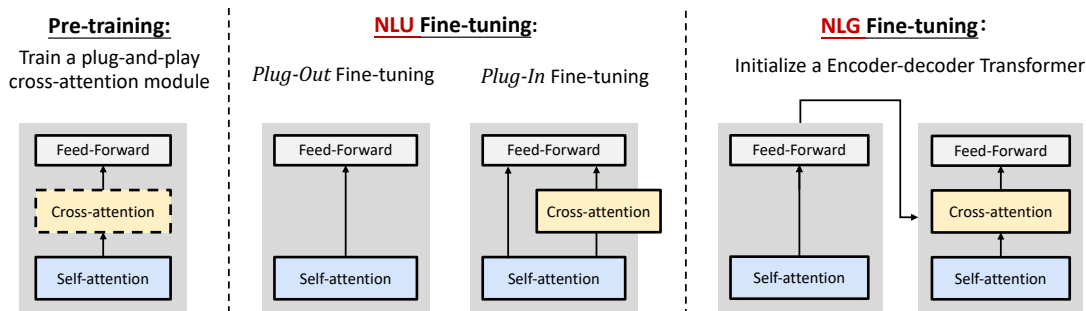
Figure 3: The overview of VECO. During pre-training, a plug-and-play cross-attention module is jointly pre-trained along with the self-attention module. When fine-tuning on natural language understanding (NLU) tasks, the cross-attention module can be either plug-in or plug-out on demand. When fine-tuning on natural language generation (NLG) tasks, VECO can initialize an encoder-decoder module (the mainstream backbone model of generation tasks) since all those necessary modules in the encoder and decoder are already pre-trained.

Note that when optimizing the objectives based on Eq 12 and Eq 13, we apply a stop-gradients operation (Chen and He, 2020) to $\mathbf{H}^L$ (i.e., $\mathbf{H}^L$ is treated as a constant in this term). This operation can largely speed up the training by avoiding the backpropagation on a $2L$-layer network. Moreover, it even stabilizes the training of deep post-layernorm Transformer, which requires non-trivial efforts regarding carefully designing learning rate schedulers and cutting-edge optimizers (Liu et al., 2020a; Bachlechner et al., 2020).

3 Fine-tuning VECO for Downstream Cross-lingual Understanding and Generation Tasks

As Figure 3 illustrated, when fine-tuning on various downstream tasks, one advantage of VECO is its flexibility for initializing both the encoder-only Transformer for understanding tasks and encoder-decoder Transformer for generation tasks. Beyond it, we also explore a fine-tuning approach combined with the characteristics of VECO .

3.1 VECO for Cross-lingual Understanding

Due to the plug-and-play cross-attention module, we explore two fine-tuning approaches:

- *Plug-Out* fine-tuning is to unplug the cross-attention module from the pre-trained model. In other words, the architecture of the fine-tuned model is almost the same as mBERT or XLM. Specifically, the contextual representations from the last layer $\mathbf{H}_x^L$ is used to predict the label of input x.

- *Plug-In* fine-tuning is to plug the cross-attention module into the fine-tuned model, if

the bilingual or automatically translated training data y is available in the downstream task. Specifically, we concatenated the two representations $[\mathbf{H}_x^L : \mathbf{S}_x^L]$ to predict the label of x, $[\mathbf{H}_y^L : \mathbf{S}_y^L]$ to predict the label of y. [4].

3.2 VECO for Cross-lingual Generation

For pre-trained encoders like XLM, it is not a trivial problem to incorporate them into the sequence-to-sequence architecture – the mainstream backbone model of generation tasks (Zhu et al., 2020). One of the drawbacks or challenges could be that the encoder-to-decoder attention is not pre-trained. Therefore, the parameters of the decoder need to be re-adjusted along with the encoder in the following fine-tuning process (Ren et al., 2019).

However, under the framework of VECO , the cross-attention is jointly pre-trained along with the whole network, making it easy to provide full initialization for sequence-to-sequence models. Specifically, the self-attention module is used to initialize both the corresponding modules in the encoder and decoder for contextual modeling, while the cross-attention module is used to initialize the encoder-to-decoder attention. It's okay whether you continue to tie the self-attention parameters during fine-tuning. Directly pre-training a sequence-to-sequence model like mBART (Liu et al., 2020b) could be another solution for NLG tasks, but we found mBART is not so effective in cross-lingual NLU tasks. We refer the reader to the Section 7 for detailed experiments and analysis.

---

[4] *Plug-In* fine-tuning is not suitable for the zero-shot setting (also called *cross-lingual transfer*) due to the lack of bilingual or translated pair (x, y)

Model	Architecture	#Parameters	Enc Layers	Dec Layers	#Languages	#Vocab	Training Data
mBERT (Devlin et al., 2019)	Encoder-only	110M	12	-	104	110k	Wikipedia
XLM (Lample and Conneau, 2019)	Encoder-only	570M	24	-	100	200k	Wikipedia
XLM-R (Conneau et al., 2019)	Encoder-only	550M	24	-	100	250k	CommonCrawl
mRASP (Lin et al., 2020)	Encoder-decoder	375M	6	6	32	64k	Translation
MMTE (Siddhant et al., 2020)	Encoder-decoder	375M	6	6	103	64k	Translation
mBART (Liu et al., 2020b)	Encoder-decoder	680M	12	12	25	250k	CommonCrawl
VECO	Flexible	662M	24*		50	250k	CommonCrawl + Translation

Table 1: Comparison of large cross-lingual models. * denotes VECO unifies the encoder and decoder.

4 Pre-training Setup

Model Configuration We pre-train a 24-layer model with 1024 embedding/hidden size and 4096 feed-forward size. We do not use language embeddings to allow our model to better deal with downstream tasks of unseen languages. We adopt the same 250K vocabulary that is also used by XLM-R (Conneau et al., 2019). Table 1 shows the other details of baselines and VECO .

Pre-Training Data We collect monolingual and bilingual corpus covering 50 languages. For monolingual training datasets, we reconstruct Common-Crawl Corpus used in XLM-R (Conneau et al., 2019). We extract 1.36TB data in 50 languages, which contains 6.5G sentences and 0.4G documents. We up/down-sample the monolingual text like XLM from each language with a smoothing parameter $\alpha = 0.5$. For bilingual data, we collect from the OPUS website [5] like previous works (Lample and Conneau, 2019; Chi et al., 2020b). There are 6.4G parallel sentences, covering 879 language pairs across 50 languages. See more statistics of training data in Appendix A.

Optimization Settings For each iteration, we alternately sample a batch of adjacent segments from the monolingual corpus and a batch of parallel sentences from bilingual datasets to conduct a pair of masked input $(\hat{x}, \hat{y})$. We adopt the translation language modeling (TLM) when the inputs are parallel bilingual sentences. Thus the overall training objective is the sum of TLM and the proposed CA-MLM objectives. During training, the model parameters except for cross-attention are initialized by XLM-R. We first freeze the parameters of XLM-R and only update the cross-attention parameters for faster convergence. Then, we jointly train the whole model. We pre-train our model with mixed-precision training using 64 Nvidia Telsa V100 32GB GPUs. Appendix A shows additional details.

---
[5] http://opus.nlpl.eu/

5 Experiments on Cross-lingual Understanding Tasks

5.1 Experimental Setup

Downstream Tasks We conduct cross-lingual NLU evaluations on XTREME (Hu et al., 2020), a representative massively multilingual benchmark that consists of 9 understanding tasks over 40 languages. XTREME tasks can be classified into four different categories: (1) sentence-pair classification: XNLI (Conneau et al., 2018), PAWS-X (Yang et al., 2019); (2) structured prediction: POS (Nivre et al., 2018), Wikiann NER (Pan et al., 2017); (3) question answering: XQuAD (Artetxe et al., 2020), MLQA (Lewis et al., 2020), TyDiQA (Clark et al., 2020); (4) sentence retrieval: BUCC 2018 (Zweigenbaum et al., 2017), Tatoeba (Artetxe and Schwenk, 2019). Tasks in the first three categories are provided: 1) golden training corpus in English, 2) translated training corpus in other languages, and 3) dev/test set in all languages. For sentence retrieval tasks, no training datasets are provided. We refer the reader to Hu et al. (2020) for additional details about the datasets.

