On-the-fly Cross-lingual Masking for Multilingual Pre-training

Xi Ai
College of Computer Science
Chongqing University
barid.x.ai@gmail.com

Bin Fang
College of Computer Science
Chongqing University
fb@cqu.edu.cn

Abstract

In multilingual pre-training with the objective of MLM (masked language modeling) on multiple monolingual corpora, multilingual models only learn cross-linguality implicitly from isomorphic spaces formed by overlapping different language spaces due to the lack of explicit cross-lingual forward pass. In this work, we present CLPM (Cross-lingual Prototype Masking), a dynamic and token-wise masking scheme, for multilingual pre-training, using a special token $[\mathcal{C}]_x$ to replace a random token x in the input sentence. $[\mathcal{C}]_x$ is an approximation of a cross-lingual prototype for x and then forms an explicit cross-lingual forward pass. We instantiate CLPM for the multilingual pre-training phase of UNMT (unsupervised neural machine translation), and experiments show that CLPM can consistently improve the performance of UNMT models on $\{De, Ro, Ne\} \leftrightarrow En$. Beyond UNMT or bilingual tasks, we show that CLPM can consistently improve the performance of multilingual models on cross-lingual classification.

1 Introduction

With tied weights across the languages and the help of language identifications (Johnson et al., 2017), multilingual models only have access to monolingual corpora in different languages. Stemming from BERT/MLM (Devlin et al., 2019) and GPT (Radford et al., 2018; Alec Radford, 2020), for cross-lingual knowledge, multilingual pre-training with the objective of MLM on multiple monolingual corpora is introduced by XLM (Lample and Conneau, 2019), explored by MASS (Song et al., 2019) and mBART (Liu et al., 2020; Lewis et al., 2020), and scaled by XLM-R (Conneau et al., 2020) and mT5 (Xue et al., 2021).

Essentially, in multilingual MLM pre-training, models are encouraged to learn implicit cross-linguality from both linguistic similarities and shared tokens (Karthikeyan et al., 2020; Wu and Dredze, 2019; Pires et al., 2019; Dufter and Schütze, 2020) for translation and cross-lingual transfer. However, it does not learn any explicit and principled cross-lingual forward pass from inputs to outputs, only relying the isomorphic space emerged from multilingual MLM pre-training by overlapping language spaces agnostically. Given the nature of translation and cross-lingual transfer, models should understand explicit cross-lingual forward passes initiating cross-lingual knowledge directly. Considering this aspect, beyond the *implicit* and *agnostic* cross-linguality, we are interested in the question: can models learn *explicit* and *principled* cross-linguality in multilingual pre-training without any supervision?

Following this idea, for multilingual pre-training, we present a dynamic and token-wise masking scheme, CLPM (Cross-lingual Prototype Masking), to compute a special token $[\mathcal{C}]_x$ representing a cross-lingual prototype for a selected token x and then replace x with $[\mathcal{C}]_x$ instead of the standard token $[\mathcal{M}]$ in multilingual MLM pre-training. We present an example in Table 1. Significantly, when predicting the selected and replaced x, we model an explicit cross-lingual forward pass from the cross-lingual prototype $[\mathcal{C}]_x$ to x.

Source	The investment fund that owned the building had to make a choice .
$[\mathcal{M}]$	The $[\mathcal{M}]$ fund $[\mathcal{M}]$ owned $[\mathcal{M}]$ building $[\mathcal{M}]$ to make a choice .
$[\mathcal{C}]_x$	The $[\mathcal{C}]_{x_1}$ fund $[\mathcal{C}]_{x_3}$ owned $[\mathcal{C}]_{x_5}$ building $[\mathcal{C}]_{x_7}$ to make a choice .

Table 1: Examples of $[\mathcal{C}]_x$ and $[\mathcal{M}]$. $\{x_1, x_3, x_5, x_7\}$ at position $\{1, 3, 5, 7\}$ are randomly selected for replacing. Then, we compute the $[\mathcal{C}]_x$ set $\{[\mathcal{C}]_{x_1}, [\mathcal{C}]_{x_3}, [\mathcal{C}]_{x_5}, [\mathcal{C}]_{x_7}\}$ for replacing and pre-train MLM without any other change, treating $[\mathcal{C}]_x$ as $[\mathcal{M}]$.

Computing $[\mathcal{C}]_x$ is a challenge on monolingual corpora without any supervision from parallel corpora, translation tables (Dufter and Schütze, 2020; Ren et al., 2019b; Chaudhary et al., 2020), or data augmentation processes (Krishnan et al., 2021;

Chaudhary et al., 2020; Tarunesh et al., 2021). Fortunately, we find that suitable candidates can be dynamically searched in the multilingual embedding space, considering the relevance between the selected token and the tokens in the other language. Meanwhile, naive token-to-token relevance is reported to misrepresent morphological variations (Artetxe et al., 2020; Czarnowska et al., 2020; Kementchedjhieva et al., 2020), which limits the improvements for translation and cross-lingual transfer tasks. Thus, we approximate multiple candidates in the other language for $[\mathcal{C}]_x$, expecting to cover morphological variations. Unfortunately, the input dependency is perturbed by $[\mathcal{C}]_x$ because $[\mathcal{C}]_x$ is not agnostic and not static as $[\mathcal{M}]$ but dynamically obtained from the other language. Eventually, it potentially results in the lack of learning internal structures of languages. To alleviate this pain but still use $[\mathcal{C}]_x$, we alternate between $[\mathcal{M}]$ and $[\mathcal{C}]_x$, where $[\mathcal{M}]$ is agnostic and does not perturb input language domain.

We attempt UNMT and (zero-shot) cross-lingual transfer tasks. For UNMT, we consider $En \leftrightarrow X$ on a rich-resource language De, a low-resource language Ro, and a dissimilar language Ne. Intuitively, CLPM yields improvements because of dynamical approximations of token-level cross-lingual information. We then justify this on cross-lingual word similarity tasks from MUSE (Lample et al., 2018b) . Beyond UNMT, we experiment with the cross-lingual classification task on XNLI (Conneau et al., 2018) to test general cross-lingual transfer CLPM improves within a pivoting-based framework.

We have three contributions. *1)* We present CLPM, a dynamic and token-wise masking scheme using special tokens $[\mathcal{C}]_x$, to form cross-lingual forward passes in multilingual pre-training. $[\mathcal{C}]_x$ is a generalized representation from multiple cross-lingual candidates. *2)* CLPM substantially improves the performance of $En \leftrightarrow X$ baseline UNMT models by $3\% \sim 8\%$ on rich-resource and low-resource languages and can facilitate training on dissimilar languages. *3)* Beyond UNMT tasks or bilingual tasks, CLPM can be used for cross-lingual classification tasks.

2 Cross-lingual Prototype Masking

Notation L_x is the language ID of language $Lang_x$. P_n stands for positions. E_R is the embedding for R. d is model/embedding dimension.

2.1 Forward Pass in Attention

Given an input sentence $X = \{x_0, x_1, ..., x_n\}$ in language $Lang_x$, the self-attention layer (Vaswani et al., 2017) performs on the sum of $X_{input} = \{E_{x_0} + E_{L_x} + E_{P_0}, ..., E_{x_n} + E_{L_x} + E_{P_n}\}$, similar to that is used in previous works of multilingual pre-training (Liu et al., 2020; Song et al., 2019; Lample and Conneau, 2019). For predicting x_i, the attention score (Bahdanau et al., 2015; Luong et al., 2015) $e_{i,j} = (E_{x_i} + E_{L_x} + E_{P_i})^T W_q^T W_k (E_{x_j} + E_{L_x} + E_{P_j})$ between query vector q_i and key vector k_j within the same sentence can be decomposed:

$$e_{i,j} = \underbrace{E_{x_i}^T W_q^T W_k E_{x_j}}_{a} + \underbrace{E_{L_x}(\cdot)}_{b} + \underbrace{E_{P_i}(\cdot)}_{c} + \underbrace{E_{P_j}(\cdot)}_{d} \quad (1)$$

where W_q and W_k are linear transformation for the query vector q_i and key vector k_j respectively, and i and j stands for position indexes. Terms (b), (c), and (d) introduce the inductive bias towards language $Lang_x$, position P_i, and position P_j respectively. When predicting x_i, we have the forward pass: $\{x_i, x_{j\setminus i}\} \rightarrow x_i$, where $x_{j\setminus i}$ denotes all the tokens around position i, and the prediction of x_i is conditioned by $\{x_i, x_{j\setminus i}\}$. The forward pass is *monolingual* because both the two sides are in the same language. In optimization, we can compute gradients from the backward pass: $\frac{\partial \varepsilon_{x_i}}{\partial E_{x_i}}$ and $\frac{\partial \varepsilon_{x_i}}{\partial E_{x_j}}$, where ε_{x_i} is the predicting error.

2.2 MLM with $[\mathcal{M}]$ and CBOW

Suppose x_i is randomly selected to be replaced by $[\mathcal{M}]$. Term (a) is changed to $E_{[\mathcal{M}]}^T W_q^T W_k E_{x_j}$. Since $[\mathcal{M}]$ does not provide prior information of x_i, Term (a) forms a built-in CBOW [1] model (Continuous Bag-of-Words (Mikolov et al., 2013)) learning CBOW or bidirectional information. The forward pass $\{[\mathcal{M}], x_{j\setminus i}\} \rightarrow x_i$ is still *monolingual* in multilingual pre-training because $[\mathcal{M}]$ is shared and agnostic for all the languages. However, the model is significantly encouraged to predict x_i by understanding neighboring tokens $x_{j\setminus i}$ in the sentence, i.e., the surrounding context or bidirectional information. Moreover, since $[\mathcal{M}]$ is overlapping and shared, and $x_{j\setminus i}$ are potentially overlapping tokens in different languages, it refines the morphology of

[1] For instance, given $X = \{x_0, [\mathcal{M}], x_2, x_3\}$, we have the forward pass: $\{x_i = [\mathcal{M}], x_{j\setminus i} = (x_0, x_1, x_3)\} \rightarrow x_2$ if predicting x_2, where $\{x_i = [\mathcal{M}], x_{j\setminus i} = (x_0, x_1, x_3)\}$ models (non-standard) CBOW (4-gram).

different languages to overlap each other for forming the isomorphic spaces (Karthikeyan et al., 2020; Wu and Dredze, 2019; Pires et al., 2019; Dufter and Schütze, 2020) and leverages domain adaptation (Ganin et al., 2016) or language adaptation (Ai and Fang, 2022b).

2.3 MLM with $[\mathcal{C}]_x$

Although the forward pass $\{[\mathcal{M}], x_{j\backslash i}\} \rightarrow x_i$ significantly enables the model to learn both cross-lingual and monolingual knowledge from the shared token $[\mathcal{M}]$ (Dufter and Schütze, 2020) and structural information of the neighboring tokens $x_{j\backslash i}$ (Karthikeyan et al., 2020; Pires et al., 2019) in multilingual MLM pre-training, learning cross-linguality is *implicit and limited*. Our idea is, we can replace $[\mathcal{M}]$ with x_i's cross-lingual prototype $[\mathcal{C}]_{x_i}$ that we explicitly have a principled *cross-lingual* forward pass: $\{[\mathcal{C}]_{x_i}, x_{j\backslash i}\} \rightarrow x_i$. In this way, we inject weak but explicit cross-lingual supervision to the model in multilingual pre-training. Therefor, we replace the selected x_i with its $[\mathcal{C}]_{x_i}$ instead of $[\mathcal{M}]$ as presented in the example (Table 1), and Term (a) is modified to $E^T_{[\mathcal{C}]_{x_i}} W^T_q W_k E_{x_j}$ accorddingly.

