Structural Contrastive Pretraining for Cross-Lingual Comprehension

Nuo Chen[1], Linjun Shou[2], Tengtao Song[3], Ming Gong[2], Jian Pei[4]
Jianhui Chang[3], Daxin Jiang[2], Jia Li[1]*
[1]Hong Kong University of Science and Technology (Guangzhou),
Hong Kong University of Science and Technology
[2]STCA, Microsoft, Beijing, [3]Peking University, China
[4] Duke University, USA
chennuo26@gmail.com, jialee@ust.hk

Abstract

Multilingual language models trained using various pre-training tasks like mask language modeling (MLM) have yielded encouraging results on a wide range of downstream tasks. Despite the promising performances, structural knowledge in cross-lingual corpus is less explored in current works, leading to the semantic misalignment. In this paper, we propose a new pre-training task named Structural Contrast Pretraining (SCP) to align the structural words in a parallel sentence, improving the models' linguistic versatility and their capacity to understand representations in multilingual languages. Concretely, SCP treats each structural word in source and target languages as a positive pair. We further propose Cross-lingual Momentum Contrast (CL-MoCo) to optimize negative pairs by maintaining a large size of the queue. CL-MoCo extends the original MoCo approach into cross-lingual training and jointly optimizes the source-to-target language and target-to-source language representations in SCP, resulting in a more suitable encoder for cross-lingual transfer learning. We conduct extensive experiments and prove the effectiveness of our resulting model, named **XLM-SCP**, on three cross-lingual tasks across five datasets such as MLQA, WikiAnn. Our codes are available at https://github.com/nuochenpku/SCP.

1 Introduction

Following the promising results of the pre-training paradigm in the monolingual natural language domain, the efforts of multilingual pre-trained language models (xPLMs) (Huang et al., 2019; Liang et al., 2020; Conneau et al., 2019; Chi et al., 2021a; Chen et al., 2022) have been proposed rapidly. In general, these xPLMs are always trained on large-scale multilingual corpora using various pre-training language modeling tasks, such as MLM (Devlin et al., 2018; Lan et al., 2020), NSP (Pires et al., 2019), CLISM (Chen et al., 2022), and TRTD (Chi et al., 2021c). In this manner, xPLMs acquire robust contextually relevant representations and, as a result, excel at a variety of downstream tasks, like question answering (Hermann et al., 2015; He et al., 2018; Chen et al., 2021a) and name entity recognition (Liang et al., 2021). For instance, Chen et al. (2022) propose to train xPLMs with CLISM and MLM, achieving remarkable performances in multilingual sequence labeling tasks (Huang et al., 2019; Lewis et al., 2020; Artetxe et al., 2019a).

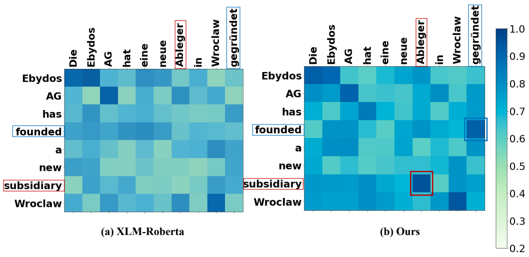

Figure 1: A visualization example from XLM-Roberta and Ours. Here we present the same sentence in English and German. The words in red and blue box refers to the aligned verb and object words, separately.

Although these pre-training tasks help xPLMs learn promising multilingual contextualized representations at hierarchical level (i.e., token or sentence-level) (Li et al., 2022a), they don't take structural knowledge into consideration. One obvious limitation of the above approaches is the semantic misalignment between structural words from different languages, leading to a bias in the understanding of the multilingual representations. We showcase the parallel sentences in English and German in Figure 1 that are quite different in the syntax structure. The main components of this sentence are "Ebydos AG" (subject), "founded" (verb), "subsidiary" (object) and "Wroclaw" (entity). Unfortunately, as one of the current state-of-the-art xPLMs: XLM-Roberta (XLM-R) (Conneau et al.,

[*] Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2023, pages 2042–2057
July 9-14, 2023 ©2023 Association for Computational Linguistics

2019) is incapable of capturing the alignment of these crucial words in German, leading to semantic deviation. Specifically, XLM-R pays less attention to the corresponding words of "founded" and "subsidiary" in German due to the sentence structure barrier between these two languages.