Fine-tuning Setting Following previous works (Conneau et al., 2019; Hu et al., 2020), we consider two typical fine-tuning settings: (1) *Cross-lingual Transfer* which fine-tunes the pre-trained model using English golden data only and directly performs inference on the test data of different target languages; (2) *Translate-Train-All* fine-tunes a multilingual model on the concatenation of all data (golden training corpus in English and translated training corpus in other languages). Note that for two sequence-labeling tasks (POS, NER), the position of token labels in the translated text generally differs from that in the source text. Following FILTER (Fang et al., 2020), we use the model trained only on the English training dataset as a teacher, to label the translated text. To have a fair comparison with the strong baseline XLM-R (Conneau et al., 2019)

Datasets	XNLI	PAWS-X	POS	NER	XQuAD	MLQA	TyDiQA	BUCC	Tatoeba	
#Languages	15	7	33	40	11	7	9	5	33	
Metrics	Acc	Acc	F1	F1	F1/EM	F1/EM	F1/EM	F1	Acc	**Avg.**
Cross-lingual Transfer: Fine-tune model on English training set and test on all languages										
MMTE[†]	67.4	81.3	73.5	58.3	64.4/46.2	60.3/41.4	58.1/43.8	59.8	37.9	59.5
mBERT[†]	65.4	81.9	70.3	62.2	64.5/49.4	61.4/44.2	59.7/43.0	56.7	38.7	59.6
XLM[†]	69.1	80.9	70.1	61.2	59.8/44.3	48.5/32.6	43.6/29.1	56.8	32.6	55.5
XLM-R[†]	79.2	86.4	72.6	65.4	76.6/60.8	71.6/**53.2**	65.1/45.0	66.0	57.3	68.1
VECO$_{out}$	**79.9**	**88.7**	**75.1**	**65.7**	**77.3/61.8**	**71.7/53.2**	**67.6/49.1**	**85.0**	**75.1**	**73.1**
Translate-Train-All: Fine-tune model on English training data and translated data of the target language										
XLM-R[‡]	82.6	90.4	-	-	80.2/65.9	72.8/54.3	66.5/47.7	-	-	-
XLM-R*	82.8	90.2	72.6	65.4	80.0/65.8	73.0/54.3	74.5/58.3	80.2	75.2	74.4
FILTER	83.9	91.4	76.2	67.7	82.4/68.0	76.2/57.7	68.3/50.9	84.5	84.5	77.0
VECO$_{out}$	83.0	91.1	75.1	65.7	79.9/66.3	73.1/54.9	75.0/58.9	89.3	86.9	77.2
VECO$_{in}$	**84.3**	**92.8**	**79.8**	**71.0**	**83.9/70.9**	**77.5/59.3**	**79.4/63.7**	**92.6**	**91.1**	**81.0**

Table 2: XTREME results on each dataset (as of ACL submission deadline). Averaged results on the four categories can be found at leaderboard: https://sites.research.google/xtreme. "†" and "‡" indicates results from Hu et al. (2020) and Fang et al. (2020), respectively. "*" indicates the results obtained by our implementation. The detailed results for each language are in Appendix D.

under the translate-train-all setting, we also show the results of XLM-R using the same fine-tuning hyperparameters as VECO .

5.2 Experimental Results

The detailed test results of nine tasks on the XTREME benchmark are shown in Table 2. It demonstrates that the proposed VECO outperforms previous cross-lingual models on all datasets. Compared to XLM-R, it averagely scores 5.0 and 6.6 points higher under the cross-lingual transfer and translation-train-all settings, respectively.

In the cross-lingual transfer setting, VECO delivers a large improvement compared to XLM-R, especially on zero-shot sentence retrieval tasks (BUCC, Tatoeba). This phenomenon reflects that our model can better build the interdependence between languages. Thus it can better mine parallel sentences in a multilingual corpus.

Under the translation-train-all setting, it can be observed that VECO with Plug-In fine-tuning (VECO$_{in}$) is better than Plug-Out fine-tuning (VECO$_{out}$). We conclude the reasons as two-fold. On the input side, the Plug-Out fine-tuning individually takes multilingual instances as input, while the Plug-In fine-tuning considers the bilingual instances [6] at each run. On the model side, the Plug-In fine-tuning can encourage correspondence across language via the cross-attention module. Note that the Plug-In fine-tuning method also outperforms FILTER (Fang et al., 2020), an enhanced cross-lingual fine-tuning method that also takes the

bilingual instance as the input of XLM-R. It further demonstrates the effectiveness of VECO and its specialized fine-tuning method.

We conclude the reasons for the above performance improvement as two-fold: 1) the introduction of bilingual data during pre-training, which is a direct way to enhance the cross-lingual ability of the model; 2) Stronger ability to enhance the interdependence and fusion among languages via the proposed CA-MLM pre-training tasks. To analyze which plays a leading role, we conduct a set of more fair experiments in Section 7.

6 Experiments on Cross-lingual Generation Tasks

6.1 Experimental Setup

Datasets We choose the machine translation (MT) task, a typical cross-lingual generation scenario. In order to illustrate the generality of our approach and have a fair comparison with the most recent state-of-the-art Transformer work (Liu et al., 2020a), we choose two most widely used datasets: WMT14 English→German (En-De) and English→French (En-Fr) translation. WMT14 En-De is a medium-resource dataset that provides 4.5M pairs for training and validation. We adopt standard newstest2014 as the test set. WMT14 En-Fr is a high-resource dataset that contains 36M pairs of parallel sentences. We use newstest2012+newstest2013 for validation and newstest2016 for test. We measure case-insensitive tokenized BLEU with multi-bleu.perl and de-

---

[6]English instance with its translated one.

Model	WMT14 En-Fr		WMT14 En-De	
	BLEU	SacreBLEU	BLEU	SacreBLEU
Randomly Initialize				
Baseline	42.9	40.4	28.7	27.8
Liu et al. (2020a)	43.8	41.8	30.1	29.5
Randomly Initialize + More Bilingual Data*				
Baseline*	-	-	30.6	29.5
Cross-lingual Model Initialize				
mBART	43.2	41.0	30.0	29.1
mRASP	44.3	41.7	30.3	-
XLM-R	43.8	41.2	30.9	29.9
VECO	**44.5**	**42.0**	**31.7**	**30.6**
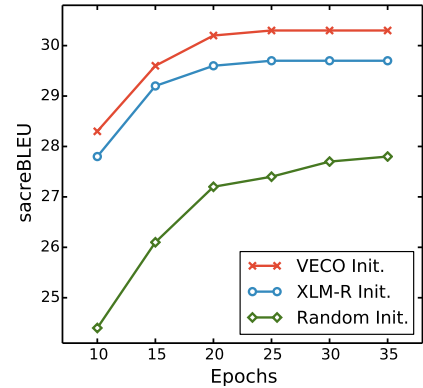
Table 3: (left) Results on machine translation. (right) Learning curves of different initialization methods.

tokenized `SacreBLEU` [7] to avoid the influence of different tokenization and normalization between models (Post, 2018).

Fine-tuning Setting We fine-tune our model using `fairseq` [8] toolkit and adopt comparable training settings with baselines. We run WMT 14 En-De and En-Fr MT experiments on 16 and 32 V100 GPUs, respectively. The batch size is 64k for En-De and 256k for En-Fr. The total training updates are set to 100k. The learning rate is 1e-4/2e-4, with linear warm-up over the first 16k steps and linear decay. We average the last 10 checkpoints and use beam search with a beam size of 5.