2.4 On-the-fly $[\mathcal{C}]_x$

To obtain $[\mathcal{C}]_{x_i}$ without any cross-lingual supervision, the starting point is the output distribution over the vocabulary V shared by all the languages. Given the multilingual model Net, we set Net to *the inference mode*, not the MLM pre-training mode, and the probability of x_i is obtained from the $softmax$ layer $Q_{x_i} = \frac{exp(h^T_{x_i\&L_x} O_{x_i})}{\sum_{k=1}^{V} exp(h^T_{x_i\&L_x} O_{x_k})}$, where $h_{x_i\&L_x} \in Net(E_x + E_{L_x})$, $E_x = \{E_{x_0}, E_{x_1}, \ldots, E_{x_n}\}$ is the embedding of the input sentence, and O_x is factorized from the output matrix O[2]. Recall that, in Eq. 1, E_{L_x} associated with tokens introduces inductive bias towards $Lang_x$ for x, so that $h_{x_i\&L_x}$ is biased by E_{L_x} towards $Lang_x$ and generalized from E_{x_i}. In this way, the output distribution over the vocabulary is biased by E_{L_x} towards $Lang_x$, and the dot-products distinguish relevant tokens from irrelevant tokens for x_i. Intuitively, we can fool the model by inputting $E_x + E_{L_y}$[3]. The result is that $h_{x_i\&L_y} \in Net(E_x + E_{L_y})$ is biased by E_{L_y} towards $Lang_y$ but still generalized from E_{x_i}. We expect $h_{x_i\&L_y}$ is an agnostic representation that is relevant to x_i but $Lang_y$. Then, we can rank the dot product $h^T_{x_i\&L_y} E_y$ to search relevant tokens for x_i in $Lang_y$ from the embedding space. We will discuss the inspiration later, and in our experiment, we show a case study that some useful candidates in the other language are obtained.

We approximate a relevant candidate set $P^Y_{x_i}$ in the other language $Lang_y$ and compute a weighted average of candidates' embeddings, where $P^Y_{x_i}$ contributes to low variance and rich information. Formally, we define $E_{[\mathcal{C}]_x} = \sum_{y \in P^Y_x} E_y W^y_x$, where$P^Y_x \subset Voc_Y$, Voc_Y is the entries of the other language in the multilingual embedding space, $0 \leq W^y_x \leq 1$ is the weight of the candidate $y \in P^Y_x$ and $\sum_{y \in P^Y_x} W^y_x = 1$. Given the model Net, we have 4 steps to compute $[\mathcal{C}]_x$ dynamically:

- **Step 1:** We set Net to *the inference mode* $\tilde{Net}$, input $E_x + E_{L_y}$ to $\tilde{Net}$, and obtain the representation $h_{x_i\&L_y} \in \tilde{Net}(E_x + E_{L_y})$ for the selected token x_i.
- **Step 2:** We calculate a full-sized set $Q = (h^T_{x_i\&L_y} O_{y_0}, ..., h^T_{x_i\&L_y} O_{y_v})$, where v equals the size of Voc_Y.
- **Step 3:** We select a candidate set $P^Y_x = (E_{y^j}, ..., E_{y^k})$ from the embedding space, according to the Top-K dot products in Q.
- **Step 4:** W compute a weight set $W^Y_x = softmax(E^T_{y^i} E_x, ..., E^T_{y^v} E_x)$ and the output $E_{[\mathcal{C}]_x} = \sum_{y \in P^Y_x} E_y W^y_x$.

Note that, to select tokens for Voc_Y, the minimum frequency is $1e - 5$ in the monolingual corpora of $Lang_y$. Meanwhile, some tokens are shared among different languages. We set the minimum frequency of shared tokens to $1e - 3$ in the monolingual corpora. These settings are used to limit the searching bound for more meaningful candidates.

Inspiration Our recipe takes the inspiration from early experiments. We pre-train a small multilingual model (12 layers and 256 d) and use our recipe to search for candidates. As presented in Table 2, a multilingual model can infer some cross-lingual candidates with our recipe because of the cross-lingual transfer phenomenon, and we can generalize these candidates for cross-lingual prototypes. Meanwhile, we are aware that the multilingual model has to be pre-trained or properly initialized

[2] Note that, in most of cases, the output matrix shares all the parameters with the embedding matrix.

[3] Empirical studies and alternatives of $E_x + E_{L_x}$ and E_x's nearest neighbors are presented in Appendix C.1.

	400k step training	50k step training
#1	It was hampered by the need for ranges to be estimated by eye , which introduced significant in@@ accuracy . [EOS]	
Reference	Erschwert wurde dies durch die Notwendigkeit , Entfernungen mit dem Auge abzuschätzen, was zu erheblichen Ungenauigkeiten führte . [EOS]	
Masked	It $[\mathcal{C}]_{x_1}$ hampered $[\mathcal{C}]_{x_3}$ $[\mathcal{C}]_{x_4}$ need $[\mathcal{C}]_{x_6}$ ranges to $[\mathcal{C}]_{x_9}$ estimated by eye , which $[\mathcal{C}]_{x_{15}}$ significant $[\mathcal{C}]_{x_{17}}$ $[\mathcal{C}]_{x_{18}}$. [EOS]	
was = $[\mathcal{C}]_{x_1}$	**war**, **wurde**, brach	., und, als
by = $[\mathcal{C}]_{x_3}$	in, , ,**durch**	in, **von**,
the = $[\mathcal{C}]_{x_4}$	**den**, **die**, **der**	., **den**, einem
for = $[\mathcal{C}]_{x_6}$	**für**, in, **dafür**	in, ., **und**
be = $[\mathcal{C}]_{x_9}$	des, ,, ben	stellt, Bau, einem
introduced = $[\mathcal{C}]_{x_{15}}$	lehnte, Schwei@@, löste	in, /, von
in@@ = $[\mathcal{C}]_{x_{17}}$	in, Gebäude@@, @-@	in, (@-@
accuracy =$[\mathcal{C}]_{x_{18}}$	Seh@@ ographie Bewertung	geber, er, studium
#2		Sie befindet sich auf 425 Meter Höhe nahe dem Schlos@@ sberg . [EOS]
Reference		It is located at an altitude of 425 meters near the Schlossberg. [EOS]
Masked		$[\mathcal{C}]_{x_0}$ $[\mathcal{C}]_{x_1}$ sich auf 425 $[\mathcal{C}]_{x_5}$ $[\mathcal{C}]_{x_6}$ $[\mathcal{C}]_{x_7}$ dem Schlos@@ sberg . [EOS]
Sie = $[\mathcal{C}]_{x_0}$	**It**, **She**, **He**	leaves, breaks, Geography
auf = $[\mathcal{C}]_{x_4}$	**in**, **at**, **on**	the, a, 29@@
Meter = $[\mathcal{C}]_{x_5}$	**metres**, **meters**, **feet**	@-@, ,, in
Höhe = $[\mathcal{C}]_{x_6}$	**altitude**,**height**, **elevation**,	,, in, an
nahe = $[\mathcal{C}]_{x_7}$	**near**, **Near**, **close**	**in**, an, ,

Table 2: Inspiration of $[\mathcal{C}]_x$ from multilingual training. References are obtained from Google Translation. We use a pre-trained small XLM model on $\{En, De\}$. To obtain more examples we randomly compute $[\mathcal{C}]_x$ for 40% of tokens. @@ is the continuing subword prefix. **bold** denotes a strong candidate that is a parallel, analogical, or relevant token/word (or its variation) in other languages. Our method can cover multiple morphological or relevant candidates(e.g., "den", "die", "der" in #1 $[\mathcal{C}]_{x_4}$) for generalizing information by weighted average.

in order to infer cross-lingual candidates by itself. We will discuss initialization later.

2.5 Alternation between $[\mathcal{M}]$ and $[\mathcal{C}]_x$

In our experiment (see row 12 $\sim$ 15 of Table 7 in Appendix), we find that we can get benefits from alternating between $[\mathcal{M}]$ and $[\mathcal{C}]_x$. Intuitively, only using $[\mathcal{C}]_x$ might perturb bidirectional knowledge and result in the lack of monolingual n-gram statistics, whereas the model can learn from using $[\mathcal{M}]$ in multilingual MLM pre-training. We also note similar observations in previous works (Chaudhary et al., 2020; Ren et al., 2019a), which use translation tables for pre-training. Another side effect we observe is that the model might pay much more attention to "prototype-word" translation knowledge instead of understanding bidirectional knowledge. Thus, to encourage the model to learn both strong bidirectional knowledge from $[\mathcal{M}]$ and cross-lingual knowledge from $[\mathcal{C}]_x$, in $t\%$ of the time of the MLM pre-training time, we use $[\mathcal{C}]_x$ for masking. For the remaining $(100 - t)\%$ of the time, we still use $[\mathcal{M}]$. Hence, we have dual objectives in multilingual MLM pre-training: $\mathcal{L}_{MLM} = \mathcal{L}_{CLPM} + \mathcal{L}_{MASK}$. With this dual objectives in mind, we can simply extend the MLM's masking strategy to: $([SAME], [RAN], [\mathcal{M}], [\mathcal{C}]_x)$ with $(10\%, 10\%, (80 - t)\%, t\%)$.

2.6 Discussion

We discuss some import components of our method. For these discussions, we provide empirical studies and show the observation of these components in §Robustness and Model Variation.

$[\mathcal{M}]$ vs. $[\mathcal{C}]_x$ *1)* $[\mathcal{M}]$ is static in the embedding space with an explicit entry, used by running a lookup operation. Meanwhile, it is used to replace all randomly selected tokens, which is *unified. 2)* In contrast, $[\mathcal{C}]_{x_i}$ or $E_{[\mathcal{C}]_{x_i}}$ is dynamically approximated during training, which is *token-wise*.

Choice of K *1)* The memory usage is proportional to the size of K. Meanwhile, large K potential increases noise for unambiguous $[\mathcal{C}]_x$. *2)* On the other hand, a small K may reduce the searching bound that computing proper $[\mathcal{C}]_x$ is hard. For instance, $K = 1$ only yields median improvements in our experiment. Our empirical study shows that it is robust to a range of K from 2 to 5, considering a trade-off between GPU memory problems and expected performance improvements.

Initialization The random initialization may raise problems. *1)* x may find some geometric close but irrelevant tokens with large dot products (Step 2) in Voc_Y, which results in a trivial candidate set (Step 3). *2) Inference mode* with random initialization is trivial. To this end, we only pre-train the multilingual model by MLM with $[\mathcal{M}]$ at the first several iterations for warm-up to form the multilingual embedding space and activate the *inference mode*, as discussed in §Inspiration. After the warm-up, the multilingual embedding space and model are initialized in a few-shot style somewhat to avoid trivial candidates. Then, we run the alternation. In our experiments, we find that this warm-up can help the model obtain new samples with cross-lingual prototypes from the other language.

