MTLS: Making Texts into Linguistic Symbols

Wenlong Fei, Xiaohua Wang, Min Hu, Qingyu Zhang, Hongbo Li
School of Computer Science and Information Engineering, HeFei University of Technology
feiwenlong@mail.hfut.edu.cn, xh_wang@hfut.edu.cn

Abstract

In linguistics, all languages can be considered as symbolic systems, with each language relying on symbolic processes to associate specific symbols with meanings. In the same language, there is a fixed correspondence between linguistic symbol and meaning. In different languages, universal meanings follow varying rules of symbolization in one-to-one correspondence with symbols. Most work overlooks the properties of languages as symbol systems. In this paper, we shift the focus to the symbolic properties and introduce MTLS: a pre-training method to improve the multilingual capability of models by *Making Texts into Linguistic Symbols*. Initially, we replace the vocabulary in pre-trained language models by mapping relations between linguistic symbols and semantics. Subsequently, universal semantics within the symbolic system serve as bridges, linking symbols from different languages to the embedding space of the model, thereby enabling the model to process linguistic symbols. To evaluate the effectiveness of MTLS, we conducted experiments on multilingual tasks using BERT and RoBERTa, respectively, as the backbone. The results indicate that despite having just over 12,000 pieces of English data in pre-training, the improvement that MTLS brings to multilingual capabilities is remarkably significant.

1 Introduction

All languages can be considered as symbolic systems (De Saussure, 1989). This is a generally accepted linguistic concept. Indeed, the meanings of words are prescribed by human convention, and all languages rely on symbolic processes to associate specific symbols with meanings. However, in natural language processing (NLP), languages are often treated as complex systems of word formation methods and syntactic rules, and their inherent properties as symbols are often overlooked. In this paper, we focus on the symbolic properties of languages and conduct a preliminary investigation. We propose MTLS: a novel pre-training method to improve the multilingual capability of models by *Making Texts into Linguistic Symbols*[1]. Remarkably, MTLS does not rely on extensive multilingual corpora, massive computational resources, or complex a priori knowledge. It can significantly improve the multilingual performance of pre-trained language models (PLMs).

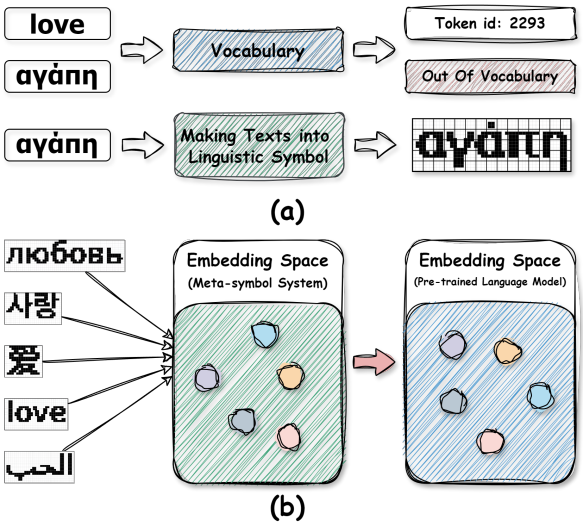

Figure 1: A brief overview of MTLS. (a) illustrates the benefits of employing mapping relations between linguistic symbols and text. (b) demonstrates the meta-symbol system can serve as a bridge between linguistic symbols and the embedding space.

By making texts into linguistic symbols, it is possible to obtain textual embeddings in any language without vocabulary. Pre-trained language models (Devlin et al., 2019; Brown et al., 2020; He et al., 2022), including multilingual models (Conneau et al., 2020; Chi et al., 2022; Xue et al., 2021), assign ids to tokens via vocabulary and then obtain textual embeddings. Language models typ-

[1]The pixel images rendered according to the word composition and writing system are called linguistic symbols.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 3521–3535
November 12-16, 2024

ically support a finite vocabulary of categorical inputs, e.g. characters, subwords or even words, and much effort has been devoted to vocabulary construction (Gerz et al., 2018). As the number of languages the model can handle increases, the vocabulary needs to be extended or reconstructed, which is a burdensome task. Taking into account the symbolic properties of languages, linguistic symbols always correspond to concrete abstract semantics and have unique compositions or structures from a visual perspective. In MTLS, words are first converted into linguistic symbols and rendered as pixel images. Then, by replacing the unique encoding in vocabularies with the linguistic symbols of the words, the problem of multilingual vocabulary construction and out-of-vocabulary (OOV) can be avoided, as shown in Figure 1(a).

Universal meanings are represented by different linguistic symbols in different symbol systems. These universal meanings underlying all human natural languages are referred to as irreducible semantic cores (Wierzbicka, 1999). This semantic core can be used as a semantic bridge for transformations between different languages. Based on this notion, some work (Sherborne and Lapata, 2022; Goswami et al., 2021; Guo et al., 2024) has been done to construct multilingual universal representations by finding universal meanings behind natural languages. We transfer this idea to the linguistic symbolic perspective and call this semantic core the meta-symbol system (MSS). The MSS can be used as a bridge between linguistic symbols of different languages. Therefore, we propose to apply the embedding space of the MSS to represent the linguistic symbols of any language, and then to construct the mapping from the embedding space of the MSS to the embedding space of the PLM, as shown in Figure 1(b). This means that PLMs can handle linguistic symbols in any language and do not need to be pre-trained again.

In this paper, we propose SSS embedding in MTLS for obtaining symbolic embeddings of linguistic symbols and then mapping them to the embedding space of the PLM. SSS embedding consists of three components: Symbolic embedding, Selective embedding, and Spatial embedding. In MTLS, the SSS embedding and the PLM embedding layers are first jointly pre-trained. Then, the SSS embedding replaces the PLM embedding layers, and the multilingual capability of the PLM can be improved. We evaluate the performance of MTLS on syntactic and semantic processing tasks in multiple languages. The results show that MTLS can significantly improve the multilingual capabilities of the model, but at the cost of some performance degradation in Latin script languages. We release the code and models at https://github.com/wenlong1019/MTLS.

2 Related Work

Multilingual Representation Learning is studied for a variety of downstream multilingual tasks. Some approaches rely on rich multilingual corpora to learn multilingual representations through extensive pre-training, such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020). Such extensive pre-training can be extremely computationally intensive, so some work (Pfeiffer et al., 2021; Ansell et al., 2021) has relied on adapter-based methods to fine-tune only the adapter layer during training. Other approaches utilize multilingual parallel corpora to learn generic representations of parallel sentences based on contrastive learning, such as InfoXLM (Chi et al., 2021) and HICTL (Wei et al., 2020). However, obtaining a high quality parallel corpus is difficult, so some work has converted monolingual corpora into parallel corpora through translation (Kvapilíková et al., 2020; Ouyang et al., 2021), or computed semantic similarities in multilingual corpora and used them as supervised labels (Goswami et al., 2021). All these approaches rely heavily on multilingual corpora, while MTLS uses only monolingual data. A further theoretical comparison between MTLS and these methods is given in Appendix A.