Baselines We consider two types of Transformer baselines: randomly initialized and cross-lingual models initialized. For random initialization, we reproduce a Transformer baseline that adopts the same architecture and fine-tuning hyperparameters as VECO but with random initialization. Besides, we compare to the state-of-the-art Deep Transformer (Liu et al., 2020a). For cross-lingual encoder-decoder models, we include mBART (Liu et al., 2020b) and mRASP (Lin et al., 2020), which show impressive results on MT. Note that since we tied the self-attention weights of each encoder layer with each decoder layer, the whole parameters of mBART and VECO are comparable. We also conduct the WMT experiments for XLM-R, following the totally same fine-tuning settings as VECO , but leaving the encoder-to-decoder attention un-initialized.

---

[7]Hash: BLEU+case.mixed+lang.en-{de,fr}+numrefs.1+ smooth.exp+test.wmt14/full+tok.13a+version.1.4.9
[8]https://github.com/pytorch/fairseq

6.2 Experimental Results

Table 3 (left) shows the results on the machine translation. We can observe that VECO can largely outperform the randomly initialized same-sized Transformer baseline by 2.3 BLEU points. Moreover, it even beats the (randomly initialized) state-of-the-art Deep-Transformer (Liu et al., 2020a), which is three times deep as VECO . Among the cross-lingual models, VECO can consistently outperform the best models, averaged on two datasets, by 0.8 BLEU points.

Table 3 (right) displays the BLEU scores of same-sized models during training. We find that VECO initialized model can get a surprising more than 28 SacreBLEU score just after 10 epochs, which is better than the final score of the randomly initialized model at 35 epochs. It reveals that VECO can provide a fairly good initialization for the machine translation model, which can converge quickly and further boost the results.

One might suspect that the main reason for the performance improvement is leveraging parallel corpus during pre-training. To figure it out, we conduct a more comparable experiment. We first train an out-of-domain Transformer model using the whole En-De parallel data ($\sim$ 68M) used in VECO pre-training, and then continue to train the model on the in-domain WMT14 En-De training dataset. Results are shown in Table 3 (left) marked with *. Under this set of a totally fair comparison, VECO still maintains a lead of 1.1 BLEU score. This directly confirms that the improvement in MT is not only due to the use of bilingual data. More importantly, CA-MLM ensures *better* use of bilingual and large-scale unlabeled multilingual corpus.

Method	#Layers	WMT14 En-De	
		BLEU	SacreBLEU
Randomly Initialize	3	28.5	27.6
	6	28.6	27.7
VECO *Initialize*	First-3	30.8	29.8
	Last-3	31.2	30.3
	First-6	31.1	30.1
	Last-6	31.5	30.5
	Full-24	**31.7**	**30.6**

Table 4: Results of utilizing VECO to initialize deep encoder and shallow decoder (3/6-layer) Transformers.

6.3 Potential of Initializing Shallow Decoder

Online translation applications usually have a restriction of inference time. The most direct way is to reduce the decoder layers since previous MT works (Liu et al., 2020a) have shown that deeper encoders are more worthwhile than deeper decoders. Based on this, we also explore the potential of the VECO to initialize deep encoder and shallow decoder Transformers, which is a blank in the cross-lingual pre-training works.

Table 4 contrasts two ways of initializing a Transformer with n decoder layers ($n < 24$) via selecting: (1) the first n layers; (2) the last n layers from a 24-layer pre-trained VECO model. We consider $n = \{3, 6\}$ to conduct experiments. We find that selecting the last n layers exhibits better performance than selecting the first n layers. It reveals that the last several layers play a more important role in making predictions over the whole vocabulary. Moreover, we can find that there is 0.2~0.3 BLEU gain when increasing the decoder layers from 3 to 6. However, we observe that only marginal improvement can be gained when further increasing the decoder layers from 6 to 24, which is also in line with the findings in Liu et al. (2020a). Regardless of the initialization method, the VECO initialized model can gain consistent 1~2 BLEU improvement over the randomly initialized model.

7 Analysis and Ablation Study

We perform an ablation study to investigate where the improvement in cross-lingual NLU and NLG tasks mainly comes from. Specifically, there are three main aspects we have studied:

1. How much performance improvement comes from the parallel translation corpus used in pre-training?

2. How effective of the CA-MLM pre-training

Data	Models	Tasks	XNLI	IWSLT
Mono.	XLM	MLM	59.8	33.7
	mBART	MLM	57.3	32.9
	VECO	CA-MLM	**60.6**	**34.0**
Bili.	XLM	MLM+TLM	64.5	33.9
	mBART	MLM+TLM	60.8	34.5
	VECO	CA-MLM +TLM	**67.7**	**36.0**

Table 5: Ablation study of small-sized models on XNLI and IWSLT14 De-En translation dataset.

task, especially compared to the MLM and TLM pre-training tasks?

3. How about pre-training a sequence-to-sequence model like mBART for NLU and NLG tasks?

To figure out these questions, we train XLM, mBART and VECO model from scratch using the same datasets and parameter settings (see Appendix A for more details). All of them is pre-trained via MLM and TLM tasks. Note that the MLM task generally refers to predict the masked words of source language, while the TLM task generally refers to predict the words of the target language. Specifically for mBART that is under the framework of encoder-decoder, the input of encoder is masked sequence $\hat{x}$, and the target of decoder is the masked words of source input x (for MLM task), or the parallel sentence y (for TLM task).

Table 5 shows the results of two representative datasets of cross-lingual NLU and NLG. We can observe that, when using monolingual corpus only, VECO can outperform XLM by 0.8 points on the XNLI dataset and 0.3 BLEU scores on the IWSLT14 De-En translation dataset. It suggests that the CA-MLM can still benefit from adjacent sentences in monolingual corpus [9], to be equipped with a stronger ability of contextual modeling. Moreover, when pre-training both on the monolingual and bilingual corpus, VECO can even achieve a larger improvement compared to XLM, with 3.2 and 2.1 points improvement on two datasets, respectively. It reveals that CA-MLM objective of VECO can better utilize the bilingual corpus, compared to only optimized by TLM and MLM of XLM.

Moreover, we find that pre-training a sequence-to-sequence model like mBART (Liu et al., 2020b)

---

[9]As noted in Section 4, we take two adjacent sentences in the monolingual corpus as (x, y).

performs worst on NLU tasks like XNLI [10], almost 6 points worse than VECO and near 2 points worse than XLM. One possible explanation could be that the unidirectional language modeling in the decoder might be sub-optimal for NLU tasks. And even on the machine translation task, mBART still performs worse than VECO when pre-training on the same bilingual datasets. We conclude that it is because that VECO can do better in the contextual modeling of source input x via a explicit masked language modeling objective in Eq 10 applied to x_2 in Figure 2 (c).

8 Related Work

mBERT (Devlin et al., 2019) is a key step towards building a unified contextual language representation over multiple languages. It simply shares all languages' vocabulary and trains a bidirectional Transformer encoder, achieving promising results in various cross-lingual NLU tasks. There have been several extensions that follow the same encoder-only backbone as mBERT. The main difference is the introduction of more training corpus (e.g., bilingual data) and pre-training tasks. XLM (Lample and Conneau, 2019) utilizes both monolingual and bilingual corpus to perform the masked language modeling. XLM-R (Conneau et al., 2019) extends to be built on RoBERTa (Liu et al., 2019) using larger monolingual training data. Other works (Huang et al., 2019; Yang et al., 2020; Chi et al., 2020b) propose new pre-training tasks to utilize the bilingual data better. However, there are two main drawbacks of these works. First, they mainly rely on the self-attention module in the Transformer encoder to implicitly build the interdependence between languages, leading to few attention patterns across languages due to the "lazy" network. Second, even though they show impressive performance improvement on cross-lingual understanding tasks like XNLI, only marginal improvement has been gained on cross-lingual generation tasks like machine translation, especially on high-resource languages.