Efficiency On-the-fly $[\mathcal{C}]_x$ will increase the training time. However, only a sub-set of tokens (typically, 15% (Devlin et al., 2019)) of the input text stream is selected for masking, and we only need to compute $[\mathcal{C}]_x$ for a sub-set of all the selected token. In our experiment, we find our method spends additional $\approx 15\%$ time on training.

Tokenization Tokenizations generating “middle" tokens, sub-tokens or non-standard word tokens might impact $[\mathcal{C}]_x$, e.g., BPE. However, the impact is relatively small given that: *1)*: the vocabularies and monolingual corpora are dominant by the standard words rather than non-standard word token, e.g., over 50% BPE vocabulary for translation task $De \leftrightarrow En$) are standard words and they make up for over the 80% of the total token frequency in the monolingual corpora; *2)*: all the representations are contextualized that sub-tokens and non-standard word tokens still represent semantics and syntactic meanings related to their original standard words (Levine et al., 2021; Ai and Fang, 2022a) (refer to the case study in Appendix C.2).

3 Empirical Study and Experiment

All the links of datasets, libraries, scripts and tools marked with ⋄ are listed in Appendix F. A preview version of code is submitted, and we will open source code on GitHub.

Pre-training Setting We use Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$,$\beta_2 = 0.999$, $\epsilon = 1e-8$, $warm\_up$ step (Vaswani et al., 2017) and $lr = 1e-4$. Dropout regularization is set to $rate = 0.1$. Readers can refer to Appendix D.1 for details.

Model Configuration Our Transformer model (Vaswani et al., 2017) is identical to XLM (Lample and Conneau, 2019), which consists of a 6-layer encoder and 6-layer decoder with 1024 word embedding size and hidden size and 4096 feed-forward filter size. We add a learnable language embedding and a learnable position embedding to each token of the input sentence for the encoder and decoder (P and L in Eq.1). We have some default configurations for our method base on the study of model robustness (see §Robustness and Model Variation): *1)* $t\% = 40\%$ that we make a balance between the two objectives: $[\mathcal{M}]$ and $[\mathcal{C}]_x$; *2)* $K = 3$ that we consider top-3 candidates for the cross-lingual prototypes; *3)* the warm-up step is 50k that $[\mathcal{M}]$ is only used at the first 50k iterations; *4)* we consider BPE for tokenization in all our experiments.

Multilingual Task We consider three multilingual tasks: **1)** UNMT for evaluation on translation tasks, **2)** cross-lingual semantic word similarity for evaluation on cross-lingual embedding tasks, and **3)** zero-shot cross-lingual classification for evaluation on cross-lingual transfer tasks.

3.1 MLM Instance

We adapt our method to three MLM instances to pre-train the multilingual model:**1)** XLM (Lample and Conneau, 2019), **2)** MASS (Song et al., 2019), and **3)** mBART (Liu et al., 2020), which can be used to pre-train a multilingual model. Readers can refer to the original report or Appendix D.2 for more instructions of these MLM instances. Significantly, to minimize changes for evaluation and comparison, we only have two changes. The first change we make is extending the masking strategy: $([SAME], [RAN], [\mathcal{M}])$ with $(10\%, 10\%, 80\%)$ to $([SAME], [RAN], [\mathcal{M}], [\mathcal{C}]_x)$ with $(10\%, 10\%, (80 - t)\%, t\%)$. Secondly, as presented in Table 1, we only apply CLPM to the input of the source side or the encoder and do not change the shifted input of the decoder in these MLM instances. Any other component is identical to the reported MLM instances.

We reimplement all the baseline models on our machine with our configurations, using official XLM⋄, Tensor2Tensor⋄, and HuggingFace⋄ as references. We compare the results of our reimplementation with the reported results on the same test set to ensure the difference less than 2% in overall performance (see Appendix E for result comparison). Then, we can confirm our reimplementation.

3.2 UNMT

Setup We consider similar language pairs $\{De, Ro\} \leftrightarrow En$, using the same dataset and test set as previous works (Lample and Conneau, 2019). Meanwhile, we share the FLoRes⋄ (Guzmán et al., 2019) task to evaluate on a dissimilar language pair $Ne \leftrightarrow English$ (Nepali). We learn shared BPE (Sennrich et al., 2016b), selecting the most frequent 60K codes from paired languages with the same criteria in (Lample and Conneau, 2019). The model is pre-trained around 400K iterations on

Language pair	$De \leftrightarrow En$		$Ro \leftrightarrow En$		$Ne \leftrightarrow En$	
multi-BLEU.perl◇ with default rules						
XLM(Lample et al., 2018c)	34.3	26.4	31.8	33.3	0.5	0.1
+ word translation tables (Chaudhary et al., 2020) ⋆	35.1	27.4	33.6	34.4	4.1	2.2
+ $[\mathcal{C}]_x$	35.9	28.1	34.4	35.3	6.6	2.8
MASS(Song et al., 2019)	35.2	28.3	33.1	35.2		
+ nearest neighbor from UBWE (Dufter and Schütze, 2020) ⋆	36.1	28.8	34.1	36.4	5.1	2.8
+ $[\mathcal{C}]_x$	36.7	29.2	34.7	36.9	7.1	3.4
sacreBleu◇ with standard settings: nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0						
mBART(Liu et al., 2020) + CC25 (Wenzek et al., 2020)	34.0	29.8	30.5	35.0	10.0	4.4
+ $[\mathcal{C}]_x$ (w/o CC25)	35.4	30.1	32.5	36.7	7.0	3.2

Table 3: Performance of UNMT. ⋆ are reimplemented. UBWE stands for unsupervised bilingual word embedding. Translation tables or UBWE are static. We use the same transformer models, BPE size, corpora, tokenization, and BLEU as the baseline models (see more details in Appendix D.3).

only monolingual corpora in different languages. And, after around 400K training iterations for translation with the standard pipeline◇ (Artetxe et al., 2018b; Song et al., 2019), according baseline models' BLEU scripts, we report BLEU computed by *multi-BLEU.perl*◇ or *sacreBleu*◇ (Post, 2018) with default rules. See more details in Appendix D.3.

Result Table 3 shows the results on the $\{De, Ro, Ne\} \leftrightarrow En$ test sets. Applying $[\mathcal{C}]_x$ consistently improves the performance of baseline models on all the similar language pairs by $3\% \sim 8\%$ and on the dissimilar pair by $2.5 \sim 7$ BLEU. The performance on the dissimilar pair is very close to SOTA: mBART25 (Liu et al., 2020), but they use 25 languages from CC25 (Wenzek et al., 2020) for pre-training. Surprisingly, our method slightly outperforms two dictionary-based works (Dufter and Schütze, 2020; Chaudhary et al., 2020) which require static translation tables from pre-trained word models, golden dictionaries, or bilingual lexicon induction (e.g., UBWE). Intuitively, as reported in (Artetxe et al., 2020; Kementchedjhieva et al., 2019; Czarnowska et al., 2019; Vania and Lopez, 2017), such word translation tables are reported to misrepresent morphological variations and are not contextualized properly, which limit the improvements for sentence translation.

For further analyses, we conduct a case study to observe the attention weights on $[\mathcal{C}]_x$ after pre-training, which is visualized in Appendix C.2. We observe that the model outputs prominent attention weights on $[\mathcal{C}]_x$ for predicting replaced tokens, so that it relies on $[\mathcal{C}]_x$, whereas the mode only using $[\mathcal{M}]$ can only rely on neighboring tokens. We can confirm the effectiveness. Concretely, CLPM shows significant effectiveness on nouns, entities, terminology words, etc., where the attention weights on the corresponding $[\mathcal{C}]_x$ are dominant. Meanwhile, the model can understand phrases, sub-tokens, and syntax structures to predict a replaced token of the phrase because the model pays equal/similar attention to each token of the phrase. We attribute this phenomena to both the alternation between $[\mathcal{C}]_x$ and $[\mathcal{M}]$ and involving neighboring x_j in $\{[\mathcal{C}]_{x_i}, x_{j\setminus i}\} \rightarrow x_i$ that the model captures token dependencies from the cross-lingual prototype or a synonym in the other language. Finally, the employment of multiple candidates is important because the model could learn morphological or relevant variations from $[\mathcal{C}]_x$ in the other language (refer to Appendix C.1),e.g., understanding relevant variations <**welches**, **welcher**, **welche**> from $[\mathcal{C}]_x$, which is essential for further translation learning in unsupervised manners.

Dose CLMP introduce new samples with cross-lingual prototypes from the other language? In addition to §Case Study, we are still interested in the representation of $E_{[\mathcal{C}]_x}$ or whether CLMP introduces new examples with cross-lingual prototypes from the other language. Intuitively, if the weights obtained in Step 4 are $\{c_1 = 0.9, c_2 = 0.05, c_3 = 0.05\}$, the representation is similar to the candidate c_1, and then c_1 is an only soft translation of x. If the weights are $\{c_1 = 0.4, c_2 = 0.3, c_3 = 0.3\}$, the representation could be different from any one of $\{c_1, c_2, c_3\}$. Thus, the representation depends on the contributions of the candidates. To further understand $E_{[\mathcal{C}]_x}$, we jointly train a discriminator to distinguish between two languages in the pre-training phase. The discriminator is trained to recognize which language an embedding belongs to. Then, we make zero-shot classification for $E_{[\mathcal{C}]_x}$ to observe which language $E_{[\mathcal{C}]_x}$ belongs to. We report the result in Figure 1. This figure suggests that CLMP introduces unseen cross-lingual prototypes for the model. We suspect that $E_{\mathcal{C}_x}$ potentially yields a generalized representation from

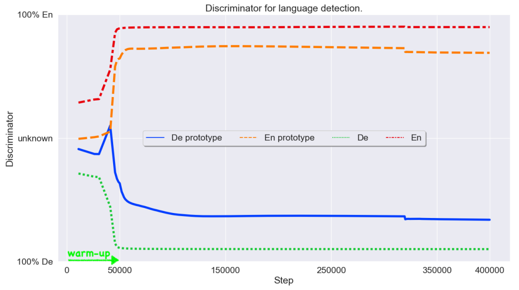


Figure 1: Discriminator performance. The discriminator with the $sigmoid$ activation is trained to recognize which language an embedding belongs to and makes zero-shot classification for a prototype. This figure indicates that CLMP introduces unseen cross-lingual prototypes for the model instead of embedding instances.

multiple relevant candidates in other languages. This is different from the method family based on translation tables. Significantly, translation tables are instances/embeddings in the embedding space, whereas cross-lingual prototypes do not exist in the embedding spaces and are new generalized samples for the model.

3.3 Robustness and Model Variation

We have some default configurations, as presented in row 2 of Table 4. This combination is obtained in our experiments. We report the results to observe the impact of K (the number of cross-lingual candidates), the warm-up initialization, the tokenization method, and the alternation $t\%$ in Appendix B. Meanwhile, in this experiment, we discuss a mean average style for cross-lingual candidates instead of the weighted average used in the default configuration, reporting results in Appendix B. Additionally, we study alternatives for initialization and training efficiency. The result is presented in Table 7. For consistence, the row number is consistent with the full results in Appendix B.