Vision-based Embedding differs from mainstream vocabulary-based embedding in that it generates embeddings based on the structure and construction of the text from a visual perspective. Some work (Meng et al., 2019; Li et al., 2021) has used vision-based embedding to obtain some potential features of hieroglyphs on writing systems. There is also some work exploring the potential of vision-based embedding. Rust et al. (2022) proposed a fully vision-based PLM and obtained results competitive with BERT (Devlin et al., 2019). Wang et al. (2024) used vision-based embedding and vocabulary-based embedding to construct a two-tower model to combine the advantages of the two embedding construction methods. The work in this paper is also an exploration of the potential of vision-based embedding.

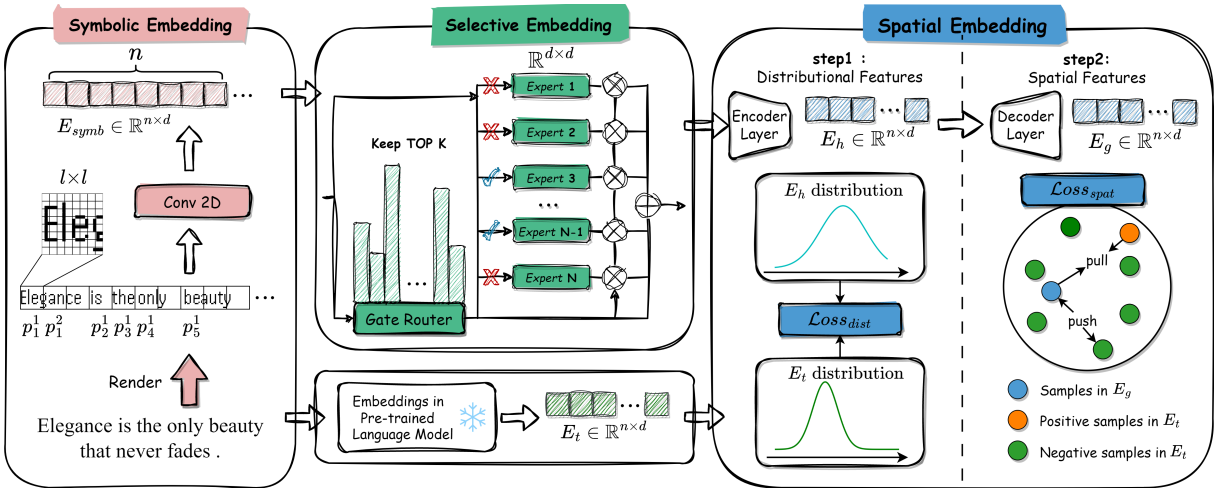

Figure 2: **An overview of our proposed MTLS**. We refer to the combination of symbolic embedding, selective embedding, and spatial embedding as SSS embedding. Symbolic embedding is used to obtain the embeddings of linguistic symbols, selective embedding obtains symbolization-specific bias embeddings of linguistic symbols, and spatial embedding maps embeddings in the space of meta-symbolic systems to the embedding space of PLMs.

3 Methodology

Figure 2 shows an overview of MTLS. MTLS uses the SSS embedding and the frozen embedding layer of the PLM for joint pre-training. Then, by simply replacing the embedding layer in the PLM with SSS embedding, the multilingual capability of the PLM can be improved.

3.1 Symbolic Embedding

The general approach to obtaining textual embeddings is to find the token id corresponding to the vocabulary, and then obtain the token embedding corresponding to the id from the embedding layer. This approach is similar to querying a dictionary, where the goal is to ensure embedding consistency for the same tokens and embedding variability between different tokens. When words are treated as linguistic symbols, consistency and differentiation no longer depend on the vocabulary to be maintained. The same words necessarily have the same glyphs or constructions, and conversely, different words are different.

Linguistic symbols can be created by rendering text into pixel images. The symbolic embeddings of the text are achieved by the patch embedding strategy similar to Vision Transformer (Dosovitskiy et al., 2020), as shown in Figure 2. Specifically, the original text is tokenized to obtain a sequence of tokens $S = [t_1, t_2, \cdots, t_{n-1}, t_n]$. Each token is rendered into a number of fixed-size patches $p \in \mathbb{R}^{l \times l}$ according to the write length, and the patch sequence of each token is denoted by $P_i = \left[p_i^1, p_i^2, \cdots, p_i^{m_i-1}, p_i^{m_i}\right]$, where m_i is the number of patches needed for the token i. The sequential concatenation of all patch sequences is the linguistic symbol of the text, which is also the input $F = [P_1, P_2, \cdots, P_{n-1}, P_n]$. Then, a convolution operation is performed on each patch in the patch sequence to obtain the embedding $v \in \mathbb{R}^d$, where d is the dimension of the embedding. The convolution kernel size is equal to the patch size. The embedding of a single token is $W_i = \left[v_i^1, v_i^2, \cdots, v_i^{m_i-1}, v_i^{m_i}\right]$, and the final symbolic embedding is $E = [W_1, W_2, \cdots, W_{n-1}, W_n]$.

Note that in the token-level task, to ensure that the length of the symbolic embedding sequence is equal to the length of the token sequence, the first patch embedding of each token is taken as the symbolic embedding of that token, giving $E = \left[v_1^1, v_2^1, \cdots, v_{n-1}^1, v_n^1\right]$. In the subsequent section, we notate the symbolic embeddings as $E_{symb} = [v_1, v_2, \cdots, v_{n-1}, v_n]$, $E_{symb} \in \mathbb{R}^{n \times d}$, where n is the length of the symbolic embedding sequence.

3.2 Selective Embedding

Universal meanings follow the symbolization rules to connect linguistic symbols, and different symbol systems have different symbolization rules. However, there are languages in the world with very similar symbolization rules, and different universal meanings may correspond to similar symbols under the same symbolization rules. Symbolic embedding only captures the construction and structural features of symbols, and is unable to perform

some self-adaptation according to the symbolization rules. And the search space resulting from symbolic embedding is clearly insufficient to represent all languages. Therefore, we design selective embedding based on Mixture-of-Experts (MoE) (Eigen et al., 2013; Shazeer et al., 2016). Linguistic symbols activate different experts to obtain symbolization-specific bias embeddings E_{bias}.

Most MoE-based methods (Du et al., 2022; Fedus et al., 2022; Xue et al., 2022) replace the feed-forward component of the Transformer layer (Vaswani et al., 2017) with the MoE layer. Each MoE layer consists of a set of independent feed-forward networks as "experts". Unlike previous work, we set the matrix $M \in \mathbb{R}^{d \times d}$ as the "experts". The symbolic embedding E_{symb} is used as the input to the selective embedding, and the matmul product of the symbolic embedding E_{symb} and the expert matrix M is computed as the result of the expert. The gating function $\mathrm{Gate}\,()$ then uses the softmax activation function to model the probability distribution over these experts. This distribution indicates the probability that each expert can accurately process the incoming symbolic embeddings. The output of the expert i can be expressed as:

$$\mathrm{Expert_i}\,(x) = \mathrm{Gate}\,(x, i) \cdot \mathrm{matmul}\,(x, M_i)\,, \tag{1}$$

$$\mathrm{Gate}\,(x, i) = \mathrm{softmax}\,(x \cdot W_g)\,[i]\,, \tag{2}$$

where x is the input symbolic embedding E_{symb} and W_g is the parameters of the gating function.