One step further, from the perspective of human behavior, when a language learner reads a sentence in another language, it can help him/her understand this sentence quickly and accurately by pointing out the structural words in a sentence, including subject, verb, object and entities. This effect will be more noticeable when the sentence is lengthy and complex. Similarly, by providing the extra clues of aligned crucial/informative words in the parallel sentence, the model can benefit from a closer gap of cross-lingual representations.

Motivated by the above factors, we design a Structural Contrastive Pretraining (SCP) task to enhance xPLMs' comprehension ability via contrastive learning, bridging the misalignment between structural words in a parallel corpus. Considering the facts that subject, verb, object (S-V-O) are the backbone of a sentence and aligned entities in cross-lingual parallel sentences convey coreference and information short-cuts (Chen et al., 2022), in this work, we consider **S-V-O** and **entities** as the structural words in a sentence, which are all insightful or crucial. Concretely, we divide the parallel corpus into a number of smaller groups. Each sub-group has two versions of the same sentence, one in the source language (high resource) and one in the target language (low resource). Each structural word in the source and target languages is considered as a positive pair.

Due to the nature of contrastive learning, wherein comparisons are made between positive and negative pairs, an increase in the number of negative pairings may potentially improve performances of the resulting model (Chen et al., 2020). Inspired by momentum contrast in computer vision (He et al., 2020), we keep a queue and employ the encoded embeddings from the previous mini-batch to increase the quantity of negative pairs. In this method, momentum contrast employs a pair of fast and slow encoders to encode the source language sentences and target language sentences, separately. And the fast encoder is saved for fine-tuning on down-stream datasets. However, directly applying this approach to cross-lingual pre-training could lead to another problem: As the fast encoder only sees the source language during pre-training, the training becomes insensitive to other target languages. As a consequence, the resulting model may underperform on cross-lingual transfer. To address this issue, we creatively incorporate the original momentum contrast into the cross-lingual setting, naming it Cross-lingual Momentum Contrast (short for CL-MoCo). Specifically, CL-MoCo utilizes two pairs of fast/slow encoders to jointly optimize source-to-target language and target-to-source language representations, further bridging the cross-lingual gap. In light of the fact that almost all down-stream cross-lingual understanding tasks only need one encoder, the two fast encoders share parameters in our pre-training.

Based on the above two proposed strategies for building positive and negative pairs in SCP, our resulting model XLM-SCP can accurately capture the alignment of sentence structures across different languages, improving the performances on cross-lingual understanding tasks. As seen in Figure 1 (b), ours successfully grasp the correspondence between sentence verbs ("founded"-"gegründet") and objects ("subsidiary"-"Ableger") in English and German. We conduct experiments with two different xPLMs encoders on three multilingual tasks to test the effectiveness of our approach: Name Entity Recognition (NER) (Sang, 2002; Pan et al., 2017), Machine Reading Comprehension (MRC) (Lewis et al., 2020; Artetxe et al., 2019b) and Part-of-Speech Tagging (POS) (Zeman et al., 2019). Extensive results show our method can improve the baseline performances across 5 datasets in terms of all evaluated metrics. For example, ours initialize from XLM-R improves the baselines from 61.35% to 63.39% on WikiAnn dataset (Pan et al., 2017).

In general, our contributions can be summarized as follows:

- We observe that misalignment of the informative and crucial structural words occurs in xPLMs, and design a new pre-trained task called SCP to alleviate this problem.

- We propose CL-MoCo via keeping a large queue to increase the amount of negative pairings via momentum updating, which pushes the model toward more nuanced learning in cross-lingual.

- We conduct extensive experiments on different tasks, demonstrating the effectiveness of our approaches.