Row 11 As aforementioned, CLPM requires additional time to compute $[\mathcal{C}]_x$. To be fair, we reduce the training steps, so that the training time is almost similar to the baseline model (row 1). CLPM outperforms the baseline model but requires less training steps, which indicates that the explicit and principled cross-lingual forward pass is more efficient (per step) than implicit isomorphic space formation for cross-linguality.

Row 17 We use UBWE (unsupervised bilingual word embedding) to initialize the bilingual embedding space. In the first 50k pre-training steps (equal to default warm-up steps), since the model parameters are still randomly initialized, we do not follow Step 1, 2, and 3 in on-the-fly $[\mathcal{C}]_x$ and directly find relevant candidates based on the dot products $E_{y^i}^T E_x$, i.e., only need Step 4. Intuitively, $E_{y^i}^T E_x$ is reliable to rank the candidates and computes the weights for $[\mathcal{C}]_x$ because UBWE provides cross-lingual entries. After 50k pre-training steps, we normally run on-the-fly $[\mathcal{C}]_x$. We observe that adapting UBWE consistently improves the performance by 2% on the similar language and $0.5 \sim 1$ BLEU on the dissimilar language because UBWE provides additional cross-lingual supervision. See all the results in Table 8.

Row 18 Vulić et al. (2020) suggest seed dictionaries for unsupervised tasks in practice. Following this idea, we download a 1k seed dictionary from Panlex◇. In the first 50k pre-training steps, we simply replace the selected token with its translation in the seed dictionary. For the out-of-the-dictionary but selected token, we replace it with normal $[\mathcal{M}]$. After 50k pre-training steps, if the selected token is in the dictionary, the translation is added to $[\mathcal{C}]_x$ as a candidate in Step 4 when running on-the-fly $[\mathcal{C}]_x$. We find that compared to the UBWE scenario, this adaptation achieves similar results on the rich-resource language $De \leftrightarrow En$ (+ 1.5%) but stronger results on the dissimilar language $Ne \leftrightarrow En$ (+ 8%). All the results are presented in Table 8.

3.4 Cross-lingual Semantic Word Similarity

Setup Given the idea of our method, we consider cross-lingual mappings of tokens. Therefore, we are interested in the isomorphism of languages' embedding spaces. To further investigate, the pre-trained UNMT model is evaluated on MUSE◇ (Lample et al., 2018b) with the provided test sets and tools, which is used to test cross-lingual word similarities on $En \leftrightarrow De$. This test can generally evaluate the degree of the isomorphism of languages' embedding spaces. We reuse the pre-trained models in our UNMT experiment. After restoration, we extract words required by the test set via shared lookup tables. For words split into 2+ sub-tokens, we average all the sub-tokens.

Result We evaluate the performance by similarities, reporting the result in Table 5. Applying $[\mathcal{C}]_x$ can increase the similarities of parallel words

Row	Model	t	Tokenization	Warm-up	Steps	K	$[\mathcal{C}]_x$ type	$De \leftrightarrow En$	
1	$[\mathcal{M}]$ (baseline)	-	BPE	-	400K	-	-	34.3	26.4
2	$[\mathcal{C}]_x$ (our baseline, default)	40%	BPE	50K	400K	3	weighted	35.9	28.1
11	$[\mathcal{C}]_x$	+	+	+	350K (similar training time)	+	+	35.1	27.2
17	$[\mathcal{C}]_x$	+	+	UBWE	+	+	+	36.5	28.8
18	$[\mathcal{C}]_x$	+	+	1k seed dictionary	+	+	+	36.9	29.1

Table 4: Model Variation. For consistence, the row number is consistent with the full results (including evaluation on K, warm-up, tokenization, and $t\%$) in Appendix B. All the models are based on XLM instance. Row 2 shows the default configurations we use in UNMT. $+$ denotes the default configuration. $-$ denotes an inapplicable term. *UBWE* denotes that we pre-train the bilingual embeddings unsupervisedly and then pre-train the entire model with our method. In *1k seed dictionary* test, the model employs a candidate from a seed dictionary.

MUSE	score
XLM(Lample and Conneau, 2019)	0.55
+$[\mathcal{C}]_x$	0.61
MASS(Song et al., 2019)⋆	0.60
+$[\mathcal{C}]_x$	0.64
mBART(Liu et al., 2020)⋆	0.59
+$[\mathcal{C}]_x$	0.64

Table 5: Performance on MUSE task. Baseline models (⋆) are reimplemented with our configurations.

Model	Avg (Acc.)
mBERT *baseline* (Wu and Dredze, 2019)	66.3
XLM (Lample and Conneau, 2019)	71.5
+ word translation tables(Chaudhary et al., 2020)	72.7
+ $[\mathcal{C}]_x$	74.0
+ MT (Lample and Conneau, 2019)	75.1

Table 6: Performance of cross-lingual classification on XNLI. MT stands for additional parallel corpora. We use the same transformer models, BPE size, corpora, tokenization, and BLEU as the baseline models (see more details in Appendix D.3).

from $\{En, De\}$, consistently improving the performance of the models on this task. It indicates that $[\mathcal{C}]_x$ helps the models to learn token-level cross-linguality in pre-training.

3.5 Cross-lingual Classification

Setup Beyond UNMT tasks or translation tasks, CLPM can consistently improve cross-lingual transfer. Then, we attempt the cross-lingual classification task on XNLI (Conneau et al., 2018) to test general cross-linguality $[\mathcal{C}]_x$ improves. For this test, we follow the standard and basic experiment (Lample and Conneau, 2019) to train a 12-layer Transformer encoder with 80k BPE on Wikipedia dumps⋄ of 15 XNLI languages. To pre-train the encoder on En corpora, considering the zero-shot classification based on finetuing En NLI dataset, we randomly compute $[\mathcal{C}]_x$ from other languages with equal probability to avoid the cross-lingual bias. For pre-training on corpora of other languages, we only compute $[\mathcal{C}]_x$ in the En domain. Note that, although we have different strategies of $[\mathcal{C}]_x$ for the languages, we still concatenated all the corpora of the languages for joint pre-training. After pre-training, we deploy a randomly initialized linear classifier and finetune the encoder and the linear classifier on the En NLI dataset with mini-batch size 16. We make zero-shot classifications for other languages. See more details in Appendix D.3.

Result We report the result in Table 6. CLPM shows effectiveness on this task, outperforming baseline models. It indicates that $[\mathcal{C}]_x$ can improve cross-lingual transfer. Meanwhile, $[\mathcal{C}]_x$ underperforms XLM + MT that uses parallel corpora to improve cross-linguality. As discussed earlier, $[\mathcal{C}]_x$ can provide token-level cross-lingual knowledge at the very least but is less effective than golden sentence-level knowledge. Although XLM + MT uses additional datasets, it somewhat sets an upper bound. On the other hand, our method outperforms dictionary-based methods (+ word translation tables). Similar to the observation in UNMT, we attribute to the effectiveness of using multiple candidates to capture morphological variations. However, to avoid cross-lingual bias, we use En as a pivot or anchor point. This could be a potential problem for further adaptation to other multilingual tasks. See limitations in Appendix A.

4 Related Work and Comparison

(Ren et al., 2019a; Chaudhary et al., 2020; Lample et al., 2018c) leverage translation tables as entries for the other languages, which are automatically generated from statistical models, e.g., n-gram models. The model forms an explicit cross-lingual forward pass: $\{[\mathcal{M}], x_{j\backslash i}\} \to t_i$, where t_i is the entry of the other language for x_i. In contrast, our method has two significant differences:

1) we focus on the left side, adapting our $[\mathcal{C}]_x$ to the inputs of MLM; *2)* our method does not rely on token/phrase-level translation tables. Dufter and Schütze (2020) present a cross-lingual forward pass: $\{nn, x_{j\setminus i}\} \rightarrow x_i$, where nn is x_i's nearest neighbor of the other language in the space of UBWE. However, UBWE is static and fixed without any interaction with the multilingual model. It might limit what it can be ultimately used for translation (Sun et al., 2019; Artetxe et al., 2018b; Lample et al., 2018a). We present a dynamic approach obtaining candidates of the other language from the model itself, which is inspired by (Sun et al., 2019; Ai and Fang, 2021b). The benefit is that embeddings and representations are contextualized when pre-training MLM on monolingual corpora in different languages (Lample and Conneau, 2019). Although it is not reliable at the very early of pre-training, we provide a compromised initialization for this problem. We also consider multiple candidates for cross-lingual prototypes instead of nn, which is softer and can cover morphological or relevant variations in the other language. On the other hand, considering cross-lingual prototypes is not a novel idea for cross-linguality, (Wang et al., 2019; Huang et al., 2019; Ai and Fang, 2021a) present methods to leverage cross-lingual prototypes to guide encoding and decoding, forming a cross-lingual forward pass by modifying inner representations of encoding and decoding: $\{[\mathcal{M}], x_{j\setminus i}\} \rightarrow \{[\mathcal{M}], h_{x_j}, h_{y_i}\} \rightarrow x_i$, where h_{y_i} is an approximation of x_i's inner representation in encoding and decoding from the other language. It results in a different direction.

We also employ the alternation strategy that can be viewed as linguistic code-switching (Scotton and Ury, 1977) somewhat, where the model is pre-trained in more linguistic varieties. In learning models, linguistic code-switching performs as data augmentation processes (Krishnan et al., 2021; Chaudhary et al., 2020; Tarunesh et al., 2021) with the help of static translation tables or lexicon induction in supervised manners. However, lexicon induction datasets or translation tables have been reported to misrepresent morphological variations and overly focus on named entities and frequent words (Artetxe et al., 2020; Czarnowska et al., 2020; Kementchedjhieva et al., 2020). In contrast, CLPM is dynamic and unsupervised, leveraging contextualized embeddings and multiple morphological variations in the model's embedding space. Meanwhile, translation tables are instances/embeddings in the embedding space, whereas cross-lingual prototypes do not exist in the embedding spaces and are new generalized samples for the model. This distinction is observed from the discriminator in Figure 1.

5 Conclusion

In this work, we present CLPM, an alternative masking scheme, to compute special tokens $[\mathcal{C}]_x$ for masking in multilingual MLM pre-training. $[\mathcal{C}]_x$ is the cross-lingual prototype for the selected word x, computed from multiple candidates dynamically and token-wise. Compared to the standard masking scheme $[\mathcal{M}]$, $[\mathcal{C}]_x$ automatically forms an explicit cross-lingual forward pass in attention mechanism, consistently improving cross-linguality in multilingual MLM pre-training. Experiments show that CLPM can consistently improve the performance of translation and cross-lingual transfer.

References

Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.

Xi Ai and Bin Fang. 2021a. Almost free semantic draft for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3931–3941.

Xi Ai and Bin Fang. 2021b. Empirical regularization for synthetic sentence pairs in unsupervised neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12471–12479.

Xi Ai and Bin Fang. 2022a. Leveraging relaxed equilibrium by lazy transition for sequence modeling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2904–2924, Dublin, Ireland. Association for Computational Linguistics.

Xi Ai and Bin Fang. 2022b. Vocabulary-informed Language Encoding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4883–4891.