During training, the gating function is learnable and is trained to activate the best K experts for each linguistic symbol. During inference, the learned gating unit dynamically selects the best K experts. Therefore, during training, more parameters can be used to represent the symbolic embedding, which can enrich the representation of the patch. During inference, only sparse experts are activated, so there is no need for excessive computation. The final bias embedding of the linguistic symbol is the weighted combination of the outputs from the selected experts. The symbolization-specific bias embeddings E_{bias} can be represented as:

$$E_{bias} = \sum \mathrm{TopK}\,(\mathrm{Expert_i}\,(x)\,, K)\,, \tag{3}$$

$$\mathrm{TopK}\,(v_i, K) = \begin{cases} v_i & \text{if } v_i \text{ in top } K \text{ elements.} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

3.3 Spatial Embedding

The symbolic embedding E_{symb} plus the symbolization-specific bias embedding E_{bias} is denoted as the embedding E_{meta} in the space of MSS. To give the PLM the ability to handle symbolic embeddings, it is necessary to map E_{meta} into the embedding space of the PLM. Simple spatial mapping methods or modules are too coarse to give the model strong generalization capabilities to handle as many languages as possible, even if they have never been seen before. Therefore, we propose spatial embedding, which employs a step-by-step approach to learn the distributional and spatial features of embeddings in the PLM. Specifically, we have designed distributional similarity loss $\mathcal{L}_{dist}$ and spatial similarity loss $\mathcal{L}_{spat}$ in spatial embedding. In addition, we use the patch at the first position as the representation of the whole word in symbolic embedding. To make the embedding of the first patch rich enough, we use the Transformer structure in the spatial embedding to obtain a dynamic context embedding representation.

In step 1, we encode the embedding E_{meta} in the space of MSS into a new embedding $E_h \in \mathbb{R}^{n \times d}$ through the encoder layer of Transformer. The new embedding E_h is constrained by the distributional similarity loss to have the same distribution with the PLM embedding $E_t \in \mathbb{R}^{n \times d}$ to learn the distributional features. This step only constrains the distribution of symbolic embeddings that are free in the representation space. In particular, we compute the Kullback-Leibler (KL) divergence between the symbolic embedding E_h and the embedding E_t for the same text. The distributional similarity loss is defined as follows:

$$P_h\,(m) = \frac{\exp\,(E_h\,(m))}{\sum \exp\,(E_h\,(m))}, \tag{5}$$

$$P_t\,(m) = \frac{\exp\,(E_t\,(m))}{\sum \exp\,(E_t\,(m))}, \tag{6}$$

$$\mathcal{L}_{dist} = \textstyle\sum_{m=1}^{n} P_h\,(m) \log \dfrac{P_h\,(m)}{P_t\,(m)}. \tag{7}$$

The similarity of the distributions does not yet allow PLMs to handle symbolic embeddings accurately. In step 2, E_h is decoded by the decoder layer of Transformer into a new embedding representation space to obtain the embedding $E_g \in \mathbb{R}^{n \times d}$. The spatial similarity loss is then used to constrain the embedding E_g to match the features of the PLM embedding space. First, the embedding E_g and the

embedding E_t are treated as being under the same embedding space, and then contrast learning (He et al., 2020) is used to learn the spatial features. In contrast learning, two embeddings of the same linguistic symbol are positive examples of each other, and the rest of the embeddings are negative examples. The spatial similarity loss is shown as follows:

$$Q^{g2t}(i,j) = \exp\left(\frac{E_g(i)^T E_t(j)}{\tau}\right), \quad (8)$$

$$Q^{g2g}(i,j) = \exp\left(\frac{E_g(i)^T E_g(j)}{\tau}\right), \quad (9)$$

$$\mathcal{L}_{spat} = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{Q^{g2t}(i,i)}{\sum_{j=1;j\neq i}^{n}\left(Q^{g2t}(i,j) + Q^{g2g}(i,j)\right)}, \quad (10)$$

where τ is a temperature parameter.

In pre-training, joint distributional similarity loss and spatial similarity loss are used as the total loss. The full pre-training objective of MTLS is

$$\mathcal{L}_{total} = \mathcal{L}_{dist} + \mathcal{L}_{spat}. \quad (11)$$

4 Experiments

In the experiments of this paper, we use two common monolingual PLMs, BERT [2] (Devlin et al., 2019) and RoBERTa [3] (Liu et al., 2019), respectively, as the backbone for pre-training in MTLS. [4] There is no multilingual corpus in the pre-training, only about 12,000 English data from the Universal Dependencies v2.10 treebanks (Nivre et al., 2020). BERT and RoBERTa with improved multilingual capabilities by MTLS are referred to as MTLS-B and MTLS-R respectively. Parameter settings for pre-training are shown in Table 4.

4.1 Tasks and Languages

We validate the effectiveness of our proposed MTLS by evaluating the syntactic and semantic processing capabilities of the model in multilingual scenarios. The dataset used for the evaluation contains 20 languages, covering most of the language families in the world. Table 5 summarizes the languages and their language families.

To validate the syntactic processing capabilities of the model, we test the performance of MTLS on the part-of-speech (POS) tagging task in multilingual languages. We use data from the Universal Dependencies v2.10 treebanks (Nivre et al., 2020), including the following languages: Arabic (ARA), Basque (EUS), Chinese (ZHO), Coptic (COP), English (ENG), Estonian (EST), Greek (ELL), Hindi (HIN), Japanese (JAN), Korean (KOR), Maltese (MLT), Persian (FAS), Tamil (TAM), Turkish (TUR), Vietnamese (VIE). In addition, to assess the ability to understand the semantics of multilingual texts, we validate the performance of MTLS on the named entity recognition (NER) task. We use the PAN-X (Pan et al., 2017), which includes the following languages: Arabic, Bulgarian (BUL), Chinese, Czech (CES), English, French (FRA), Greek, Japanese, Korean, Persian, Russian (RUS), Tamil, Turkish, Urdu (URD), Vietnamese.

4.2 POS Tagging Task

We compare the performance of BERT and roBERTa with and without MTLS in the multilingual POS tagging task. In order to fully validate the improvement of MTLS on the multilingual capabilities of PLMs, we test the performance in the fine-tune setting and in the cross-language zero-shot setting [5], respectively. We also show the performance of the multilingual models mBERT [6] (Devlin et al., 2019) and XLM-R [7] (Conneau et al., 2020). We show the results of the POS tagging task in Table 1.

Both BERT and RoBERTa perform better after pre-training with MTLS, both in the fine-tune setting and in the cross-language zero-shot setting. In the fine tuning setting, the accuracy of MTLS-B and MTLS-R is significantly improved in ZHO, KOR, and COP. In particular, in COP, MTLS-B and MTLS-R achieve nearly 66.6% and 52.2% accuracy improvements respectively. This demonstrates that our proposed MTLS can significantly improve the multilingual syntactic processing ability of the model. In the cross-language zero-shot setting, the improvement brought by MTLS is relatively average. We attribute this to the fact that multilingual data is not used in the pre-training of MTLS and is not visible to MTLS-B and MTLS-R during cross-

[2] https://huggingface.co/google-bert/bert-base-cased.

[3] https://huggingface.co/FacebookAI/roberta-base.

[4] BERT and RoBERTa are chosen because they do not use multilingual corpura in pre-training and do not have a strong multilingual representation capability.