2 Related Work

Multilingual Pre-trained Language Models To date, transformer-based large-scale PLMs have become the standard in natural language processing and generation (Devlin et al., 2018; Liu et al., 2019; Lan et al., 2020; Sun et al., 2020). Currently, more and more communities are working to bring PLMs into the actual world of various languages (xPLMs), and several efforts have been proposed such as XLM-Roberta (Conneau et al., 2019) (short for XLM-R), info-XLM (Chi et al., 2021a), CLISM (Chen et al., 2022). These works are pre-trained on a large multilingual corpus with token-level or sentence-level pre-training tasks. Despite their promising performances in multiple down-stream tasks, they all don't explicitly consider structural knowledge in the parallel corpus.

Contrastive Learning As a result of its potential to improve upon existing methods for learning effective representations, contrastive learning (Hadsell et al., 2006) has gained popularity in recent years. It works by grouping representations that are semantically close together (*positives*) in an embedding space and then pushing apart others (*negatives*) that are not neighbors. Contrastive learning objective has been particularly successful in different contexts of natural language processing (Gao et al., 2021; Wu et al., 2020). Moreover, several efforts (Chen et al., 2021a, 2022; Gao et al., 2021; Chen et al., 2021b; You et al., 2021; You et al.; Chen et al., 2023b,a) are well-designed for cross-lingual language understanding. For instance, Liang et al. (2022) proposed multi-level contrastive learning towards cross-lingual spoken language understanding. Chen et al. (2022) employed contrastive learning to learn noise-invariant representation from multilingual corpora for downstream tasks. Different from previous works, we utilize contrastive learning to learn the alignments of the structural words (Tang et al., 2023; Li et al., 2022b), leading to a more comprehensive and accurate understanding on the cross-lingual sentence.

Momentum Contrast Recently, several works (Yang et al., 2021; Wu et al., 2022) have explored momentum contrast in natural language understanding tasks, such as sentence representation and passage retrieval. Specifically, Yang et al. (2021) propose xMoCo to learn a dual-encoder for query-passage matching via two pairs of fast/slow encoders. Although we share a similar topic on momentum contrast, our research questions, application areas, and methods differ. xMoco are designed for query-matching tasks while our proposed CL-MoCo is tailored for cross-lingual representation learning. Moreover, Yang et al. (2021) employs two different encoders for query and passage, separately. However, we share parameters of the two fast encoders in our training. At last, we focus on the representation learning of cross-lingual transfer, but they only take monolingual into consideration.

Recent works Recently, several works (Schuster et al., 2019; Pan et al., 2021; Chi et al., 2021b; Ouyang et al., 2021) also focus on word alignment for multilingual tasks. For clarity, we list some key differences: All of them align each token in the parallel corpus in an "all-to-all" fashion, but we only consider structural words like S-V-O via contrastive learning. The motivations are: (1) In our pilot analysis and experiments, we have two different settings in the proposed SCP: a. training the model with only structural words; b. training the model with all tokens in the sentences. Experimentally, we observe that they achieve comparable performances on MRC tasks but the latter achieves slightly worse results on NER tasks. This is due to the fact that aligning some words with no precise meaning like stopwords may have visible side effects on token-level tasks like NER. (2) Futhermore, the latter could result in more computation cost than the current method. (3) From a human perspective, structural words are the backbone of each sentence, and a solid grasp of them is sufficient to strengthen the management of the majority of situations.

3 Methodology

In this section, we first illustrate our proposed Structural Contrastive Pretraining (SCP) in detail. Then we introduce how to incorporate our method with momentum contrast. Due to the fact that our proposed methods are flexible and can be built on top of any xPLMs, we leverage $\mathcal{E}$ to represent a series of pre-trained language models, where $\mathcal{E}$ could be the E_{fast} in Section 3.2. We aim at enhancing $\mathcal{E}$'s ability to capture consistency between parallel structural representations via SCP. The overview of our approach is illustrated in Figure 2.