Rewon Child David Luan Dario Amodei Ilya Sutskever Alec Radford, Jeffrey Wu. 2020. [GPT-2] Language Models are Unsupervised Multitask Learners. *OpenAI Blog*, 1(May):1–7.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018b. Unsupervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. 2020. A Call for More Rigor in Unsupervised Cross-lingual Learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7375–7388. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Ond rej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ond rej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (wmt18). In *Proceedings of the Third Conference on Machine Translation*, pages 272–307, Belgium, Brussels. Association for Computational Linguistics.

Pi Chuan Chang, Michel Galley, and Christopher D Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *3rd Workshop on Statistical Machine Translation, WMT 2008 at the Annual Meeting of the Association for Computational Linguistics, ACL 2008*, pages 224–232.

Aditi Chaudhary, Karthik Raman, Krishna Srinivasan, and Jiecao Chen. 2020. Dict-mlm: Improved multilingual pre-training using bilingual dictionaries. *arXiv preprint arXiv:2010.12566*.

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

Paula Czarnowska, Sebastian Ruder, Edouard Grave, Ryan Cotterell, and Ann Copestake. 2019. Don't forget the long tail! a comprehensive analysis of morphological generalization in bilingual lexicon induction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 974–983, Hong Kong, China. Association for Computational Linguistics.

Paula Czarnowska, Sebastian Ruder, Edouard Grave, Ryan Cotterell, and Ann Copestake. 2020. Don't forget the long tail! A comprehensive analysis of morphological generalization in bilingual lexicon induction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 974–983.

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*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Philipp Dufter and Hinrich Schütze. 2020. Identifying elements essential for BERT's multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 4423–4437, Online. Association for Computational Linguistics.

Yaroslav Ganin, Hugo Larochelle, and Mario Marchand. 2016. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research*, 17:1–35.

Francisco Guzmán, Peng Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The Flores evaluation datasets for low-resource machine translation: Nepali-English and Sinhala-English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 6098–6111.

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. 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. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

K Karthikeyan, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual bert: An empirical study. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Yova Kementchedjhieva, Mareike Hartmann, and Anders Søgaard. 2019. Lost in evaluation: Misleading benchmarks for bilingual dictionary induction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3336–3341, Hong Kong, China. Association for Computational Linguistics.

Yova Kementchedjhieva, Mareike Hartmann, and Anders Søgaard. 2020. Lost in evaluation: Misleading benchmarks for bilingual dictionary induction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3336–3341.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Jitin Krishnan, Antonios Anastasopoulos, Hemant Purohit, and Huzefa Rangwala. 2021. Multilingual code-switching for zero-shot cross-lingual intent prediction and slot filling. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 211–223, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Advances in neural information processing systems*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018b. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018c. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Yoav Levine, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. 2021. PMI-Masking: Principled masking of correlated spans. In *9th International Conference on Learning Representations, ICLR 20201- Conference Track Proceedings*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Mauro Mezzini. 2018. Empirical study on label smoothing in neural networks. In *WSCG 2018 - Short papers proceedings*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? pages 4996–5001.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Shuo Ren, Yu Wu, Shujie Liu, Ming Zhou, and Shuai Ma. 2019a. Explicit cross-lingual pre-training for unsupervised machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 770–779, Hong Kong, China. Association for Computational Linguistics.

Shuo Ren, Zhirui Zhang, Shujie Liu, Ming Zhou, and Shuai Ma. 2019b. Unsupervised neural machine translation with smt as posterior regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 241–248.

Carol Myers Scotton and William Ury. 1977. Bilingual strategies: The social functions of code-switching.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Haipeng Sun, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2019. Unsupervised bilingual word embedding agreement for unsupervised neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1235–1245, Florence, Italy. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 4, pages 3104–3112.

Ishan Tarunesh, Syamantak Kumar, and Preethi Jyothi. 2021. From machine translation to code-switching: Generating high-quality code-switched text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3154–3169, Online. Association for Computational Linguistics.

Clara Vania and Adam Lopez. 2017. From characters to words to in between: Do we capture morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 2016–2027.

Ashish Vaswani, Google Brain, 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.

Ivan Vulić, Goran Glavaš, Roi Reichart, and Anna Korhonen. 2020. Do we really need fully unsupervised cross-lingual embeddings? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4407–4418.

Yiren Wang, Yingce Xia, Fei Tian, Fei Gao, Tao Qin, CengZiang Zhai, and Tie-Yan Liu. 2019. Neural Machine Translation with Soft Prototype. In *Advances in Neural Information Processing Systems*.

Guillaume Wenzek, Marie Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012.

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.

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

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations*.

A Limitations

In this work, we present a general masking scheme for multilingual MLM pre-training on multilingual corpora. Experiments show that our method can work for similar languages (including low-resource ones and high-resource ones) and dissimilar languages. However, we only experiment with dissimilar language Ne. More experiments are required for dissimilar and distant languages.

When computing $[\mathcal{C}]_x$ for more than 3 languages, to avoid cross-lingual bias, we adapt our method to a pivoting-based framework, using En as a pivot or anchor point. Although we show this framework can work for cross-lingual classification tasks, this could be a potential problem for further adaptation to other multilingual tasks, which requires further experiments. Intuitively, we can compute $[\mathcal{C}]_x$ in random languages instead of only in En with a balanced sample strategy.

Our method provides a general framework to leverage cross-lingual prototypes for multilingual MLM pre-training, but the scope of study is limited. We believe there are some other solutions. For instance, we can leverage linguistic varieties for masking, but the question is how to obtain linguistic varieties without using parallel corpora. Perhaps, we can consider word frequencies because Zipf's law indicates that words appear with different frequencies, and one may suggest similar meaning words appear with relatively similar frequencies in a pair of languages. Most importantly, solutions should further consider morphological variations, since in this paper we prove morphological variations are significantly beneficial.

B Robustness and Model Variation

We have some default configurations for our method, as presented in row 2 of Table 7. In this experiment, we observe the impact of K (the number of cross-lingual candidates), the warm-up initialization, the tokenization method, and the alternation $t\%$. We consider the weighted average of cross-lingual candidates for $[\mathcal{C}]_x$, and additionally we consider the mean average style in this experiment. For initialization, we further study alternatives. The result is presented in Table 7.

Row $3 \sim 6$ Models with a common choice of K ($1 \sim 5$) outperform the baseline model. However, $K = 1$ (a single candidate) yields median improvements. Meanwhile, when $K = 1$, our method is similar to (Dufter and Schütze, 2020; Chaudhary et al., 2020) who employ static and word translation tables (e.g., UBWE and dictionary) for obtaining a single candidate, and they have similar results. Intuitively, the model cannot capture morphological variations and synonyms in the other language when only using one candidate, as discussed in the experiment of UNMT, but they are important in translation. It proves the significance of using multiple candidates.

Row $7 \sim 9$ Warm-up is necessary to facilitate $[\mathcal{C}]_x$. Although a small amount of warm-up steps is enough, it is a disadvantage of $[\mathcal{C}]_x$ somewhat. We believe there is a significant potential for development of other new alternatives. We present two options in row 17 and row 18 (see the following text).

Row 10 Also, we can see there is no significant difference between the word-level tokenization and the BPE tokenization. Although the BPE tokenization gains slightly better performance, the improvement we believe is from the effectiveness of the BPE tokenization itself not the discrepancy of $[\mathcal{C}]_x$.

Row 11 As aforementioned, CLPM requires additional time to compute $[\mathcal{C}]_x$. To be fair, we reduce the training steps, so that the training time is almost similar to the baseline model (row 1). In a similar training time, CLPM outperforms the baseline model but requires smaller training steps, which indicates that the explicit and principled cross-lingual forward pass is more efficient (per step) than implicit isomorphic space formation for cross-linguality.

Row	Model	t	Tokenization	Warm-up	Steps	K	$[\mathcal{C}]_x$ type	$De \leftrightarrow En$	
1	$[\mathcal{M}]$ (baseline)	-	BPE	-	400K	-	-	34.3	26.4
2	$[\mathcal{C}]_x$ (our baseline, default)	40%	BPE	50K	400K	3	weighted	35.9	28.1
3	$[\mathcal{C}]_x$	+	+	+	+	1	+	34.9	27.3
4	$[\mathcal{C}]_x$	+	+	+	+	2	+	35.8	27.9
5	$[\mathcal{C}]_x$	+	+	+	+	4	+	36.0	28.0
6	$[\mathcal{C}]_x$	+	+	+	+	5	+	35.9	28.1
7	$[\mathcal{C}]_x$	+	+	20k	+	+	+	35.1	27.1
8	$[\mathcal{C}]_x$	+	+	100K	+	+	+	35.8	28.0
9	$[\mathcal{C}]_x$	+	+	200K	+	+	+	35.3	27.5
10	$[\mathcal{C}]_x$	+	Word-level	+	+	+	+	35.8	28.0
11	$[\mathcal{C}]_x$	+	+	+	350K (similar training time)	+	+	35.1	27.2
12	$[\mathcal{C}]_x$	10%	+	+	+	+	+	35.6	28.0
13	$[\mathcal{C}]_x$	70%	+	+	+	+	+	34.8	27.2
14	$[\mathcal{C}]_x$	from 0 to 70%	+	+	+	+	+	35.4	27.7
15	$[\mathcal{C}]_x$	only $[\mathcal{C}]_x$ (no $[\mathcal{M}]$)	+	+	+	+	+	34.1	26.5
16	$[\mathcal{C}]_x$	+	+	+	+	+	mean	35.3	27.8
17	$[\mathcal{C}]_x$	+	+	UBWE	+	+	+	36.5	28.8
18	$[\mathcal{C}]_x$	+	+	1k seed dictionary	+	+	+	36.9	29.1

Table 7: Model Variation. All the models are based on XLM instance. Row 2 shows the default configurations we use in UNMT. $+$ denotes the default configuration. $-$ denotes an inapplicable term. For *mean*, we average the embeddings of candidates instead of weighted averaging. *UBWE* denotes we pre-train the bilingual embeddings unsupervisedly and then pre-train the entire model with our method. In *1k seed dictionary* test, the model employs a candidate from a seed dictionary.

Row $12 \sim 14$ We alternate between $[\mathcal{C}]_x$ and $[\mathcal{M}]$ because we consider learning the morphology and internal structure of languages from $[\mathcal{M}]$ like BERT. Note that the baseline model (row 1) is equivalent to $t = 0$ (only use $[\mathcal{M}]$). We observe that $t = \{10\%, 40\%, 70\%\}$ significantly outperform $t = \{0\}$. This confirms our intuition for the need of dual objectives that the UNMT model greedily obtains the explicit cross-linguality from $[\mathcal{C}]_x$ and the bidirectional/language knowledge from $[\mathcal{M}]$. We also consider the scenario that we increase t from 0 to 70% linearly, achieving competitive performance with $t = \{10\%, 40\%, 70\%\}$.