[5] We first fine-tune it on the English corpus and then transfer directly to other languages for inference.

[6] https://huggingface.co/google-bert/bert-base-multilingual-cased.

[7] https://huggingface.co/FacebookAI/xlm-roberta-base.

Model	Fine-Tune								Zero-Shot						
	ARA	ZHO	COP	ENG*	KOR	TAM	VIE*	AVG	ARA	ZHO	COP	KOR	TAM	VIE*	AVG
mBERT	94.2	58.5	26.9	93.3	96.3	87.0	82.5	77.0	55.1	66.6	5.2	56.0	53.5	57.6	49.0
XLM-R	97.2	96.9	26.6	97.4	97.0	87.7	93.4	85.2	72.3	48.0	5.9	61.0	62.2	60.3	51.6
BERT	93.8	59.4	26.7	**97.1**	60.3	45.5	**83.8**	66.6	**15.6**	14.3	4.0	13.3	14.2	**41.9**	17.2
MTLS-B	**93.9**	**88.1**	**93.3**	91.4	**90.0**	**67.4**	77.5	**86.0**	12.5	**28.9**	**17.5**	**23.4**	**27.9**	34.0	**24.0**
RoBERTa	91.5	74.4	40.8	**97.3**	55.3	**67.9**	**86.1**	73.3	11.7	25.5	9.5	15.6	16.0	**39.5**	19.6
MTLS-R	**92.9**	**80.1**	**93.0**	90.7	**81.9**	63.6	76.3	**82.6**	**12.5**	**26.6**	**18.5**	**18.5**	**26.4**	31.8	**22.4**

Table 1: Results of the POS tagging task in the fine-tune setting and in the zero-shot setting. Accuracy is used as the evaluation metric. * indicates that the language with Latin scripts. We show partial results in this table and complete experimental results are shown in Table 6.

language zero-shot, making it difficult to activate even strong multilingual capabilities in this setup.

The trade-off for such a significant performance gain is performance degradation for languages with Latin scripts. Both MTLS-B and MTLS-R have some performance degradation in ENG and VIE, either in the fine-tune setting or in the zero-shot setting. The performance degradation is due to the fact that the embedding layers of the original BERT and RoBERTa are replaced by the SSS embedding. The integrity of the models is compromised, with a reduced ability to process languages with Latin scripts. In the pre-training of MTLS, only 12,000 English data are used, which is far less than the pre-training data in BERT and RoBERTa, and thus not enough to recover the original English processing ability.

It should be emphasised that BERT and RoBERTa perform very poorly in non-Latin script languages due to the low coverage of vocabulary in these languages, which creates a noticeable OOV problem. MTLS uses symbolic embedding instead of vocabulary, so there is no OOV problem, which is an advantage of MTLS. Only a small amount of English data is used in the pre-training of MTLS, and subwords that do not appear in the pre-training dataset are also unseen for MTLS-B and MTLS-R. Therefore, we believe it is fair to compare BERT and MTLS-B, RoBERTa and MTLS-R, respectively. In addition, we further explore the effect of symbolic embedding on the model in Section 5.2.

Compared to mBERT and XLM-R, MTLS-B and MTLS-R perform worse. However, considering that mBERT and XLM-R use massive multilingual corpora for pre-training, while there is only a very small amount of monolingual data in MTLS, we believe that MTLS still has some advantages. In particular, MTLS-B and MTLS-R significantly outperform mBERT and XLM-R in COP. This result can be attributed to the fact that mBERT and XLM-R do not use Coptic data in their pre-training. The multilingual model still suffers from a sudden drop in performance when confronted with an unseen language, even after a lot of resources and effort have been spent in pre-training. There is a limit to the multilingual capability of the model obtained by pre-training on a large multilingual corpus. MTLS does not use any multilingual corpus and still achieves better performance in unseen languages. This demonstrates the powerful generalization ability of MTLS.

4.3 NER Task

To evaluate the effect of MTLS on understanding the semantics of multilingual texts, we compare the performance of the models on the multilingual NER task. We use the same experimental setup as in the POS tagging task and present the results of the NER task in Table 2.

In the NER task, MTLS-B and MTLS-R still significantly outperform BERT and RoBERTa in most languages in both monolingual fine-tuning and cross-lingual zero-shot, while performing poorly in Latin-written languages such as ENG and VIE. This result is consistent with experimental results in the POS tagging task. Overall, MTLS significantly improves the ability of the model to understand multilingual semantics.

It is worth noting that MTLS-B and MTLS-R show smaller performance gains on the NER task, with an average F1 score gain of less than 10 in the fine tuning setting. According to our research study, the reason may be the replacement of vocabulary-based embedding layer with symbolic embedding. In previous work, Rust et al. (2022) found that vision-based PLMs outperform vocabulary-based

Model	Fine-Tune									Zero-Shot							
	ARA	BUL	ENG*	KOR	RUS	URD	VIE*	ZHO	AVG	ARA	BUL	KOR	RUS	URD	VIE*	ZHO	AVG
mBERT	85.0	89.2	82.3	85.4	86.0	49.8	91.4	83.1	81.5	38.6	73.3	52.4	57.9	23.2	64.7	38.9	49.8
XLM-R	86.3	87.5	81.0	85.1	85.5	71.0	91.3	79.2	83.4	31.5	69.5	37.5	56.2	21.4	63.8	19.8	42.8
BERT	54.9	66.6	**81.5**	47.0	60.1	32.8	**80.5**	55.8	59.9	1.3	6.8	4.8	8.0	1.2	**43.2**	1.6	9.6
MTLS-B	**70.7**	**73.9**	63.0	**66.4**	**68.9**	**56.3**	75.9	**64.5**	**67.4**	**5.0**	**33.4**	**6.1**	**20.0**	**3.2**	31.3	**2.2**	**14.5**
RoBERTa	43.6	58.7	**80.0**	59.2	52.6	29.7	71.0	**71.5**	58.3	0.5	3.7	4.3	6.5	0.2	**36.7**	1.2	7.6
MTLS-R	**64.4**	**70.2**	56.1	**64.7**	**65.0**	**54.7**	**73.5**	63.8	**64.0**	**1.5**	**29.8**	**5.2**	**18.3**	**3.2**	27.7	**1.9**	**12.5**

Table 2: Results of the NER task in the fine-tune setting and in the zero-shot setting. F1 score is used as the evaluation metric. * indicates that the language with Latin scripts. We show partial results in this table and complete experimental results are shown in Table 7.

Model	POS		NER	
	FT	ZS	FT	ZS
MTLS-B	**88.6**	**26.9**	**68.1**	**18.8**
— w/o PT	30.3	23.2	4.0	1.6
— w/o SE	75.9	23.1	61.4	16.0
— w/o DSL	80.7	24.2	67.4	18.4
— w/o SSL	42.4	23.9	8.4	0.9

Table 3: Results of the ablation study on MTLS. The table shows the average of the results for 15 languages separately, and the complete results are shown in Table 8 and Table 9. In the POS tagging task, accuracy is the evaluation metric. In the NER task, the F1 score is the evaluation metric. FT indicates fine-tune and ZS indicates zero-shot.

models in syntactic tasks and lag behind in semantic tasks. Our experimental results are consistent with this finding. Therefore, we believe that MTLS can lead to more significant performance gains in syntactic tasks.