3.1 Structural Contrastive Pretraining

Definition To bridge the misalignment between structural words from different languages, we for-

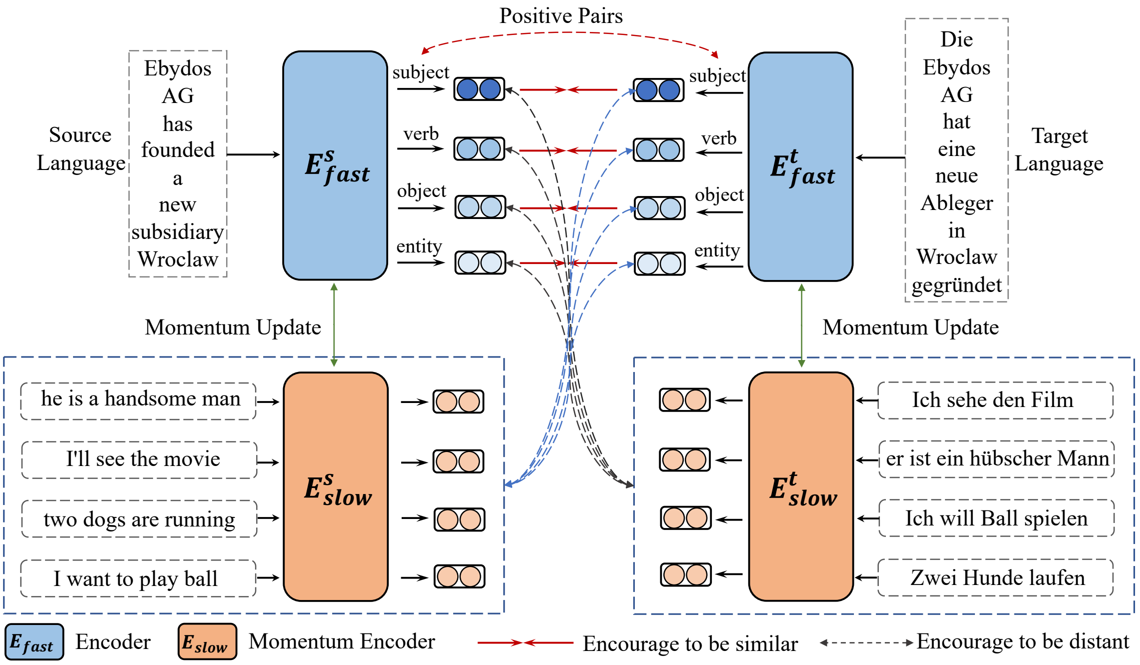

Figure 2: Overview of our proposed method. Two pairs of fast/slow encoders are used in the proposed CL-MoCo. E^s_{fast} and E^s_{slow} encode the sentences in source language. E^t_{fast} and E^t_{slow} are designed for target languages.

mulate a new pre-training task named Structural Contrastive Pretraining (SCP) from the unlabeled data. In this part, we introduce how to collect the structural words in the inputs. Given a source language input sentence $\mathbf{s}^s$ and its target language counterpart $\mathbf{s}^t$, we start by using current online name entity recognition tools (e.g., Spacy) to select structural words in the source language, including the subject, verb, object, and entities in the sentence[1]. As some extracted words are illogical due to the performance limitations of commercially available NER tools, these uninformative words could result in sub-optimizing the model during pre-training. Hence, we follow (Chen et al., 2022) to filter out some uninformative spans:

- Any spans that include solely stop words will be eliminated.
- Selected structural words should not include any punctuation.
- The maximum sequence length of an entity is limited to 6.

As the translation of the same phrase may vary when it is entered independently or combined with a full sentence, we utilize an off-the-shelf alignment tool, GIZA++ (Pei et al., 2020) to align the corresponding ones of the selected structural words in the target language. As a result, we can get structural words $\mathtt{W}^s = \{\mathtt{w}^s_1, \mathtt{w}^s_2, ..., \mathtt{w}^s_k\}$ in $\mathbf{s}^s$ and their counterparts $\mathtt{W}^t = \{\mathtt{w}^t_1, \mathtt{w}^t_2, ..., \mathtt{w}^t_k\}$ in $\mathbf{s}^t$. Notice that the length of k could be more than 4 when there are multiple entities in the sentence.