Row 15 We have a question: does $[\mathcal{C}_x]$ hurt learning language knowledge? Although $[\mathcal{M}]$ itself cannot provide any supervision, the model can learn strong language knowledge by understanding bidirectional information. By considering this fact, using $[\mathcal{C}_x]$ instead of $[\mathcal{M}]$ potentially fails in learning language knowledge, even though the cross-lingual forward pass:$\{[\mathcal{C}]_{x_i}, x_{j\backslash i}\} \rightarrow x_i$ involves the neighboring token x_j. To investigate, we experiment with only using $[\mathcal{C}]_x$. Compared to only using $[\mathcal{M}]$, only using $[\mathcal{C}_x]$ does degrade the performance of UNMT. We suspect that the translation is not fluent due to the lack of learning bidirectional knowledge with the help of $[\mathcal{M}]$. However, applying the alternation strategy can mitigate the pain, and row $12 \sim 15$ show the alternation strategy can consistently improve performance on translation. Our intuition is that both cross-linguality and language knowledge are essential for translation, similar to the observation in (Zhang et al., 2021; Ai and Fang, 2022a).

Row 16 As we consider the weighted average of the candidate set, we are aware that the mean-average style is also an alternative. The test shows that the weighted-average style outperforms the mean-average style. We conjecture that the weighted-average style can compute more reliable cross-lingual prototypes because for some unambiguous tokens, the mean-average style may pay much more attention to low-weight candidates. For instance, if the weights in Step 4 are $\{0.9, 0.15, 0.05\}$, computing $[\mathcal{C}]_x$ is forced to pay much more attention to "0.05" by the mean-average style, which is unnecessary. On the other hand, the margin is not large. We suspect that the candidate set covers morphological variations and synonyms. Therefore, they have similar weights after the $softmax$ normalization, which results in a similar output from the weighted average and the mean average.

Row 17 Inspired by UBWE (unsupervised bilingual word embedding) (Lample et al., 2018a; Artetxe et al., 2018a, 2016, 2017), we are aware that we can pre-train cross-lingual embeddings for the multilingual model before multilingual MLM pre-training instead of the random initialization with the warm-up. To this end, we use the MUSE◇ (Lample et al., 2018a)'s UBWE method to initialize the bilingual embedding space. In the first 50k pre-training steps (equal to default warm-up steps), since the model parameters are still randomly initialized, we do not follow Step 1, 2, and 3 in on-the-fly $[\mathcal{C}]_x$ and directly find relevant candidates based on the dot products $E_{y^i}^T E_x$, i.e., only need Step 4. Intuitively, $E_{y^i}^T E_x$ is reliable to rank the candidates and computes the weights for $[\mathcal{C}]_x$, especially at the early iterations, because UBWE provides cross-lingual entries. After 50k pre-training steps, we normally run on-the-fly $[\mathcal{C}]_x$. We observe that adapting UBWE consistently improves the performance by 2% on the similar language and $0.5 \sim 1$ BLEU on the dissimilar language because UBWE provides additional cross-lingual supervision. All the results are presented in Table 8.

Row 18 (Vulić et al., 2020) suggest seed dictionaries for unsupervised tasks in practice. Following this idea, we download a 1k seed dictionary from Panlex◇. In the first 50k pre-training steps, we simply replace the selected token with its translation in the seed dictionary. For the out-of-the-dictionary but selected token, we replace it with normal $[\mathcal{M}]$. After 50k pre-training steps, if the selected token is in the dictionary, the translation is added to $[\mathcal{C}]_x$ as a candidate in Step 4 when running on-the-fly $[\mathcal{C}]_x$. We find that compared to the UBWE scenario, this adaptation achieves similar results on the rich-resource language $De \leftrightarrow En$ (+ 1.5%) but stronger results on the dissimilar language $Ne \leftrightarrow En$ (+ 8%). All the results are presented in Table 8.

C Additional Experiment

C.1 Alternatives

Given an input word and the current model Net, we compute $[\mathcal{C}]_x$ by *1)* computing the contextualized representation by setting the model to the inference mode with the target language embedding $\tilde{Net}(E_x + E_{L_y})$, *2)* computing $softmax$ over the contextualized representations in the output (embedding) layer, *3)* selecting the Top-k embeddings with the highest $softmax$ score, and (*4)* computing a weighted average over the selected embeddings. Essentially, we use the target language embedding for biasing the representations towards the target language. The question remains as to how well it works. Meanwhile, two alternatives are interesting: *1)* $\tilde{Net}(E_x + E_{L_x})$, which uses the source language embedding to compute representations; *2)* Top-k Nearest Embedding, which computes candidates by using Top-k Nearest Embeddings in the embedding space without using the inference mode. In Table 9, we provide an empirical study for $\tilde{Net}(E_x + E_{L_y})$, $\tilde{Net}(E_x + E_{L_x})$, and Top-k Nearest Embedding. Our observations are:

- Top-k Nearest Embedding seems to find over-shared tokens. For instance, in #3, it finds $[\mathcal{C}]_{x_8}$ = <to, for, by> for <to>, where <to, for, by> are shared by all the languages. With cross-lingual transfer in mind, we believe that a candidate set only covering over-shared tokens is not a good one, e.g., <to, for, by> is not a good candidate set crossing En to De. Meanwhile, Top-k Nearest Embedding is not good at finding strong candidates.
- $\tilde{Net}(E_x + E_{L_x})$ is better than Top-k Nearest Embedding because $\tilde{Net}(E_x + E_{L_x})$ do not obtain too much over-shared tokens.
- Compared to $\tilde{Net}(E_x + E_{L_x})$, $\tilde{Net}(E_x + E_{L_y})$ (our suggestion) will change the score of the full-sized set $Q = (h_{x_i\&L_y}^T O_{y_0}, ..., h_{x_i\&L_y}^T O_{y_v})$ (Step 2). These scores are very dense, so that small changes cause significant differences. Then, $\tilde{Net}(E_x + E_{L_y})$ is better to rank candidates than $\tilde{Net}(E_x + E_{L_x})$.

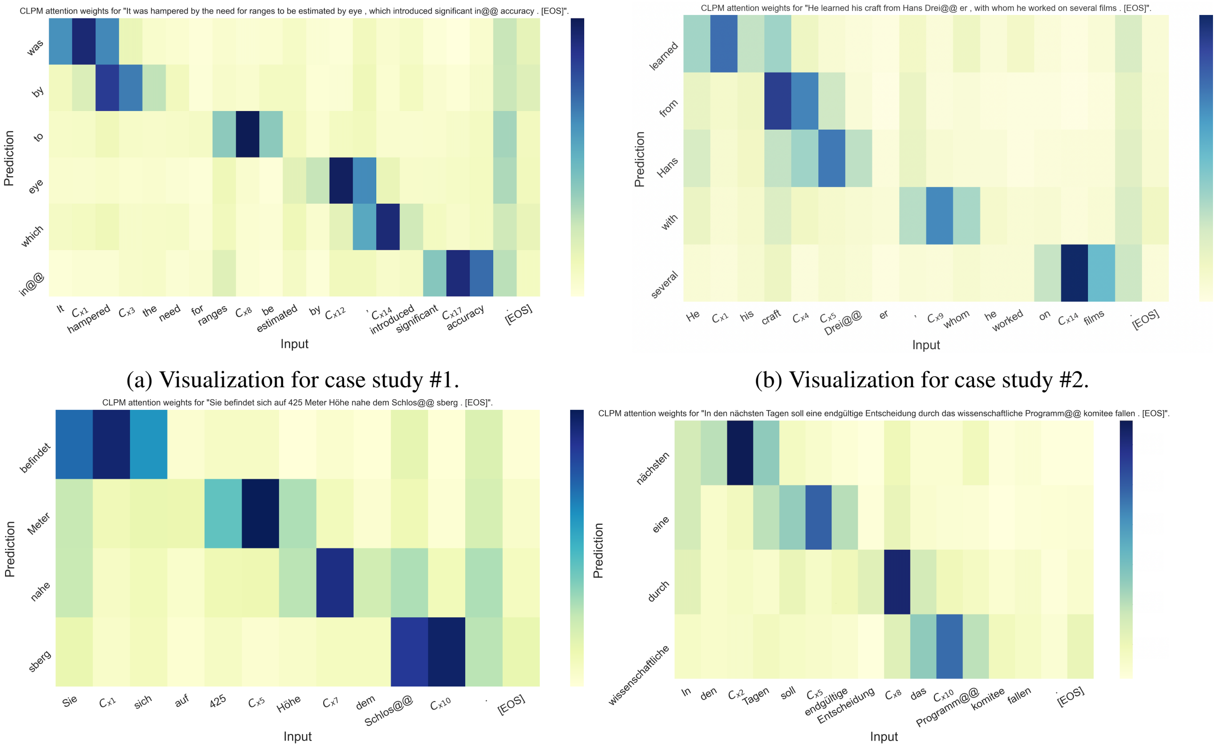
In conclusion, $\tilde{Net}(E_x + E_{L_y})$ shows the advance in: *1)* it does not consider too many over-shared tokens; *2)* $\tilde{Net}(E_x + E_{L_y})$ with the target language embedding is better to rank candidates than $\tilde{Net}(E_x + E_{L_x})$; *3)* $\tilde{Net}(E_x + E_{L_y})$ can cover multiple morphological or relevant candidates (e.g., $[\mathcal{C}]_{x_5}$ = **<metres**, **metre**, **yards>** in #4) for generalizing information by weighted average. In this way, $\tilde{Net}(E_x + E_{L_y})$ finds better cross-lingual prototypes, which results in better generalized information by weighted average.

C.2 Case Study

To further probe the results, we use pre-trained weights from UNMT and compute $[\mathcal{C}]_x$ for the selected tokens of sentences, obtaining 3 candidates for each token. *We observe attention weights on* $[\mathcal{C}]_x$. Our case study of Table 2 shows that for predicting replaced tokens, the model outputs prominent attention weights on corresponding $[\mathcal{C}]_x$, so

Language pair	$De \leftrightarrow En$		$Ro \leftrightarrow En$		$Ne \leftrightarrow En$	
XLM(Lampple et al., 2018c)	34.3	26.4	31.8	33.3	0.5	0.1
+ UBWE $\star$	34.0	27.0	33.3	34.1	4.9	1.3
+ $[\mathcal{C}]_x$	35.9	28.1	34.4	35.3	6.6	2.8
+ $[\mathcal{C}]_x$ + UBWE (for wam-up with Step 1,2 and 3)	36.5	28.8	35.1	36.0	8.3	3.2
+ $[\mathcal{C}]_x$ + 1K seed dictionary (Vulić et al., 2020) (for warm-up with Step 1,2 and 3)	36.5	28.9	35.7	36.5	9.1	4.0

Table 8: Incorporation of UBWE. $\star$ models are reimplemented with our configurations. We find that XLM can employ UBWE to improve the performance of low-resource languages and dissimilar languages. However, it has a limited impact on rich-resource languages. CLPM obtains more gains from UBWE.



(a) Visualization for case study #1.

(b) Visualization for case study #2.

(c) Visualization for case study #3.

(d) Visualization for case study #4.

Figure 2: Case study of CLPM. These figures show that the model understands $[\mathcal{C}]_x$ (also see Table 9) in the context.