4.4 Ablation Study

To evaluate the effect of each component, we perform the ablation study of MTLS in the POS tagging and NER task. For ease of analysis, we make the following definitions:

1) PT denotes pre-training on an English dataset;
2) SE denotes selective embedding;
3) DSL denotes distributional similarity loss;
4) SSL denotes spatial similarity loss.

As shown in Table 3, MTLS significantly outperforms MTLS w/o * (* indicates components) in both tasks, demonstrating the importance of each component in MTLS. Although only 12,000 English data are used in the pre-training, the effect of the pre-training is extremely significant. In the fine-tune setting, pre-training results in an average accuracy improvement of nearly 58.3% in the POS tagging task, and there is an average F1 score improvement of 64.1 in the NER task. Such results reflect the importance of pre-training and the effectiveness of the two pre-training losses. These results are also predictable, because the non-embedded part of the PLM is unable to handle direct symbolic embeddings, and pre-training is needed to constrain the mapping of symbolic embeddings into the embedding representation space of the PLM.

In the fine-tune setting, not using selective embedding (w/o SE) results in a 12.7% decrease in average accuracy in the POS tagging task and a 6.7 f1 score decrease on average in the NER task. This is because selective embedding obtains a bias embedding to the symbolization process for each linguistic symbol, expanding the representation space of the MSS. A larger representation space means that the symbolic embedding can contain more semantics.

Using distributional similarity loss without spatial similarity loss (w/o SSL) can produce very poor results. Spatial embedding employs a step-by-step strategy. Step 1 uses distributional similarity loss, and step 2 uses spatial similarity loss. The distributional similarity loss only limits the distribution of the embedding, and the spatial similarity loss also limits the distribution to some extent, but also greatly limits the representation space of the embedding. Therefore, not using spatial similarity loss results in the embedding of the MSS not being accurately mapped to the embedding space of the PLM, and the PLM being unable to process the symbolic embedding. In contrast, using only spatial similarity loss (w/o DSL) may result in reduced generalization of the model due to overly stringent constraints, thus affecting performance.

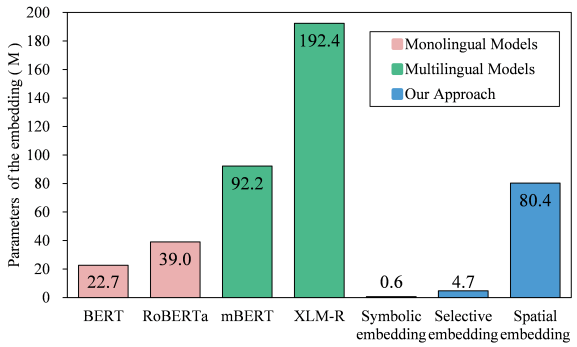

Figure 3: Parameter comparison between SSS embedding in MTLS and embeddings of PLMs.

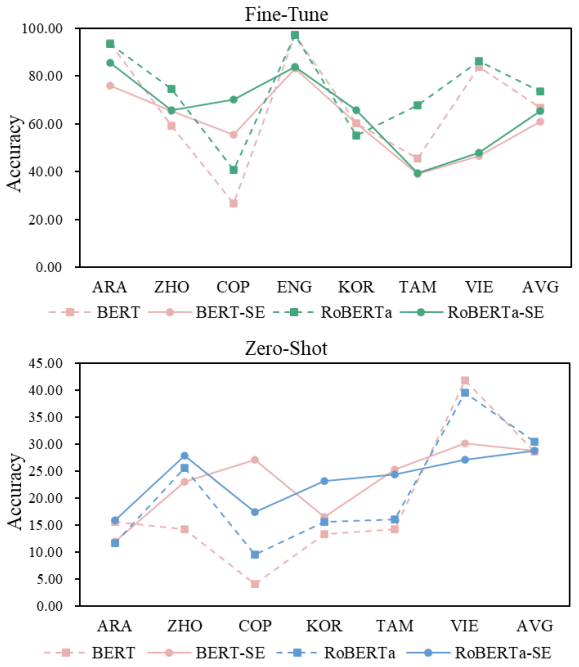

Figure 4: Results of whether or not to use symbolic embedding in the multilingual POS tagging task in the fine-tune and zero-shot settings, respectively. -SE indicates the symbolic embedding. Complete experimental results are shown in Table 10.

5 Further analysis

5.1 Parameter Analysis

We further analyze MTLS on the number of parameters. Since the SSS embedding is used to replace the PLM embedding layer, we only compare the parameters of the PLM embedding layer with those of the SSS embedding, as shown in Figure 3. The embedding parameters of BERT and RoBERTa are much less than the embedding parameters of multilingual models. This is because the vocabulary in multilingual models needs to cover all the subwords of the language as much as possible, and the increase of the subwords means the increase of the embedding parameters. And the number of parameters in SSS embedding is between the multilingual and monolingual models, which we think is acceptable. Multilingual models perform better with a larger number of parameters, but their dependence on large multilingual corpora is one of their main limitations. However, MTLS can significantly improve the multilingual capabilities of monolingual models without multilingual corpora.

5.2 Effects of Symbolic Embedding

To explore the effect of symbolic embedding on PLMs, we replace the embedding layers of BERT and RoBERTa with symbolic embedding, called BERT-SE and RoBERTa-SE. The performances of BERT and BERT-SE, RoBERTa and RoBERTa-SE are compared on multilingual POS tagging tasks, as shown in Figure 4.

Replacing vocabulary-based embedding with symbolic embedding alone is not effective in improving the performance of the PLM in the multilingual task, demonstrating the importance of selective embedding and spatial embedding. We believe the reasons for the lack of significant results are that the symbolic embedding is completely randomly initialized and has far fewer parameters than the original embedding. The parameters of symbolic embedding are only about 2.6% of the embedding in BERT, and about 1.5% of the embedding in RoBERTa. However, in the fine-tune setting, BERT-SE outperforms BERT in ZHO and COP. RoBERTa-SE also outperforms RoBERTa in COP and KOR. Combined with the significant improvement in multilingual capability for PLMs brought by MTLS, we argue that symbolic embedding is one of the key factors for the effectiveness of MTLS. Furthermore, we argue that symbolic embedding has natural advantages for multilingual tasks.

6 Conclusion

In this paper, we explore the properties of languages as symbolic systems and propose MTLS: a pre-training method to improve the multilingual capability of models by *Making Texts into Linguistic Symbols*. We also propose SSS Embedding, which can obtain the symbolic embedding of any language and map the symbolic embedding to the embedding space of the PLM. By replacing the PLM embedding layer with SSS embedding, the model can be made to process linguistic symbols

in any language. It should be emphasized that the pre-training of MTLS requires only a small amount of monolingual data and does not require multilingual corpora. In addition, MTLS reuses most of the parameters in the PLM, so pre-training costs few computational resources. Our experimental results show that MTLS can significantly improve the multilingual capability of PLMs, at the cost of some performance degradation in Latin-script languages.

7 Limitations

In this paper, we focus on the symbolic properties of languages and propose MTLS. Our results show that MTLS has good performance, but this is only a preliminary investigation, and there are some areas for further in-depth study. Here we highlight the limitations of our current work and the direction of our future work.