Pre-training It is essential to obtain the representations of each word from $\mathtt{W}^s$ and $\mathtt{W}^t$ in SCP. Before going further, we first formulate the input sequences as:

$$\mathbf{X}^s = \{\texttt{[CLS]}\mathbf{s}^s\texttt{[SEP]}\} \tag{1}$$

$$\mathbf{X}^t = \{\texttt{[CLS]}\mathbf{s}^t\texttt{[SEP]}\} \tag{2}$$

where [CLS] and [SEP] denote the special beginning and separated tokens. $\mathbf{X}^s$ and $\mathbf{X}^t$ refer to the input sequences in source and target languages, respectively.

Then we can pass $\mathbf{X}^s$ and $\mathbf{X}^t$ into the $\mathcal{E}$, producing contextualized representations of each token in the sequences:

$$\mathcal{H}^s = \mathcal{E}(\mathbf{X}^s) \qquad \mathcal{H}^t = \mathcal{E}(\mathbf{X}^t) \tag{3}$$

where $\mathcal{H}^s \in \mathbf{R}^{l\times d}, \mathcal{H}^t \in \mathbf{R}^{l\times d}$, l and d represent the max sequence length and hidden size, separately. Subsequently, for each word $\mathtt{w}^s_i \in \mathtt{W}^s$, where i $\in$ [1,k], we obtain its representation $\mathcal{H}^s_i$ from $\mathcal{H}^s$. Similarly, we can get its positive pair representation $\mathcal{H}^t_i$ from $\mathcal{H}^t$. Notice that we can not directly

[1] If the extracted words of one sentence are none, we would remove it

employ $\mathcal{H}_i^s$ and $\mathcal{H}_i^t$ in our SCP because $\mathtt{w}_i^s$ and $\mathtt{w}_i^t$ may produce multiple sub-tokens after tokenization. Therefore, we apply extra aggregation function $\mathcal{F}$ on $\mathcal{H}_i^s$ and $\mathcal{H}_i^t$ to obtain the final representations:

$$\mathbf{r}_i^s = \mathcal{F}(\mathcal{H}_i^s) \qquad \mathbf{r}_i^t = \mathcal{F}(\mathcal{H}_i^t) \tag{4}$$

where $\mathcal{F}$ refers to the average pooling of the beginning and ending tokens representations of $\mathcal{H}_i^s$ and $\mathcal{H}_i^t$. $\mathbf{r}_i^s$ and $\mathbf{r}_i^t$ belong to $\mathbf{R}^{1\times d}$. Intuitively, ($\mathbf{r}_i^s$, $\mathbf{r}_i^t$) are regarded as positive pairs in SCP.

3.2 Cross-lingual Momentum Contrast

In this part, we first introduce how to apply momentum contrast on our method in a straight way. Then we illustrate our proposed CL-MoCo.

MoCo As opposed to merely collecting from mini-batch negatives, we use the momentum contrast approach to increase the number of negatives by maintaining a queue of constant size. In particular, the queued embeddings are gradually replaced. When the current mini-batch's sentence embeddings are queued, the "oldest" ones in the queue are eliminated if the queue is full. Intuitively, when directly applying momentum contrast on cross-lingual training, we can employ a pair of encoders E_{fast} and E_{slow}. In one training step, E_{fast} encodes $\mathbf{s}^s$ into $\mathcal{H}^s$ and E_{slow} maps $\mathbf{s}^t$ into $\mathcal{H}^t$. We employs momentum update on the encoder E_{slow}, thereby turning E_{slow} into a sluggish moving-average duplicate of the encoder E_{fast}, to lessen the discrepancy. Formally, we update the E_{slow} in the following way:

$$E_{slow} \longleftarrow \lambda E_{fast} + (1-\lambda)E_{slow} \tag{5}$$

where λ determines how quickly the slow encoder updates parameters and is normally set to a small positive value. After pre-training, only E_{fast} (E_{fast} is equal to $\mathcal{E}$) is saved for fine-tuning and E_{slow} will be discarded.