	$\tilde{N}et(E_x + E_{L_y})\,([\mathcal{C}]_x)$	$\tilde{N}et(E_x + E_{L_x})$	Top-k Nearest Embedding
#1	The investment fund that owned the building had to make a choice . [EOS]		
Reference	Der Investmentfonds, dem das Gebäude gehörte , musste sich entscheiden . [EOS]		
Masked	The $[\mathcal{C}]_1$ $[\mathcal{C}]_2$ that $[\mathcal{C}]_4$ $[\mathcal{C}]_5$ $[\mathcal{C}]_6$ $[\mathcal{C}]_7$ to $[\mathcal{C}]_9$ a choice . [EOS]		
investment = $[\mathcal{C}]_2$	Aufsichts@@, Förder@@, **Einnahmen**	Aufsichts@@, Förder@@, **Einnahmen**	Milliarden, Denkmalschutz, Kritiken
fund = $[\mathcal{C}]_{x_2}$	wurf, **funde**, **Förderung**	**funde**, **Förderung**, wurf	Nachlass, **funde**, firma
owned = $[\mathcal{C}]_{x_4}$	**gehörte**, **kaufte**, **Eigentum**	**Eigentum**, **gehörte**, **kaufte**	entstammte, geprägten, erbaute
building = $[\mathcal{C}]_{x_6}$	**Gebäude**, **gebäude**, **Anlage**	**Gebäude**, **gebäude**, **Gebäudes**	**gebäude**, **gebäudes**, **Gebäude**
had = $[\mathcal{C}]_{x_7}$	kam, **hatte**, **war**	kam, **hatte**, gab	entstammte, Seinen, Zur
make = $[\mathcal{C}]_{x_9}$	Stand@@, **machten**, **macht**	**machten**, Stand@@, **macht**	Ist, bestritt, bestes
#2	He learned his craft from Hans Drei@@ er , with whom he worked on several films . [EOS]		
Reference	Sein Handwerk lernte er bei Hans Dreier , mit dem er an mehreren Filmen arbeitete . [EOS]		
Masked	He $[\mathcal{C}]_{x_1}$ his craft $[\mathcal{C}]_{x_4}$ Hans $[\mathcal{C}]_{x_6}$ $[\mathcal{C}]_{x_7}$, $[\mathcal{C}]_{x_9}$ whom he $[\mathcal{C}]_{x_{12}}$ on several films .		
learned = $[\mathcal{C}]_{x_1}$	**stammte**, **stammten**, **stammt**	**stammte**, **stammten**, **stammt**	**entstammte**, **erlernte**, **studierte**
from = $[\mathcal{C}]_{x_4}$	**von**, **Von**, **vom**	**von**, **Von**, **vom**	**Von**, **Vom**, ;
Drei@@ = $[\mathcal{C}]_{x_6}$	**Drei**@@, Zwei@@, Vier@@	**Drei**@@, Zwei@@, Mehr@@	**Drei**@@, drei@@, Fünf@@
er = $[\mathcal{C}]_{x_7}$	er, es, der	er, es, der	er, sie, es
with = $[\mathcal{C}]_{x_9}$	**mit**, **in**, **Mit**	**mit**, **in**, **Mit**	**Mit**, Beim, wobei
worked=$[\mathcal{C}]_{x_{12}}$	**arbeitete**, **wirkte**, **arbeiteten**	**wirkte**, **arbeitete**, **gearbeitet**	promovierte, kandidierte, **studierte**
#3	It was hampered by the need for ranges to be estimated by eye , which introduced significant in@@ accuracy . [EOS]		
Reference	Erschwert wurde dies durch die Notwendigkeit , Entfernungen mit dem Auge abzuschätzen, was zu erheblichen Ungenauigkeiten führte . [EOS]		
Masked	It $[\mathcal{C}]_{x_1}$ hampered by $[\mathcal{C}]_{x_4}$ need $[\mathcal{C}]_{x_6}$ ranges $[\mathcal{C}]_{x_8}$ be estimated by $[\mathcal{C}]_{x_{12}}$, $[\mathcal{C}]_{x_{14}}$ introduced significant $[\mathcal{C}]_{x_{17}}$ accuracy . [EOS]		
was = $[\mathcal{C}]_{x_1}$	**war**, **wurde**, **als**	**war**, ,, **wurde**	(, welches, **Was**
hampered = $[\mathcal{C}]_{x_2}$	hauptsächlich, Gesundheit@@, durchgeführt	hauptsächlich, Gesundheit@@, durchgeführt	angesichts, hinsichtlich, entstammte
the = $[\mathcal{C}]_{x_4}$	**den**, **die**, [EOS]	**die**, **den**, [EOS]	**die**, :, **den**
for = $[\mathcal{C}]_{x_6}$	**für**, **dafür**, in	**für**, **dafür**, in	**für**, **Für**, in
to = $[\mathcal{C}]_{x_8}$	to, **dem**, sich	to, **dem**, erweitert	to, for, by(×)
which = $[\mathcal{C}]_{x_{14}}$	**welches**, **welcher**, **welche**	**welches**, **welcher**, **welche**	**welches**, **welchen**, **welcher**
in@@ = $[\mathcal{C}]_{x_{17}}$	inen, höher, .	inen, unge@@, höher	inen, unter@@, auf@@
#4	Die Gleis@@ anlage war so ausgestattet , dass dort elektrisch betriebene Wagen eingesetzt werden konnten . [EOS]		
Reference	The track system was equipped in such a way that electrically operated cars could be used there . [EOS]		
Masked	$[\mathcal{C}]_{x_0}$ Gleis@@ $[\mathcal{C}]_{x_2}$ $[\mathcal{C}]_{x_3}$ so $[\mathcal{C}]_{x_5}$ $[\mathcal{C}]_{x_6}$ $[\mathcal{C}]_{x_7}$ dort elektrisch $[\mathcal{C}]_{x_{10}}$ $[\mathcal{C}]_{x_{11}}$ eingesetzt werden konnten . [EOS]		
Die = $[\mathcal{C}]_{x_0}$	**The**, In, [EOS]	**The**, In, Decline	His, Her, **The**
anlage = $[\mathcal{C}]_{x_2}$	**facility**, **facilities**, **Complex**	**facility**, **facilities**, **Complex**	anime, HMS, {
war = $[\mathcal{C}]_{x_3}$	**was**, crew. remained	**was**, crew. remained	was, :, ;
ausgestattet = $[\mathcal{C}]_{x_5}$	**equipped**, **fitted**, yan	**equipped**, **fitted**, engines	whose, **equipped**, dae
, = $[\mathcal{C}]_{x_6}$	**,**, [EOS], ;	**,**, ;, [EOS]	**,**, ;, [EOS]
dass = $[\mathcal{C}]_{x_7}$	**why**, **how**, **whether**	**why**, **whether**, resources	**whether**, **why**, unlike
betriebene = $[\mathcal{C}]_{x_{10}}$	**operated**, like, isha	like, **operated**, isha	Romanized, whose, starring
Wagen = $[\mathcal{C}]_{x_{11}}$	**drove**, **cars**, GP	**drove**, **cars**, GP	Stakes, fled, dancer
#5	In den nächsten Tagen soll eine endgültige Entscheidung durch das wissenschaftliche Programm@@ komitee fallen . [EOS]		
Reference	A final decision is to be made by the scientific program committee in the next few days . [EOS]		
Masked	In den $[\mathcal{C}]_{x_2}$ Tagen soll $[\mathcal{C}]_{x_5}$ endgültige $[\mathcal{C}]_{x_7}$ durch das $[\mathcal{C}]_{x_{10}}$ Programm@@ $[\mathcal{C}]_{x_{12}}$ fallen . [EOS]		
nächsten = $[\mathcal{C}]_{x_2}$	**next**, past, host	**next**, past, **Next**	**next**, **nearest**, longest
eine = $[\mathcal{C}]_{x_5}$	**a**, **someone**, formed	**a**, **someone**, formed	**someone**, **a**, Her
Entscheidung = $[\mathcal{C}]_{x_7}$	**vision**, left, Note	**vision**, left, Note	Shortly, p.m., {
wissenschaftliche = $[\mathcal{C}]_{x_{11}}$	**scientific**, **research**, **journal**	**scientific**, **research**, **journal**	peer, **doctoral**, remembered
komitee = $[\mathcal{C}]_{x_{12}}$	**committee**, **Congress**, **body**	**committee**, **Congress**, **body**	{, **Laboratory**, certified
#6	Sie befindet sich auf 425 Meter Höhe nahe dem Schlos@@ sberg . [EOS]		
Reference	It is located at an altitude of 425 meters near the Schlossberg. [EOS]		
Masked	$[\mathcal{C}]_{x_0}$ $[\mathcal{C}]_{x_1}$ sich auf 425 $[\mathcal{C}]_{x_5}$ $[\mathcal{C}]_{x_6}$ $[\mathcal{C}]_{x_7}$ dem Schlos@@ $[\mathcal{C}]_{x_{10}}$. [EOS]		
auf = $[\mathcal{C}]_{x_3}$	**on**, **in**, **below**	**in**, **on**, an	an, **in**, **On**
Meter = $[\mathcal{C}]_{x_5}$	**metres**, **metre**, **yards**	**metres**, **metre**, **yards**	**metres**, meters, **metre**
Höhe = $[\mathcal{C}]_{x_6}$	**elevation**, **depth**, sales	**elevation**, **depth**, sales	**altitude**, **elevation**, excess
nahe = $[\mathcal{C}]_{x_7}$	**near**, **inside**, security	**near**, **inside**, security	**near**, **Near**, nicknamed
sberg = $[\mathcal{C}]_{x_{10}}$	say, sort, sing	say, sort, sing	p.m., re, Bros.

Table 9: Examples of $[\mathcal{C}]_x$ and alternatives. *Although we compute generalized information from the candidate set by weighted average, candidates are significant for generalizing information intuitively.* The goal of this table is to show some examples of the candidates for $[\mathcal{C}]_x$. References are obtained from Google Translation. We use the pre-trained weights from UNMT experiments on $\{En, De\}$. To obtain more examples we randomly compute $[\mathcal{C}]_x$ for 40% of tokens. @@ is the continuing subword prefix. × denotes the method that only finds over-shared tokens because of scripts. For instance, *1)* En <the> appears in De, but preferably it should be paired with De words such as <das, die, der, den> instead of itself; *2)* De {war} should be aligned to En <was> instead of En <war>. With cross-lingual transfer in mind, we believe that a candidate set only covering over-shared tokens is not a good one, e.g., $[\mathcal{C}]_{x_8}$=<to, for, by> is not a good candidate set crossing En to De in #3. **bold** denotes a strong candidate that is a parallel, analogical, or relevant token/word (or its variation) in other languages. The model can cover multiple morphological or relevant candidates. For instance , in #3, our method finds $[\mathcal{C}]_{x_{14}}$ = <**welches**, **welcher**, **welche** > for generalizing information by weighted average.

that it relies on $[\mathcal{C}]_x$ to predict the replaced tokens, whereas a model can only rely on neighboring tokens to predict the replaced tokens if only using $[\mathcal{M}]$. Since $[\mathcal{C}]_x$ is the cross-lingual prototype, the model can learn cross-linguality from the $[\mathcal{C}]_x$. We can confirm the effectiveness of $[\mathcal{C}]_x$.