A feasible solution for cross-language generation is to pre-train a denoising auto-encoder like mBART (Liu et al., 2020b). It extends BART (Lewis et al., 2019) to the multilingual setting, demonstrating significant gains in low/medium-resource machine translation, but

with a decrease in high resource languages. Unlike mBART, Chi et al. (2020a) first trains an encoder via MLM and then frozen the encoder to train the decoder only via two generative tasks. A similar approach is also proposed in Liang et al. (2020) and Lin et al. (2020), with the main difference in the joint training of encoder-decoder with code-switch tricks. However, all these cross-lingual models emphasize training a dedicated model for NLG. Thus they may hurt the NLU capabilities of the model. The ablation study in Section 7 also validates that it is sub-optimal to train an encoder-encoder network for NLU tasks.

This paper endeavors to build a unified cross-lingual model for NLU and NLG tasks via a plug-and-play cross-attention module. More importantly, the cross-attention module plays a role in the explicit alignment of encoded representations of different languages, thus largely contributing to building a unified cross-lingual model.

9 Conclusion

We present VECO, a variable and flexible cross-lingual pre-training model, targets at explicitly capturing the interdependence between languages via a plug-and-play cross-attention module. Based on the flexible characteristics, VECO can initialize both NLU preferred encoder-only and NLG specialized encoder-decoder Transformer. Moreover, we also introduce a Plug-In fine-tuning approach to encourage the fusion between languages, combining the feature of VECO and cross-language downstream tasks.

Taken together, VECO achieves consistent improvements on various language understanding and generation tasks, broadening the way of thinking about pre-trained backbone architecture and fine-tuning methods under the cross-lingual scenario.

---

[10]We follow BART (Lewis et al., 2019) by utilizing the final representation from the decoder for classification tasks.

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, ACL 2020, Online, July 5-10, 2020*, pages 4623–4637. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Trans. Assoc. Comput. Linguistics*, 7:597–610.

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Garrison W Cottrell, and Julian McAuley. 2020. Rezero is all you need: Fast convergence at large depth. *arXiv preprint arXiv:2003.04887*.

Xinlei Chen and Kaiming He. 2020. Exploring simple siamese representation learning. *CoRR*, abs/2011.10566.

Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. 2020a. Cross-lingual natural language generation via pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

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*.

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. In *Transactions of the Association of Computational Linguistics*.

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, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of EMNLP 2018*, pages 2475–2485.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Yuwei Fang, Shuohang Wang, Zhe Gan, Siqi Sun, and Jingjing Liu. 2020. FILTER: An enhanced fusion method for cross-lingual language understanding. *arXiv preprint arXiv:2009.05166*.

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*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating Cross-lingual Extractive Question Answering. In *Proceedings of ACL 2020*.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, et al. 2020. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. *arXiv preprint arXiv:2004.01401*.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2649–2663. Association for Computational Linguistics.

Xiaodong Liu, Kevin Duh, Liyuan Liu, and Jianfeng Gao. 2020a. Very deep transformers for neural machine translation. *arXiv preprint arXiv:2008.07772*.

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

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

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, et al. 2018. Universal dependencies 2.2.

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 ACL 2017*, pages 1946–1958.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation*.

Shuo Ren, Yu Wu, Shujie Liu, Ming Zhou, and Shuai Ma. 2019. Explicit cross-lingual pre-training for unsupervised machine translation. volume abs/1909.00180.

Aditya Siddhant, Melvin Johnson, Henry Tsai, Naveen Ari, Jason Riesa, Ankur Bapna, Orhan Firat, and Karthik Raman. 2020. Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 8854–8861.

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*.

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*.

Jian Yang, Shuming Ma, Dongdong Zhang, Shuangzhi Wu, Zhoujun Li, and Ming Zhou. 2020. Alternating language modeling for cross-lingual pre-training. In *Proceedings of the 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3685–3690. Association for Computational Linguistics.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating BERT into neural machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Pierre Zweigenbaum, Serge Sharoff, and Reinhard Rapp. 2017. Overview of the second BUCC shared task: Spotting parallel sentences in comparable corpora. In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora, BUCC@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 60–67. Association for Computational Linguistics.

A   Pre-Training Details

For monolingual data, following XLM-R (Conneau et al., 2019), we build a clean CommonCrawl Corpus using an open-source tool CCNet (Wenzek et al., 2019). There are 1.36TB monolingual data in 50 languages before up/down-sampling. Table 6 reports the language codes and statistics of pre-training data. We collect bilingual corpus in 50 languages from the OPUS website[11], including MultiUN, UNPC, Bombay, EU-bookshop, OpenSubtitles2018, Tanzil, GlobalVoices, ParaCrawl, MultiParaCrawl, DGT, Tilde, Europarl, Wikipedia, ECB, TED2013, News-Commentary, Ubuntu, Books, UN, infopankki-v1, EUconst, and Bianet. In total, there are 1TB bilingual training data before pre-processing, covering 879 language pairs. Table 7 lists the statistics for each language pair. We then apply subword tokenization directly on raw text data using Sentence Piece Model (Kudo and Richardson, 2018) without any additional preprocessing.

We use the whole corpus to train VECO and a subset ($\sim 1/4$) that contains 33 languages to train small-sized XLM, mBART and VECO . The full set of pre-training hyperparameters for small-sized and large-sized VECO (default) are listed in Table 8.

B   More details about Illustrated Attention

The models illustrated with attention patterns in Figure 1 of main paper (not appendix), are the base-sized XLM [12] and XLM-R [13]. We show the attention scores averaged on all heads in the middle layer.

C   Fine-Tuning Details on XTERME

We select the model with the best average result over all the languages on the dev sets, by searching the learning rate over [5e-6,8e-6,1e-5,2e-5,3e-5] for the *Cross-lingual Transfer* setting and [5e-6,6e-6,7e-6,8e-6,9e-6] for *Translate-Train-All* setting, training epoch over [3,5,10], and batch size over [16,32,64].

D   Detailed Results on XTREME

The detailed results of each XTREME task under the cross-lingual transfer and translate-train-all settings on all languages are listed in the following tables.

---

[11]http://opus.nlpl.eu/
[12]https://huggingface.co/xlm-mlm-tlm-xnli15-1024
[13]https://huggingface.co/xlm-roberta-base

Language	#Document(M)	#Sentence(M)	Size(GB)
af	0.023	0.522	0.107
ar	2.823	42.659	11.786
bg	0.919	14.743	5.217
bn	0.750	9.217	4.264
cs	3.980	55.754	9.668
de	21.410	310.942	66.333
el	1.740	24.334	9.737
en	130.087	2,215.534	479.099
es	17.569	267.764	58.774
et	0.347	5.252	0.877
eu	0.342	5.216	0.613
fr	15.819	267.888	58.023
fa	2.506	43.570	13.831
fi	1.530	23.790	3.940
fy	0.027	0.537	0.054
gu	0.039	0.519	0.228
gd	0.009	0.126	0.020
he	0.755	12.338	3.073
hi	0.536	7.303	3.762
hu	1.816	29.962	6.421
id	3.417	60.908	11.528
it	9.336	133.006	30.854
ja	27.967	588.926	71.785
jv	0.002	0.138	0.030
ka	0.141	1.756	0.766
kk	0.061	1.545	0.448
ko	11.609	227.396	27.837
lt	0.552	7.996	1.480
lv	0.281	4.159	0.798
ms	0.334	3.762	0.455
ml	0.162	2.615	1.025
my	0.045	0.893	0.306
mr	0.059	0.708	0.365
pl	6.642	93.760	19.082
pt	8.623	128.107	25.612
ne	0.080	0.829	0.429
nl	6.513	85.997	16.648
ru	35.887	580.291	203.105
ro	1.944	31.929	7.056
si	0.132	2.927	0.902
sw	0.057	0.945	0.179
ta	0.876	20.376	6.422
te	0.288	4.995	1.721
tr	18.547	291.081	40.321
th	6.278	117.826	27.941
tl	0.166	5.611	0.679
vi	12.183	234.071	37.919
ur	0.460	7.509	2.003
yo	0.0002	0.003	0.0005
zh	27.067	497.408	87.005
Total	382.735	6,475.444	1,360.526

Table 6: The statistics of monolingual pre-training corpus.