• It is feasible to use multilingual corpora in pre-training for MTLS. However, PLMs that use monolingual corpora for pre-training (e.g. BERT and RoBERTa) do not contain multilingual subwords in their vocabularies. In order to learn the embedding space of the model, only the corpus of the corresponding language can be used for pre-training.

• For multilingual PLMs, it is obvious that MTLS for monolingual pre-training only does not lead to any improvement, as we have also verified in our previous work. MTLS for multilingual pre-training cannot avoid the dependence on multilingual corpora, which goes against our original intention. Of course, we expect that the multilingual capability of the model can be improved by MTLS for multilingual pre-training, but this remains to be verified.

• The non-embedding parameters of the PLM are reused in MTLS to reduce the computational consumption. However, by simply modifying MTLS for multilingual pre-training, it is possible to construct a multilingual model based entirely on linguistic symbols. We will investigate this in future work.

There are also limitations and possible future research directions for making texts into linguistic symbols.

• Rendering text into linguistic symbols leads to a hundreds-fold increase in the storage capacity of the data. This puts a strain on computational memory during training and inference.

• Using symbolic embeddings instead of vocabularies results in models that are unable to generate discrete words for the generation task.

• This paper provides a preliminary exploration of symbolic embedding as an alternative to vocabulary, and does not go into great depth on some details. For example, the choice of fonts and the clarity of linguistic symbols. More advanced methods for encoding linguistic symbolic embeddings may yield better results.

References

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. Mad-g: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4762–4781.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, He-Yan Huang, and Ming Zhou. 2021. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588.

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Bo Zheng, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, He-Yan Huang, et al. 2022. Xlm-e: Cross-lingual language model pre-training via electra. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6170–6182.

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*. Association for Computational Linguistics.

Ferdinand De Saussure. 1989. *Cours de linguistique générale*, volume 1. Otto Harrassowitz Verlag.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.

David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. 2013. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Daniela Gerz, Ivan Vulić, Edoardo Maria Ponti, Roi Reichart, and Anna Korhonen. 2018. On the relation between linguistic typology and (limitations of) multilingual language modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 316–327.

Koustava Goswami, Sourav Dutta, Haytham Assem, Theodorus Fransen, and John Philip McCrae. 2021. Cross-lingual sentence embedding using multi-task learning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9099–9113.

Ping Guo, Xiangpeng Wei, Yue Hu, Baosong Yang, Dayiheng Liu, Fei Huang, et al. 2024. Emma-x: An em-like multilingual pre-training algorithm for cross-lingual representation learning. *Advances in Neural Information Processing Systems*, 36.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2022. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

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

Ivana Kvapilíková, Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Ondřej Bojar. 2020. Unsupervised multilingual sentence embeddings for parallel corpus mining. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 255–262.

Yunxin Li, Yu Zhao, Baotian Hu, Qingcai Chen, Yang Xiang, Xiaolong Wang, Yuxin Ding, and Lin Ma. 2021. Glyphcrm: Bidirectional encoder representation for chinese character with its glyph. *arXiv preprint arXiv:2107.00395*.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for chinese character representations. *Advances in Neural Information Processing Systems*, 32.

Joakim Nivre, Filip de Marneffe, Marie-Catherine an Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie-m: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 27–38.

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

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. Unks everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203.

Phillip Rust, Jonas F Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2022. Language modelling with pixels. In *The Eleventh International Conference on Learning Representations*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2016. Outrageously large neural networks:

The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Tom Sherborne and Mirella Lapata. 2022. Zero-shot cross-lingual semantic parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4134–4153.

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

Xiaohua Wang, Wenlong Fei, Min Hu, Qingyu Zhang, and Aoqiang Zhu. 2024. Mevtr: A multilingual model enhanced with visual text representations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11247–11261.

Xiangpeng Wei, Rongxiang Weng, Yue Hu, Luxi Xing, Heng Yu, and Weihua Luo. 2020. On learning universal representations across languages. In *International Conference on Learning Representations*.

Anna Wierzbicka. 1999. *Emotions across languages and cultures: Diversity and universals*. Cambridge university press.

Fuzhao Xue, Ziji Shi, Futao Wei, Yuxuan Lou, Yong Liu, and Yang You. 2022. Go wider instead of deeper. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8779–8787.

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.

A Theoretical Comparison

To compare our proposed MTLS with previous multilingual representation learning approaches, the PLM is denoted as $f(\cdot; E; N)$, where $\cdot$ denotes the input, E denotes the parameters of the embedding layer, and N denotes the parameters of the non-embedding. The pre-training of the model is defined as:

$$E, N = \arg_{(E,N)} \min \left(\mathcal{L} \left[f(x; E; N), g(x) \right] \right), \tag{12}$$

where $x \in D$, $\mathcal{L}$ denotes the pre-training loss function, x denotes the training data, $g(x)$ denotes the ground truth of x under the pre-training task, and D denotes the corpus.

In the modeling setup above, we represent the multilingual model as $f_m(\cdot; E_m; N_m)$, where E_m and N_m are the parameters in the multilingual model. Methods (e.g., mBERT and XLM-R) for constructing multilingual models by multilingual corpora can be represented as:

$$E_m, N_m = \arg_{(E_m,N_m)} \min \left(\mathcal{L} \left[f(x; E_m; N_m), g(x) \right] \right), \tag{13}$$

where $x \in D_m$, D_m is the multilingual corpus. Most multilingual modeling approaches using parallel corpora can also be represented by Equation 13, but in the parallel corpus approaches $x = x_p + x_q$, $x_p \in D_p$, $x_q \in D_q$. D_p and D_q are corpora of different languages. With Equation 12 and 13, it is easy to see that the essence of the above approach to learning multilingual representations is to rely on a multilingual corpus to fine-tune the parameters of the language model $f(\cdot; E; N) \Rightarrow f_m(\cdot; E_m; N_m)$. Obviously, these methods require not only a large multilingual corpus, but also the fine-tuning of almost all the parameters of the model.

In the adapter-based approaches, the monolingual model $f_s(\cdot; E_s; N_s)$ is used as a backbone, which does not change the parameters of the language model and trains only the adapter layers that are inserted into the model, thus greatly reducing the training consumption. The adapter-based language model can be represented as $f_s(\cdot; E_s; N_s; A)$, where A denotes the parameters in the adapter layers. The methods for constructing multilingual models based on the adapter layers can be represented as $f_s(\cdot; E_s; N_s) \Rightarrow f_m(\cdot; E_s; N_s; A)$, and the training process can be represented as:

$$A = \arg_{(A)} \min \left(\mathcal{L} \left[f(x; E_s; N_s; A), g(x) \right] \right), \tag{14}$$

where $x \in D_m$.

All of the above methods require the support of multilingual corpura in the training process. In addition, these methods are limited by the vocabulary. The multilingual vocabulary must be reconstructed before training. In contrast, our proposed MTLS avoids vocabulary construction by the symbolic embedding.