For example, to predict <Meter> (Figure 2c), our method finds possible translation for $[\mathcal{C}]_{x_5}$ = **<metres**, **metre**, **yards>**, and the attention weight on its $[\mathcal{C}]_{x_5}$ dominates others. We conjecture that our method shows significant effectiveness on nouns, entities, terminology words, etc., because parallel, analogical, or relevant words of these words in other languages might be easily inferred. Meanwhile, it shows the importance of using multiple candidates because the model might understand linguistic varieties. Besides, in this way, the model can yield generalized representations from $[\mathcal{C}]_x$ in the other language (Step 4), which might be useful for translation and cross-lingual transfer. Furthermore, as discussed in §2.6, the model can handle sub-word tokens because for predicting <in@@> (Figure 2a), the model pays similar attention to its $[\mathcal{C}]_{x_{17}}$ and its neighboring token <accuracy>, where <in@@> and <accuracy> are split from <inaccuracy>. It indicates that the model can consider the sub-token's cross-lingual prototype in the context. We attribute this phenomenon to both the alternation between $[\mathcal{C}]_x$ and $[\mathcal{M}]$ and involving neighboring x_j in $\{[\mathcal{C}]_{x_i}, x_{j\backslash i}\} \to x_i$ that the model captures token dependencies from the cross-lingual prototype in the other language with the same semantic. Surprisedly, to predict <which> (Figure 2a) with its $[\mathcal{C}]_{x_{14}}$ = **<welches**, **welcher**, **welche** >, the model seems to understand some syntax structures because the model pays more attention to <,> than <introduced>, where $[\mathcal{C}]_{x_{14}}$ and <,> might jointly represent the syntax structure <, which>.

Recall the discriminator 1, which confirms that cross-lingual prototypes belong to one language but do not exist in the embedding space, i.e., not used in discriminator training. The model cannot only rely on cross-lingual prototypes to recover masked tokens because cross-lingual prototypes are not translations. The model has to consider both cross-lingual prototypes and the context, understanding the generalized information of cross-lingual prototypes in the context. The case study confirms this as attention weights observed from neighboring tokens around $[\mathcal{C}]_x$.

D Experiment Setting

D.1 Pre-training

Our code is implemented on Tensorflow 2.2 (Abadi et al., 2016). We use Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$,$\beta_2 = 0.999$, $\epsilon = 1e-8$, and $lr = 1e-4$. Dropout regularization is set to $rate = 0.1$. The mini-batch size is set to 8192 tokens for all experiments. We sample sentences from different languages with the balance strategy (Lample and Conneau, 2019).

D.2 MLM Instance

We adapt our method to three MLM instances: XLM (Lample and Conneau, 2019), MASS (Song et al., 2019), and mBART (Liu et al., 2020), which can be used to pre-train the multilingual model. We follow the instructions of BERT (Devlin et al., 2019) and these three MLM instances that each selected token is replaced with the probabilities $([SAME], [RAN], [\mathcal{M}]) = (10\%, 10\%, 80\%)$.

XLM XLM is similar to BERT (Devlin et al., 2019) but uses text streams of an arbitrary number of sentences. Following the instruction, we randomly select 15% of the tokens from the input sentence for replacing.

MASS MASS is different from XLM and BERT but similar to SpanBERT (Joshi et al., 2020), using spans to replace consecutive tokens. Given an input sentence with length N, we randomly select consecutive tokens with length $N/2$ for replacing.

mBART mBART applies spans to replace consecutive tokens for a text instance of two concatenated random sentences and perturbs the order of the two concatenated sentences for prediction. We randomly select 35% of the tokens in each instance for replacing by sampling a span length according to a Poisson distribution $\lambda = 3.5$ and swap the two sentences within each instance.

Significantly, to minimize changes for evaluation, we only have two changes.

- We extend the masking strategy: $([SAME], [RAN], [\mathcal{M}])$ with $(10\%, 10\%, 80\%)$ to $([SAME], [RAN], [\mathcal{M}], [\mathcal{C}]_x)$ with $(10\%, 10\%, (80-t)\%, t\%)$.

- Secondly, as presented in Table 1, we only apply CLPM to the input of the source side or the encoder. Other components of the framework are identical to the reported MLM instances, and we do not change the shifted input of the decoder in seq2seq learning (Sutskever et al., 2014).

D.3 Setup

UNMT Setup We consider the same dataset used in previous works. Specifically, we first retrieve monolingual corpora $\{De, En\}$ from WMT 2018◇ (Bojar et al., 2018) including all available $NewsCrawl$ datasets from 2007 through 2017 and monolingual corpora Ro from WMT 2016◇ (Bojar et al., 2016) including $NewsCrawl$ 2016. We report $\{De, Ro\} \leftrightarrow En$ on *newstest2016*. Meanwhile, we share the FLoRes◇ (Guzmán et al., 2019) task to evaluate on a dissimilar language pair $Ne \leftrightarrow English$ (Nepali). We download the dataset and test set with provided script. Ne is tokenized by Indic-NLP Library◇. For others, we use the Moses tokenizer◇ developed by (Koehn et al., 2007). We use fastBPE◇ to learn shared BPE (Sennrich et al., 2016b), selecting the most frequent 60K tokens from concatenated corpora of paired languages with the same criteria in (Lample and Conneau, 2019). The model is pre-trained around 400K iterations on only monolingual corpora of paired languages. Then, we still train MLM but eventually train the translation task on synthetic parallel sentences by running on-the-fly back-translation (Sennrich et al., 2016a), which is the standard pipeline◇ of UNMT (Artetxe et al., 2018b; Song et al., 2019). After around 400K iterations, according baseline models' BLEU scripts, we report BLEU computed by *multi-BLEU.perl*◇ or *sacreBleu*◇ (Post, 2018) with default rules. In the training phase, we use Adam optimizer (Kingma and Ba, 2015) with parameters $\beta_1 = 0.9$,$\beta_2 = 0.997$ and $\epsilon = 10^{-9}$, and a dynamic learning rate with $warm\_up = 8000$ (Vaswani et al., 2017) ($learning\_rate \in (0, 7e^{-4}]$) is employed. We set dropout regularization with a drop rate $rate = 0.1$ and label smoothing with $gamma = 0.1$ (Mezzini, 2018).

Cross-ling Classification Setup Beyond UNMT tasks or bilingual tasks, our method can be applied to multilingual tasks. Then, we attempt the cross-lingual classification task on XNLI (Conneau et al., 2018) to test general cross-linguality $[\mathcal{C}]_x$ improves. For this test, we follow the standard and basic experiment (Lample and Conneau, 2019) to train a 12-layer Transformer encoder with 80k BPE on Wikipedia dumps◇ of 15 XNLI languages. To tokenize $\{Zh, Th\}$, we use Stanford Word Segmenter◇ and PyThaiNLP◇ respectively. For the others, we use the Moses tokenizer◇ with default rules. Similarly, we use fastBPE◇ and the balanced strategy (Lample and Conneau, 2019) to learn BPE. While there are two settings in this task, we only report the results of the zero-shot classification. To pre-train the encoder on En corpora, considering the zero-shot classification based on finetuing En NLI dataset, we randomly compute $[\mathcal{C}]_x$ from other languages with equal probability to avoid the cross-lingual bias. For pre-training on corpora of other languages, we only compute $[\mathcal{C}]_x$ in the $English$ domain. Note that, although we have different strategies of $[\mathcal{C}]_x$ for different languages, we still concatenated all the corpora of the languages for joint pre-training. After pre-training on the corpora, we deploy a randomly initialized linear classifier and finetune the encoder and the linear classifier on the En NLI dataset with minibatch size 16. We use Adam optimizer (Kingma and Ba, 2015) with $lr = 5e - 4$ and linear decay of lr. After fintuning, we make zero-shot classifications for other languages.

E Performance

E.1 UNMT

We compare our reimplementation with reported results in Table 10.

E.2 Cross-lingual Classification

We show the results of XNLI for each language in Table 11.

F Source

We list all the links of dataset, tools, and other sources in Table 12.

Language pair	$De \leftrightarrow En$		$Ro \leftrightarrow En$		$Ne \leftrightarrow En$	
multi-BLEU.perl◇ with default rules						
XLM(Lample et al., 2018c) *reported*	34.3	26.4	31.8	33.3	0.5	0.1
XLM(Lample et al., 2018c) ⋆	33.9	26.3			0.6	0.2
+ $[\mathcal{C}]_x$	35.9	28.1	34.4	35.3	6.6	2.8
MASS(Song et al., 2019) *reported*	35.2	28.3	33.1	35.2		
MASS(Song et al., 2019)⋆	35.0	28.0			0.9	0.3
+ $[\mathcal{C}]_x$	36.7	29.2	34.7	36.9	7.1	3.4
sacreBleu◇ with standard settings: nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0						
mBART(Liu et al., 2020) *reported* +CC25	34.0	29.8	30.5	35.0	10.0	4.4
mBART(Liu et al., 2020)⋆	33.7	29.4			2.0	1.1
+ $[\mathcal{C}]_x$	35.4	30.1	32.5	36.7	7.0	3.2

Table 10: Performance of UNMT. Baseline models (⋆) are reimplemented with our configurations.

Model	en	fr	es	de	el	bg	ru	tr	ar	vi	th	zh	hi	sw	ur	Avg
baseline(Conneau et al., 2018)	73.7	67.7	68.7	67.7	68.9	67.9	65.4	64.2	64.8	66.4	64.1	65.8	64.1	55.7	58.4	65.6
mBERT (Wu and Dredze, 2019)	82.1	73.8	74.3	71.1	66.4	68.9	69	61.6	64.9	69.5	55.8	69.3	60.0	50.4	58.0	66.3
XLM (Lample and Conneau, 2019)	83.2	76.5	76.3	74.2	73.1	74.0	73.1	67.8	68.5	71.2	69.2	71.9	65.7	64.6	63.4	71.5
+ word translation tables(Chaudhary et al., 2020)																72.7
+ $[\mathcal{C}]_x$	84.8	78.1	78.0	76.7	75.8	76.6	74.7	71.6	71.9	74.2	71.8	74.9	67.4	67.2	66.5	74.0
+ MT (Lample and Conneau, 2019)	85.0	78.7	78.9	77.8	76.6	77.4	75.3	72.5	73.1	76.1	73.2	76.5	69.6	68.4	67.3	75.1

Table 11: Performance of cross-lingual classification on XNLI. MT stands for additional parallel corpora.

Item	Links
WMT 2016	http://www.statmt.org/wmt16/translation-task.html
WMT 2018	http://www.statmt.org/wmt18/translation-task.html
FLoRes	https://github.com/facebookresearch/flores
Indic-NLP Library	https://github.com/anoopkunchukuttan/indic_nlp_library
XLM	https://github.com/facebookresearch/XLM
multi-BLEU.perl	https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-BLEU.perl
Moses tokenizer	https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl
Kytea	http://www.phontron.com/kytea/
XTREME	https://github.com/google-research/xtreme
fastBPE	https://github.com/glample/fastBPE
MUSE	https://github.com/facebookresearch/MUSE
Cambridge Dictionary	https://dictionary.cambridge.org/
WikiExtractor	https://github.com/attardi/wikiextractor
PyThaiNLP	https://github.com/PyThaiNLP/pythainlp
Stanford Word Segmenter (Chang et al., 2008)	https://nlp.stanford.edu/software/segmenter.html
Tensor2Tensor	https://github.com/tensorflow
HuggingFace	https://huggingface.co

Table 12: Links of source.