Pair	#Sent(K)	Pair	#Sent(K)	Pair	#Sent(K)	Pair	#Sent(K)	Pair	#Sent(K)	Pair	#Sent(K)	Pair	#Sent(K)	Pair	#Sent(K)	Pair	#Sent(K)
af-ar	12.34	bg-my	0.08	de-he	12751.69	en-tr	46584.82	eu-zh	19.76	fy-vi	34.95	id-pt	6825.29	ko-sw	6.74	pl-es	46863.47
af-bg	18.19	bg-ne	0.01	de-hi	106.11	en-ur	781.60	fa-fi	4485.62	gd-es	21.62	id-ro	7944.59	ko-ta	13.74	pl-pt	72437.93
af-bn	1.19	bg-nl	30757.50	de-hu	24409.40	en-vi	3563.39	fa-fr	4507.06	gd-it	13.26	id-ru	5039.44	ko-te	0.93	pl-ru	19170.23
af-cs	17.93	bg-pl	33043.03	de-id	4786.89	en-yo	0.13	fa-he	4944.80	gd-pl	12.29	id-si	366.00	ko-th	230.84	pl-sw	1424.02
af-de	19.28	bg-pt	30058.54	de-it	35936.62	en-zh	28952.02	fa-hi	186.23	gd-pt	18.90	id-sw	30.56	ko-tl	1.21	pl-tl	1039.37
af-el	29.83	bg-ro	38925.52	de-ja	1472.72	es-et	18090.74	fa-hu	5201.51	gd-ru	10.39	id-ta	35.37	ko-tr	1246.58	pl-tr	32470.18
af-en	44.70	bg-ru	17423.43	de-ka	123.12	es-eu	793.59	fa-id	3220.00	gd-tr	14.12	id-te	13.30	ko-ur	57.21	pl-ur	391.99
af-es	34.31	bg-si	460.50	de-kk	3.72	es-fa	5696.70	fa-it	4243.56	he-hi	57.85	id-tl	7.80	ko-vi	345.79	pl-vi	3790.71
af-et	6.34	bg-sw	10.80	de-ko	776.89	es-fi	34222.07	fa-ja	1072.14	he-hu	23959.87	id-tr	8017.99	ko-zh	56.43	pt-ro	33802.95
af-fa	3.07	bg-ta	27.14	de-lt	9134.99	es-fr	96233.21	fa-ko	92.32	he-id	6362.29	id-ur	172.71	lt-lv	6546.76	pt-ru	14698.48
af-fi	10.25	bg-te	17.14	de-lv	8532.06	es-he	27060.49	fa-kk	1.01	he-it	19908.66	id-vi	2081.70	lt-ml	66.40	pt-si	450.40
af-fr	18.56	bg-th	2733.84	de-ml	294.16	es-hi	85.35	fa-lv	228.40	he-ja	1683.29	it-ja	1613.05	lt-ms	393.89	pt-sw	13.06
af-fy	36.94	bg-tl	6.69	de-ms	1228.82	es-id	8015.69	fa-ml	308.49	he-ka	149.06	it-ka	106.70	lt-pl	9965.36	pt-te	26.37
af-he	14.53	bg-tr	31179.35	de-my	0.68	es-it	49423.51	fa-ms	1072.22	he-kk	2.38	it-ko	1125.97	lt-pt	7663.84	pt-th	2561.09
af-hi	1.15	bg-ur	71.60	de-ne	0.28	es-ja	1929.41	fa-ne	0.01	he-lt	1220.91	it-lt	7359.92	lt-ro	5786.22	pt-tl	10.35
af-hu	16.32	bg-vi	2855.13	de-nl	34909.49	es-kk	2.54	fa-nl	5010.64	he-lv	461.81	it-lv	6607.27	lt-ru	950.02	pt-tr	27428.79
af-id	4.56	bg-zh	746.27	de-pt	32610.10	es-ka	181.19	fa-pt	4998.09	he-ml	250.07	it-ml	235.96	lt-si	106.53	pt-vi	73.57
af-it	15.01	bn-cs	340.51	de-ro	24261.82	es-ko	1229.50	fa-ru	4205.20	he-ms	1455.61	it-my	0.36	lt-sw	0.02	pt-yo	0.05
af-ja	1.98	bn-de	346.51	de-ru	10904.25	es-lt	7702.99	fa-si	292.78	he-my	0.05	it-ne	1.02	lt-ta	13.04	ro-ru	19568.56
af-lt	0.65	bn-el	340.94	de-si	324.86	es-ml	339.71	fa-ta	83.30	he-nl	22186.61	it-nl	37644.29	lt-th	263.89	ro-si	504.24
af-lv	1.08	bn-en	752.08	de-sw	45.61	es-ms	1731.36	fa-th	1201.04	he-pl	24962.23	it-pl	35037.31	lt-tl	1.36	ro-sw	10.72
af-ml	2.18	bn-es	480.35	de-ta	42.32	es-my	2.50	fa-tl	7.02	he-pt	21226.36	it-pt	35301.98	lt-tr	1377.40	ro-ta	33.50
af-ms	1.31	bn-et	252.68	de-te	12.81	es-ne	2.87	fa-ur	568.00	he-ro	26370.15	it-ro	32153.38	lt-ur	4.47	ro-te	24.44
af-nl	22.61	bn-eu	42.42	de-th	1695.53	es-nl	46908.79	fa-vi	1514.04	he-ru	14873.77	it-ru	17669.12	lt-vi	486.84	ro-th	2874.73
af-pl	1096.89	bn-fa	391.89	de-tl	12.91	es-pt	47542.26	fi-fr	28973.81	he-si	435.87	it-si	9.93	lt-zh	40.65	ro-tl	8.61
af-pt	22.68	bn-fi	279.35	de-tr	17579.53	es-ro	48229.60	fi-he	17820.49	he-sw	0.06	it-sw	15.77	lv-ml	23.32	ro-tr	36549.61
af-ro	32.19	bn-fr	373.13	de-ur	218.89	es-ru	55569.05	fi-hi	55.60	he-ta	23.99	it-ta	17.39	lv-ms	163.28	ro-ur	73.55
af-ru	15.41	bn-he	302.62	de-vi	2284.70	es-si	512.22	fi-hu	27350.30	he-te	18.65	it-te	9.71	lv-nl	6622.81	ro-vi	3207.73
af-si	0.98	bn-hu	321.36	de-zh	587.96	es-sw	41.33	fi-it	26756.85	he-th	2666.00	it-th	2447.55	lv-pl	9460.93	ro-zh	947.91
af-ta	1.13	bn-id	360.65	el-en	55078.46	es-ta	31.19	fi-ja	1599.82	he-tl	6.58	it-tl	1.36	lv-pt	6672.14	ru-si	340.11
af-th	2.08	bn-ja	142.19	el-es	46876.21	es-te	21.76	fi-kk	3.41	he-tr	25179.32	it-tr	25770.29	lv-ro	4833.77	ru-sw	84.77
af-tr	24.22	bn-ka	8.68	el-et	16463.57	es-tl	13.55	fi-lt	7507.00	he-ur	20.57	it-ur	69.89	lv-ru	435.73	ru-ta	61.50
af-vi	3.30	bn-ko	93.92	el-eu	673.93	es-tr	39805.02	fi-lv	6732.38	he-vi	2813.73	it-vi	2542.41	lv-si	34.42	ru-te	10.80
ar-bg	23090.32	bn-lt	96.24	el-fa	5137.52	es-ur	79.44	fi-ms	1276.96	he-zh	563.24	it-yo	0.10	lv-sw	0.01	ru-th	2194.91
ar-bn	378.28	bn-lv	41.21	el-fi	28885.65	es-vi	3215.16	fi-nl	30693.72	hi-hu	60.05	it-zh	473.74	lv-te	4.01	ru-tl	13.43
ar-cs	24147.25	bn-ml	93.14	el-fr	38560.84	es-yo	0.12	fi-pt	29269.50	hi-id	85.85	ja-kk	1.21	lv-th	108.92	ru-tr	19317.60
ar-de	12733.65	bn-ms	203.84	el-he	22042.85	es-zh	28688.60	fi-ru	12403.26	hi-it	60.12	ja-ka	35.37	lv-tr	515.30	ru-ur	417.23
ar-el	22486.60	bn-my	0.78	el-hi	62.26	et-eu	406.33	fi-si	391.99	hi-ja	46.14	ja-lt	281.74	lv-ur	1.08	ru-vi	2289.72
ar-en	60392.55	bn-ne	0.78	el-hu	34559.75	et-fa	3085.41	fi-ta	20.08	hi-ka	0.80	ja-lv	99.97	lv-vi	209.40	ru-yo	0.10
ar-es	57561.29	bn-nl	331.34	el-id	7098.25	et-fi	15969.08	fi-te	17.13	hi-ko	33.66	ja-ml	79.78	lv-zh	14.71	ru-zh	28138.59
ar-et	9738.71	bn-pt	333.59	el-it	34337.63	et-fr	15697.59	fi-tl	5.91	hi-lt	23.67	ja-ms	489.33	ml-ms	101.75	si-ta	6.33
ar-eu	578.30	bn-ro	337.94	el-ja	1740.08	et-he	9814.49	fi-tr	22551.99	hi-lv	12.61	ja-nl	1716.42	ml-nl	268.10	si-te	1.85
ar-fa	5679.85	bn-si	47.49	el-ka	167.39	et-hi	43.98	fi-ur	19.43	hi-ml	30.28	ja-pl	1756.87	ml-pt	280.62	si-th	109.38
ar-fi	17169.90	bn-sw	23.91	el-kk	2.33	et-id	4282.23	fi-vi	2517.08	hi-ms	40.38	ja-pt	1843.14	ml-ro	325.97	si-tl	3.02
ar-fr	50632.52	bn-ta	15.67	el-ko	1130.94	et-it	14462.11	fr-he	21218.88	hi-ne	0.04	ja-ru	1491.65	ml-ru	310.59	si-tr	492.12