Pre-training in MTLS can be expressed as: $f_s(\cdot; E_s; N_s) \Rightarrow f_m(\cdot; E_x; N_s)$, where E_x denotes SSS Embedding. In most PLMs, non-embeddings occupy a larger proportion of the parameters, so the parameters N_s of non-embeddings are not involved in the pre-training of MTLS. Another difference between MTLS and previous work is that all previous work required large multilingual datasets for support, whereas we use only a small amount of data in a single language for training. The pre-training of MTLS can be expressed as:

$$E_x = \arg_{(E_x)} \min \left(\mathcal{L} \left[f(x; E_x; N_s; A), g(x) \right] \right), \tag{15}$$

where $x \in D_s$, D_s is a monolingual corpus.

Parameter	**Value**
Patch size l	16
Embedding dimension d	768
Experts number N	8
Selected experts number K	2
Temperature parameter τ	0.2
Encoder hidden size	768
Encoder feedforward dimension	2048
Encoder num layers	6
Decoder hidden size	768
Decoder feedforward dimension	2048
Decoder num layers	6
Dropout probability	0.1
Hidden activation	ReLU
Learning rate	5e-5
Optimizer	AdamW (Kingma and Ba, 2014; Loshchilov and Hutter, 2019)
Adam betas	(0.9, 0.999)
Adam epsilon	1e-8
Weight decay	0.05
Training steps	30,000
Batch size	128

Table 4: Parameter settings for pre-training in MTLS.

Language	ISO 639-3	Language Family	Script
Arabic	ARA	Afro-Asiatic	Arabic
Basque	EUS	Language Isolate	Latin
Bulgarian	BUL	Indo-European	Cyrillic
Chinese	ZHO	Sino-Tibetan	Chinese
Coptic	COP	Afro-Asiatic	Coptic
Czech	CES	Indo-European	Latin
English	ENG	Indo-European	Latin
Estonian	EST	Uralic	Latin
French	FRA	Indo-European	Latin
Greek	ELL	Indo-European	Greek
Hindi	HIN	Indo-European	Devanagari
Japanese	JAN	Japonic	Japanese
Korean	KOR	Koreanic	Korean
Maltese	MLT	Afro-Asiatic	Latin
Persian	FAS	Indo-European	Persian
Russian	RUS	Indo-European	Cyrillic
Tamil	TAM	Dravidian	Tamil
Turkish	TUR	Turkic	Latin
Urdu	URD	Indo-European	Perso-Arabic
Vietnamese	VIE	Austro-Asiatic	Latin

Table 5: Overview of languages in our experiments, including language families and scripts.

Model	ARA	EUS*	ZHO	COP	ENG*	EST*	ELL	HIN	JAN	KOR	MLT*	FAS	TAM	TUR*	VIE*	AVG
Fine-Tune																
mBERT	94.2	93.7	58.5	26.9	93.3	95.3	95.8	86.0	98.4	96.3	90.9	95.9	87.0	89.5	82.5	85.6
XLM-R	97.2	96.5	96.9	26.6	97.4	97.7	98.2	98.0	98.8	97.0	94.7	96.5	87.7	92.1	93.4	91.2
BERT	93.8	**95.1**	59.4	26.7	**97.1**	**96.5**	**95.9**	86.6	88.1	60.3	**93.4**	**96.1**	45.5	**90.7**	**83.8**	80.6
MTLS-B	**93.9**	89.9	**88.1**	**93.3**	91.4	93.1	90.6	**94.4**	**96.0**	**90.0**	85.2	94.7	**67.4**	83.9	77.5	**88.6**
RoBERTa	91.5	**93.7**	74.4	40.8	**97.3**	**95.7**	**94.3**	**95.6**	86.7	55.3	**93.0**	**94.6**	**67.9**	**89.9**	**86.1**	83.8
MTLS-R	**92.9**	87.8	**80.1**	**93.0**	90.7	90.7	89.8	92.7	**94.2**	**81.9**	83.2	94.3	63.6	81.5	76.3	**86.2**
Zero-Shot																
mBERT	55.1	64.3	66.6	5.2	-	73.9	70.0	62.3	50.4	56.0	19.2	71.2	53.5	66.7	57.6	55.1
XLM-R	72.3	69.6	48.0	5.9	-	83.6	83.1	69.9	33.9	61.0	25.8	76.4	62.2	74.3	60.3	59.0
BERT	**15.6**	34.1	14.3	4.0	-	**41.9**	26.0	14.0	13.1	13.3	24.4	**17.6**	14.2	**42.9**	**41.9**	22.7
MTLS-B	12.5	**39.8**	**28.9**	**17.5**	-	40.9	**29.1**	**19.1**	**21.9**	**23.4**	**30.4**	11.5	**27.9**	39.5	34.0	**26.9**
RoBERTa	11.7	**42.3**	25.5	9.5	-	**44.4**	18.9	2.8	**26.7**	15.6	30.0	9.4	16.0	**40.2**	**39.5**	23.7
MTLS-R	**12.5**	37.4	**26.6**	**18.5**	-	39.5	**31.1**	**16.7**	17.5	**18.5**	**34.1**	**12.2**	**26.4**	37.4	31.8	**25.7**

Table 6: Complete results of the POS tagging task in the fine tune setting and in the zero shot setting. Accuracy is used as the evaluation metric. * indicates that the language with Latin scripts.

Model	ARA	BUL	CES*	ELL	ENG*	FAS	FRA*	JAN	KOR	RUS	TAM	TUR*	URD	VIE*	ZHO	AVG
Fine-Tune																
mBERT	85.0	89.2	90.0	85.4	82.3	90.2	88.4	59.3	85.4	86.0	77.8	90.6	49.8	91.4	83.1	82.3
XLM-R	86.3	87.5	89.5	88.6	81.0	91.5	86.0	71.2	85.1	85.5	83.5	91.1	71.0	91.3	79.2	84.5
BERT	54.9	66.6	**82.6**	70.9	**81.5**	49.1	**83.7**	58.3	47.0	60.1	46.8	**85.1**	32.8	**80.5**	55.8	63.7
MTLS-B	**70.7**	**73.9**	75.5	**75.0**	63.0	**70.2**	71.1	**58.4**	**66.4**	**68.9**	**57.4**	74.2	**56.3**	75.9	**64.5**	**68.1**
RoBERTa	43.6	58.7	**80.9**	56.2	**80.0**	32.9	**83.0**	**68.6**	59.2	52.6	22.9	**83.9**	29.7	71.0	**71.5**	59.6
MTLS-R	**64.4**	**70.2**	72.0	**71.0**	56.1	**69.6**	67.8	57.0	**64.7**	**65.0**	**54.0**	67.1	**54.7**	**73.5**	63.8	**64.7**
Zero-Shot																
mBERT	38.6	73.3	75.6	63.3	-	34.2	74.6	23.2	52.4	57.9	41.1	65.9	23.2	64.7	38.9	51.9
XLM-R	31.5	69.5	71.4	64.4	-	32.9	69.2	9.6	37.5	56.2	43.7	64.0	21.4	63.8	19.8	46.8
BERT	1.3	6.8	**45.0**	7.7	-	0.7	**54.5**	0.9	4.8	8.0	0.5	36.8	1.2	**43.2**	1.6	15.2
MTLS-B	**5.0**	**33.4**	40.4	**26.3**	-	**2.6**	44.9	**1.1**	**6.1**	**20.0**	**8.3**	**37.9**	**3.2**	31.3	**2.2**	**18.8**
RoBERTa	0.5	3.7	**45.8**	5.6	-	0.3	**56.0**	**1.8**	4.3	6.5	3.9	**38.0**	0.2	**36.7**	1.2	14.6
MTLS-R	**1.5**	**29.8**	38.7	**24.4**	-	**1.3**	38.7	1.0	**5.2**	**18.3**	**6.5**	33.5	**3.2**	27.7	**1.9**	**16.6**

Table 7: Complete results of the NER task in the fine-tune setting and in the zero-shot setting. F1 score is used as the evaluation metric. * indicates that the language with Latin scripts.