ar-he	20577.16	bn-th	129.60	el-lt	7400.42	et-ja	1176.51	fr-hi	68.31	hi-nl	92.46	ja-si	162.96	ml-si	28.01	si-ur	4.95
ar-hi	96.26	bn-tl	2.05	el-lv	6549.40	et-ka	110.02	fr-id	6235.29	hi-pl	681.08	ja-ta	18.92	ml-sw	12.47	si-vi	210.15
ar-hu	23770.38	bn-tr	441.74	el-ml	302.85	et-ko	492.79	fr-it	41162.37	hi-pt	62.44	ja-te	5.68	ml-ta	15.90	si-zh	14.28
ar-id	6989.56	bn-ur	108.74	el-ms	1547.63	et-lt	7431.17	fr-ja	1608.52	hi-ro	82.89	ja-tl	10.06	ml-th	81.03	sw-ta	6.24
ar-it	20070.27	bn-vi	219.57	el-my	0.55	et-lv	6728.85	fr-ka	139.63	hi-ru	142.53	ja-tr	439.25	ml-tl	3.30	sw-th	6.24
ar-ja	1847.98	bn-zh	85.24	el-ne	1.04	et-ml	179.99	fr-kk	1.34	hi-si	11.41	ja-ur	61.41	ml-tr	439.25	sw-tr	91.95
ar-ka	161.65	cs-de	24049.84	el-nl	37188.78	et-nl	16560.63	fr-ko	991.60	hi-sw	12.52	ja-vi	679.31	ml-ur	100.52	sw-ur	50.29
ar-kk	1.28	cs-el	35372.28	el-pt	35491.54	et-pl	19633.08	fr-lt	9440.34	hi-ta	41.00	ja-zh	104.37	ml-vi	124.30	sw-yo	0.03
ar-ko	1262.60	cs-en	54470.47	el-ro	37986.26	et-ro	15880.62	fr-ml	278.47	hi-te	23.18	ka-ko	17.13	ml-zh	34.77	sw-zh	19.31
ar-lt	1177.67	cs-et	17819.46	el-ru	17052.36	et-ru	6630.25	fr-ms	1423.08	hi-th	37.53	ka-lt	30.49	ms-nl	1409.07	ta-te	21.16
ar-lv	433.66	cs-eu	686.53	el-si	466.44	et-si	331.22	fr-my	0.06	hi-tl	0.51	ka-ml	6.56	ms-pt	1523.57	ta-th	14.15
ar-ml	348.33	cs-fa	5417.48	el-sw	4.85	et-sw	0.01	fr-ne	1.45	hi-tr	176.39	ka-nl	155.10	ms-ro	1732.68	ta-tr	77.76
ar-ms	1555.33	cs-fi	28031.47	el-ta	20.44	et-te	14.44	fr-pt	42850.13	hi-ur	101.10	ka-pt	165.00	ms-ru	1210.56	ta-vi	49.89
ar-my	0.18	cs-fr	34876.02	el-te	18.10	et-th	1746.50	fr-si	393.48	hi-vi	32.99	ka-ru	104.82	ms-si	204.06	ta-zh	13.02
ar-ne	0.41	cs-he	24503.29	el-th	2505.71	et-tr	11408.82	fr-tr	29245.91	hi-zh	25.57	ka-si	7.96	ms-sw	8.99	te-th	0.96
ar-nl	21273.78	cs-hi	86.86	el-tl	10.13	et-ur	19.52	fr-vi	2752.32	hu-id	7253.81	ka-tl	1.27	ms-ta	15.24	te-tr	18.84
ar-pl	24819.83	cs-id	7310.27	el-tr	31048.88	et-zh	405.30	fr-yo	0.10	hu-it	33513.06	ka-ur	1.98	ms-te	4.70	te-vi	9.34
ar-pt	20379.56	cs-it	33935.96	el-ur	24.36	eu-fa	14.34	fr-zh	28008.77	hu-ja	1767.63	ka-vi	53.58	ms-th	413.17	th-tl	7.28
ar-ro	26187.15	cs-ka	163.35	el-vi	2966.14	eu-fi	581.61	fy-es	49.12	hu-ka	165.84	kk-lt	0.83	ms-tl	7.26	th-tr	3054.07
ar-ru	45992.72	cs-kk	1.26	el-yo	0.11	eu-fr	636.16	fy-he	44.06	hu-kk	2.58	kk-lv	1.13	ms-tr	1754.22	th-ur	58.65
ar-si	483.96	cs-ko	1199.62	el-zh	649.81	eu-hi	9.98	fy-it	47.88	hu-ko	1168.66	kk-ms	1.12	ms-ur	68.94	th-vi	672.82
ar-sw	16.52	cs-lt	7694.12	en-es	156560.00	eu-hu	663.68	fy-ja	37.61	hu-lt	7623.58	kk-nl	1.85	ms-vi	851.69	th-zh	133.45
ar-ta	37.15	cs-lv	6745.84	en-et	22284.30	eu-id	568.66	fy-pl	95.81	hu-lv	6776.32	kk-pl	77.88	ms-zh	85.86	tl-tr	14.51
ar-te	19.33	cs-ml	319.93	en-eu	805.78	eu-it	435.12	fy-ru	45.83	hu-ml	279.13	kk-ro	2.35	my-nl	0.10	tl-vi	5.86
ar-th	2959.96	cs-ms	1592.17	en-fa	7462.52	eu-ja	139.14	fy-sw	0.37	hu-ms	1581.43	kk-ru	2.22	my-pt	0.10	tr-ur	473.08
ar-tl	7.58	cs-ne	0.07	en-fi	42783.36	eu-ko	72.17	fy-tr	45.40	hu-my	0.06	kk-th	0.93	my-ro	0.03	tr-vi	3178.03
ar-tr	26683.62	cs-nl	34427.07	en-fr	161519.91	eu-lt	108.12			hu-nl	33904.34	kk-tr	2.59	my-ru	0.81	tr-zh	1029.21
ar-ur	126.33	cs-pt	32469.01	en-fy	126.19	eu-lv	36.81			hu-pl	39869.14	kk-vi	1.18	my-sw	0.15	ur-vi	12.52
ar-vi	2875.00	cs-ro	39226.31	en-gd	47.02	eu-ml	42.72			hu-pt	31715.19	ko-lt	148.54	my-ur	0.02	ur-zh	99.78
ar-yo	0.01	cs-ru	19703.43	en-he	35233.87	eu-ms	129.20			hu-ru	19172.99	ko-lv	57.10	my-zh	0.13	vi-zh	148.22
ar-zh	28120.22	cs-si	454.26	en-hi	1844.38	eu-nl	619.88			hu-si	460.99	ko-ms	291.25	ne-pt	0.38		
bg-bn	310.12	cs-sw	17.34	en-hu	55233.87	eu-pt	641.30			hu-sw	0.68	ko-nl	1120.75	ne-ro	0.04		
bg-cs	34502.46	cs-ta	32.81	en-id	9677.33	eu-ru	435.12			hu-ta	20.63	ko-pl	2722.47	ne-ru	1.30		
bg-de	19852.81	cs-te	18.72	en-it	76257.21	eu-si	34.56			hu-te	17.57	ko-pt	1119.49	ne-sw	0.05		
bg-el	32130.86	cs-th	2858.53	en-ja	2177.89	eu-ta	3.35			hu-th	2867.23	ko-ro	1242.76	ne-tr	0.03		
bg-en	47247.04	cs-tl	7.44	en-ka	199.98	eu-te	0.73			hu-tl	10.79	ko-ru	959.46	ne-ur	0.06		
bg-es	39728.55	cs-tr	32797.28	en-kk	3.71	eu-tl	2.60			hu-tr	32494.90	ko-si	58.66	ne-zh	0.01		
bg-et	15188.54	cs-ur	122.87	en-ko	1493.95	eu-tr	722.77			hu-ur	23.32			nl-pt	37775.73		
bg-eu	605.10	cs-vi	3040.14	en-lt	10992.89	eu-vi	201.28			hu-vi	2974.61			nl-ro	36051.60		
bg-fa	4927.53	cs-zh	894.87	en-lv	9883.08					hu-zh	730.70			nl-ru	16582.78		
bg-fi	25191.01	de-el	30170.64	en-ml	573.95					id-it	5831.16			nl-si	410.92		
bg-fr	30185.98	de-en	83872.47	en-ms	2050.83					id-ja	1271.31			nl-sw	31.38		
bg-he	22887.40	de-es	41634.80	en-my	2.43					id-ka	85.07			nl-ta	39.21		
bg-hi	71.38	de-et	15186.40	en-ne	2.89					id-kk	1.03			nl-te	16.07		
bg-hu	34293.44	de-eu	534.93	en-nl	65918.54					id-ko	605.78			nl-th	2548.14		
bg-id	7047.21	de-fa	3948.14	en-pl	59729.77					id-lt	855.43			nl-tr	28822.22		
bg-it	27649.85	de-fi	25753.06	en-pt	61861.36					id-lv	342.36			nl-ur	171.71		
bg-ja	1658.40	de-fr	44392.06	en-ro	60415.46					id-ml	230.67			nl-vi	2748.28		
bg-ka	193.27			en-ru	65105.13					id-ms	1614.63			nl-zh	866.75		
bg-kk	3.40			en-si	601.16					id-my	0.11						
bg-ko	1056.96			en-sw	171.65					id-ne	0.07						
bg-lt	5604.11			en-ta	125.96					id-nl	6493.33						
bg-lv	4748.15			en-te	27.22												
bg-ml	283.77			en-tl	16.03												
bg-ms	1506.56															**Total**	6,421,152.04

Table 7: The statistics of bilingual (parallel) pre-training corpus.

Pre-training Hyperparameters	Large	Small
Number of layers	24	6
Hidden Size	1024	768
FFN inner hidden size	4096	3072
Attention heads	16	12
Attention head size	64	64
Embedding Size	1024	768
Mask percent (monolingual/ bilingual)	15%/25%	15%/25%
Learning Rate Decay	Linear	Linear
Warmup steps	12k	12k
Learning Rate	2e-4	3e-4
Adam ϵ	1e-6	1e-6
Adam β_1	0.9	0.9
Adam β_2	0.98	0.999
Attention Dropout	0.1	0.1
Dropout	0.1	0.1
Weight Decay	0.01	0.01
Max Sequence Length (monolingual/bilingual)	512/128	512/128
Batch Size (monolingual/bilingual)	1024/4096	1024/4096
Train Steps	240k	240k
Total Parameters	662M	247M

Table 8: The pre-training hyperparameters.

Model	en	ar	bg	de	el	es	fr	hi	ru	sw	th	tr	ur	vi	zh	**Avg.**
Cross-lingual Transfer																
XLM-R	88.7	77.2	83.0	82.5	80.8	83.7	82.2	75.6	79.1	71.2	77.4	78.0	71.7	79.3	78.2	79.2
VECO$_{out}$	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
Translate-Train-All																
XLM-R	88.6	82.2	85.2	84.5	84.5	85.7	84.2	80.8	81.8	77.0	80.2	82.1	77.7	82.6	82.7	82.6
VECO$_{out}$	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
VECO$_{in}$	89.3	83.7	87.0	85.9	85.8	87.3	86.7	81.8	83.6	79.9	82.5	84.3	77.7	84.4	84.0	84.3

Table 9: XNLI accuracy scores for each language.

Model	en	de	es	fr	ja	ko	zh	**Avg.**
Cross-lingual Transfer								
XLM-R	94.7	89.7	90.1	90.4	78.7	79.0	82.3	86.4
VECO$_{out}$	96.2	91.3	91.4	92.0	81.8	82.9	85.1	88.7
Translate-Train-All								
VECO$_{out}$	96.4	93.0	93.0	93.5	87.2	86.8	87.9	91.1
VECO$_{in}$	96.5	94.4	94.3	94.0	89.0	90.3	91.0	92.8

Table 10: PAWS-X accuracy scores.

Model	de	fr	ru	zh	**Avg.**
Cross-lingual Transfer					
XLM-R	67.5	66.5	73.5	56.7	66.0
VECO$_{out}$	89.6	84.6	87.4	78.5	85.0
Translate-Train-All					
VECO$_{out}$	93.0	88.7	89.9	85.7	89.3
VECO$_{in}$	95.4	91.9	93.1	89.9	92.6

Table 11: BUCC F1 results.