Model	ARA	EUS*	ZHO	COP	ENG*	EST*	ELL	HIN	JAN	KOR	MLT*	FAS	TAM	TUR*	VIE*	AVG
Fine-Tune																
MTLS-B	**93.9**	**89.9**	**88.1**	**93.3**	**91.4**	**93.1**	**90.6**	94.4	**96.0**	90.0	**85.2**	**94.7**	**67.4**	**83.9**	77.5	**88.6**
— w/o PT	33.8	28.8	27.6	16.5	26.8	26.2	21.7	34.1	36.0	51.6	19.4	42.3	26.5	30.9	32.1	30.3
— w/o SE	85.8	63.6	85.0	92.8	78.8	73.5	68.3	92.7	93.7	77.3	68.6	86.2	42.7	68.1	60.9	75.9
— w/o DSL	87.0	67.6	84.2	92.7	80.1	76.2	71.2	**94.5**	95.9	**90.3**	70.6	87.0	63.1	71.6	**78.4**	80.7
— w/o SSL	33.8	25.2	27.6	15.9	26.5	26.2	21.5	94.4	91.9	57.5	26.0	86.6	26.5	31.0	45.5	42.4
Zero-Shot																
MTLS-B	12.5	**39.8**	28.9	17.5	-	**40.9**	**29.1**	19.1	**21.9**	**23.4**	**30.4**	11.5	**27.9**	**39.5**	**34.0**	**26.9**
— w/o PT	**33.7**	23.9	28.9	16.8	-	28.4	22.8	14.2	17.3	15.3	22.6	12.1	26.6	31.8	30.0	23.2
— w/o SE	12.1	31.6	**29.1**	13.5	-	33.6	23.0	19.4	19.1	22.3	24.2	13.8	21.5	34.0	26.0	23.1
— w/o DSL	18.9	32.6	28.2	13.5	-	34.1	22.7	18.4	19.8	22.2	25.4	**16.3**	19.7	36.8	30.2	24.2
— w/o SSL	22.3	29.9	25.3	**18.3**	-	33.7	25.0	**19.5**	14.7	16.7	25.4	14.7	26.6	38.4	24.5	23.9

Table 8: Complete results of the ablation study on the POS tagging task. Accuracy is used as the evaluation metric. * indicates that the language with Latin scripts.

Model	ARA	BUL	CES*	ELL	ENG*	FAS	FRA*	JAN	KOR	RUS	TAM	TUR*	URD	VIE*	ZHO	AVG
Fine-Tune																
MTLS-B	**70.7**	**73.9**	**75.5**	**75.0**	**63.0**	70.2	**71.1**	**58.4**	**66.4**	**68.9**	57.4	**74.2**	**56.3**	**75.9**	64.5	**68.1**
— w/o PT	0.0	0.0	0.1	14.6	2.5	8.9	0.1	10.3	6.4	0.1	0.0	0.0	3.9	0.1	13.3	4.0
— w/o SE	64.6	70.5	70.3	70.7	54.5	65.6	65.3	50.7	50.4	62.8	52.0	65.9	53.3	72.2	52.3	61.4
— w/o DSL	70.5	72.5	75.2	74.1	60.7	**72.7**	69.5	57.2	65.2	67.3	**58.1**	73.4	54.8	74.1	**65.0**	67.4
— w/o SSL	0.0	0.0	0.1	0.0	4.2	8.4	0.1	57.2	46.0	0.1	0.0	0.0	0.0	0.1	10.1	8.4
Zero-Shot																
MTLS-B	**5.0**	33.4	**40.4**	**26.3**	-	**2.6**	**44.9**	1.1	6.1	**20.0**	8.3	**37.9**	3.2	**31.3**	2.2	**18.8**
— w/o PT	4.2	0.9	1.4	0.9	-	0.7	2.3	0.2	2.4	1.9	1.2	1.1	0.1	4.8	0.2	1.6
— w/o SE	2.2	27.9	34.7	24.0	-	1.1	37.5	1.5	6.8	17.4	7.2	33.2	1.3	26.9	**2.9**	16.0
— w/o DSL	2.7	**33.8**	40.2	**26.3**	-	1.9	42.7	**1.6**	**7.1**	19.6	**9.2**	37.7	**3.7**	28.8	2.5	18.4
— w/o SSL	0.0	1.2	1.6	0.7	-	0.0	3.8	0.0	0.0	2.0	0.6	0.5	0.0	1.7	0.0	0.9

Table 9: Complete results of the ablation study on the NER task. F1 score is used as the evaluation metric. * indicates that the language with Latin scripts.

Model	ARA	EUS*	ZHO	COP	ENG*	EST*	ELL	HIN	JAN	KOR	MLT*	FAS	TAM	TUR*	VIE*	AVG
Fine-Tune																
BERT	**93.8**	**95.1**	59.4	26.7	**97.1**	**96.5**	**95.9**	**86.6**	**88.1**	**60.3**	**93.4**	**96.1**	**45.5**	**90.7**	**83.8**	**80.6**
BERT-SE	76.1	68.8	**65.4**	**55.6**	83.1	75.1	56.9	82.4	83.8	**60.3**	44.1	83.7	38.9	62.4	46.5	65.5
RoBERTa	**91.5**	**93.7**	**74.4**	40.8	**97.3**	**95.7**	**94.3**	**95.6**	**86.7**	55.3	**93.0**	**94.6**	**67.9**	**89.9**	**86.1**	**83.8**
RoBERTa-SE	85.5	70.8	65.8	**70.3**	83.9	71.5	76.3	83.0	84.8	**65.6**	19.2	86.0	39.2	62.0	47.9	67.5
Zero-Shot																
BERT	**15.6**	34.1	14.3	4.0	-	**41.9**	26.0	14.0	13.1	13.3	24.4	**17.6**	14.2	**42.9**	**41.9**	22.7
BERT-SE	11.9	**38.3**	**23.0**	**27.0**	-	40.0	**31.7**	**16.4**	**14.8**	**16.4**	**29.8**	12.1	**25.3**	36.9	30.1	**25.3**
RoBERTa	11.7	**42.3**	25.5	9.5	-	**44.4**	18.9	2.8	**26.7**	15.6	**30.0**	9.6	16.0	**40.2**	**39.5**	23.7
RoBERTa-SE	**15.9**	37.9	**27.8**	**17.4**	-	40.0	**30.2**	**15.3**	20.4	**23.2**	29.1	**14.2**	**24.3**	36.9	27.0	**25.7**

Table 10: Results of whether or not to use symbolic embedding in the multilingual POS tagging task in the fine-tune and zero-shot settings, respectively. -SE indicates the symbolic embedding.