Model	af	ar	bg	de	el	en	es	et	eu	fa	fi	fr	he	hi	hu	id	it
Cross-lingual Transfer																	
XLM-R	89.8	67.5	88.1	88.5	86.3	96.1	88.3	86.5	72.5	70.6	85.8	87.2	68.3	76.4	82.6	72.4	89.4
VECO$_{out}$	88.3	67.4	87.4	88.5	86.7	95.9	89.0	87.8	75.1	70.9	86.2	88.9	67.5	76.2	82.9	72.9	89.9
Translate-Train-All																	
VECO$_{in}$	92.5	73.7	93.4	91.8	90.4	95.2	91.3	90.6	79.1	79.8	89.5	91.4	79.1	80.6	88.4	74.8	91.8

	ja	kk	ko	mr	nl	pt	ru	ta	te	th	tl	tr	ur	vi	yo	zh	**Avg.**
Cross-lingual Transfer																	
XLM-R	15.9	78.1	53.9	80.8	89.5	87.6	89.5	65.2	86.6	47.2	92.2	76.3	70.3	56.8	24.6	25.7	73.8
VECO$_{out}$	31.4	79.3	53.1	84.3	89.8	88.3	90.2	64.3	85.8	48.0	93.7	77.2	69.2	58.1	26.2	39.4	75.1
Translate-Train-All																	
VECO$_{in}$	45.1	78.0	63.7	84.5	92.7	90.1	92.6	72.6	88.5	55.2	88.8	76.8	75.0	70.5	24.3	63.0	79.8

Table 12: POS results (Accuracy) for each language.

Model	en	af	ar	bg	bn	de	el	es	et	eu	fa	fi	fr	he	hi	hu	id	it	ja	jv
Cross-lingual Transfer																				
XLM-R	84.7	78.9	53.0	81.4	78.8	78.8	79.5	79.6	79.1	60.9	61.9	79.2	80.5	56.8	73.0	79.8	53.0	81.3	23.2	62.5
VECO$_{out}$	83.8	77.5	48.2	83.9	77.2	79.4	79.3	75.4	80.4	68.3	68.2	80.6	80.1	55.0	71.0	80.9	52.9	81.7	19.4	63.2
Translate-Train-All																				
VECO$_{in}$	80.7	82.5	66.4	84.1	78.4	82.2	82.4	79.7	84.7	78.2	68.8	84.9	79.1	69.7	76.6	85.1	77.3	83.8	21.3	70.3

	ka	kk	ko	ml	mr	ms	my	nl	pt	ru	sw	ta	te	th	tl	tr	ur	vi	yo	zh
Cross-lingual Transfer																				
XLM-R	71.6	56.2	60.0	67.8	68.1	57.1	54.3	84.0	81.9	69.1	70.5	59.5	55.8	1.3	73.2	76.1	56.4	79.4	33.6	33.1
VECO$_{out}$	67.1	51.2	59.9	63.4	65.0	70.0	56.1	83.4	83.1	71.3	70.5	60.5	56.2	1.4	71.3	80.4	69.3	76.0	37.4	29.1
Translate-Train-All																				
VECO$_{in}$	77.0	67.2	71.0	73.3	74.1	71.8	63.8	85.5	80.8	72.8	77.0	69.1	67.5	2.6	74.0	85.2	71.5	76.4	32.8	31.0

Table 13: NER results (F1) for each language.

Model	en	ar	de	el	es	hi	ru	th	tr	vi	zh	**Avg.**
Cross-lingual Transfer												
XLM-R	86.5 / 75.7	68.6 / 49.0	80.4 / 63.4	79.8 / 61.7	82.0 / 63.9	76.7 / 59.7	80.1 / 64.3	74.2 / 62.8	75.9 / 59.3	79.1 / 59.0	59.3 / 50.0	76.6 / 60.8
VECO$_{out}$	87.6 / 76.5	73.6 / 56.1	79.8 / 62.2	79.6 / 61.6	81.2 / 61.6	74.7 / 57.6	78.7 / 62.1	72.8 / 60.6	75.1 / 58.3	79.0 / 59.8	69.2 / 59.2	77.3 / 61.8
Translate-Train-All												
VECO$_{out}$	88.3/77.9	76.9/61.1	80.5/64.6	81.5/64.1	84.2/66.8	78.8/62.5	80.2/66.1	77.0/70.4	77.8/62.2	82.5/63.7	71.6/69.4	79.9/66.3
VECO$_{in}$	90.2/79.5	81.8/66.4	85.4/69.8	85.3/69.0	87.2/70.8	83.7/67.9	85.6/71.6	80.0/74.7	82.4/68.6	85.8/68.3	74.9/73.1	83.9/70.9

Table 14: XQuAD results (F1 / EM) for each language.

Model	en	ar	de	es	hi	vi	zh	**Avg.**
Cross-lingual Transfer								
XLM-R	83.5 / 70.6	66.6 / 47.1	70.1 / 54.9	74.1 / 56.6	70.6 / 53.1	74.0 / 52.9	62.1 / 37.0	71.6 / 53.2
VECO$_{out}$	83.6 / 70.5	65.0 / 44.6	69.8 / 54.6	73.8 / 55.6	69.1 / 51.4	73.1 / 51.8	67.3 / 43.6	71.7 / 53.2
Translate-Train-All								
VECO$_{out}$	84.1/71.3	67.8/47.1	70.7/55.8	74.6/56.6	71.1/53.4	74.8/54.4	68.8/45.8	73.1/54.9
VECO$_{in}$	87.5/75.5	72.3/52.1	75.7/61.1	78.8/61.6	76.6/58.6	79.3/59.1	72.1/46.8	77.5/59.3

Table 15: MLQA results (F1 / EM) for each language.

Model	en	ar	bn	fi	id	ko	ru	sw	te	**Avg.**
Cross-lingual Transfer										
XLM-R	71.5 / 56.8	67.6 / 40.4	64.0 / 47.8	70.5 / 53.2	77.4 / 61.9	31.9 / 10.9	67.0 / 42.1	66.1 / 48.1	70.1 / 43.6	65.1 / 45.0
VECO$_{out}$	71.3 / 58.2	73.1 / 52.8	58.9 / 42.5	70.9 / 55.1	77.2 / 60.0	54.2 / 39.9	66.1 / 37.6	65.8 / 45.7	70.6 / 50.7	67.6 / 49.1
Translate-Train-All										
VECO$_{out}$	77.2/64.8	77.0/57.5	72.2/56.6	76.6/59.3	80.0/64.4	63.4/52.2	72.8/50.5	79.4/67.1	76.0/58.0	75.0/58.9
VECO$_{in}$	79.4/65.2	80.1/60.9	80.8/68.1	81.6/65.5	84.3/69.7	65.4/50.4	77.8/55.8	83.7/74.1	81.0/63.4	79.4/63.7

Table 16: TyDiQA-GolP results (F1 / EM) for each language.

Model	af	ar	bg	bn	de	el	es	et	eu	fa	fi	fr	he	hi	hu	id	it	ja
Cross-lingual Transfer																		
XLM-R	58.2	47.5	71.6	43	88.8	61.8	75.7	52.2	35.8	70.5	71.6	73.7	66.4	72.2	65.4	77	68.3	60.6
VECO$_{out}$	48.2	70.9	86.7	57.7	97.5	81.5	94.8	89.7	62.9	82.1	87.9	88.8	74.7	80.7	87.6	89.6	89.2	83.2
Translate-Train-All																		
VECO$_{out}$	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
VECO$_{in}$	88.5	88.7	91.5	84.2	98.9	91.5	97.9	96.4	85.8	95.3	95.9	95.6	89.6	97.0	95.1	94.2	94.1	94.0

	jv	ka	kk	ko	ml	mr	nl	pt	ru	sw	ta	te	th	tl	tr	ur	vi	zh
Cross-lingual Transfer																		
XLM-R	14.1	52.1	48.5	61.4	65.4	56.8	80.8	82.2	74.1	20.3	26.4	35.9	29.4	36.7	65.7	24.3	74.7	68.3
VECO$_{out}$	17.6	58.5	53.9	75.3	80.1	64.2	94.4	92.8	88.6	37.4	61.9	65.8	84.5	52.5	89.3	64.3	85.8	82.7
Translate-Train-All																		
VECO$_{out}$	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
VECO$_{in}$	49.3	86.6	83.7	91.2	97.1	87.9	97.6	96.1	93.8	82.6	88.9	95.3	95.1	79.8	97.6	91.4	97.2	95.2

Table 17: Tatoeba results (Accuracy) for each language