With the enqueued sentence embeddings, our optimized objective of ($\mathbf{r}_i^s$, $\mathbf{r}_i^t$) is formulated as $\mathcal{L}_i$:

$$-\log \frac{\exp(\Psi(\mathbf{r}_i^s, \mathbf{r}_i^t)/\tau)}{\sum_{j=1}^{N}(\exp(\Psi(\mathbf{r}_i^s, \mathbf{r}_j^t)/\tau) + \sum_{m=1}^{M}\exp(\Psi(\mathbf{r}_i^s, \mathbf{r}_m)/\tau)} \tag{6}$$

where N and M are the size of the mini-batch and the queue, respectively. $\mathbf{r}_m$ denotes a sentence embedding in the momentum-updated queue, and τ represents the temperature. Moreover, Ψ refers to the cosine similarity function.

CL-MoCo In the above method, target language sentences are only encoded by the slow encoder, which is not directly affected by the gradients from the loss. Moreover, the fast encoder only encodes the source languages in pre-training, making it insensitive to the input sequences in other low-resource languages. These two problems could make the encoder sub-optimized and unable to learn reasonable cross-lingual representations. Therefore, we propose CL-MoCo to alleviate the above issues. In particular, CL-MoCo employs two sets of fast/slow encoders: E_{fast}^s and E_{slow}^s for source languages and E_{fast}^t and E_{slow}^t for target languages. In addition, two separate queues Q^s and Q^t are used to store previous encoded sentence embeddings in source and target languages, respectively. The vectors encoded by E_{slow}^s and E_{slow}^t will be pushed into Q^s and Q^t, separately. In CL-MoCo, we jointly optimize the two sets of encoders to learn effective source-to-target language and target-to-source language representations, and Eq.5 can be extended as:

$$E_{slow}^s \longleftarrow \lambda E_{fast}^s + (1-\lambda)E_{slow}^s \tag{7}$$

$$E_{slow}^t \longleftarrow \lambda E_{fast}^t + (1-\lambda)E_{slow}^t \tag{8}$$

Hence, the optimized objective of positive pair ($\mathbf{r}_i^s$, $\mathbf{r}_i^t$) in source-to-target language can be formulated as $\mathcal{L}_i(\mathbf{r}_i^s, \mathbf{r}_i^t)$:

$$-\log \frac{\exp(\Psi(\mathbf{r}_i^s, \mathbf{r}_i^t)/\tau)}{\sum_{j=1}^{N}(\exp(\Psi(\mathbf{r}_i^s, \mathbf{r}_j^t)/\tau) + \sum_{q^s \in Q^s}^{M}\exp(\Psi(\mathbf{r}_i^s, \mathbf{r}_{q^s})/\tau)} \tag{9}$$

Similarly, our CL-MoCo works in both ways, and the objective in target-to-source language $\mathcal{L}_i(\mathbf{r}_i^t, \mathbf{r}_i^s)$ is:

$$-\log \frac{\exp(\Psi(\mathbf{r}_i^t, \mathbf{r}_i^s)/\tau)}{\sum_{j=1}^{N}(\exp(\Psi(\mathbf{r}_i^t, \mathbf{r}_j^s)/\tau) + \sum_{q^t \in Q^t}^{M}\exp(\Psi(\mathbf{r}_i^t, \mathbf{r}_{q^t})/\tau)} \tag{10}$$

For all selected structural words in $\mathbf{s}^s$ and $\mathbf{s}^t$, the overall objective of our SCP can be summarized as:

$$\mathcal{L}_{scp} = \sum_{i=1}^{k}((\mathcal{L}_i(\mathbf{r}_i^s, \mathbf{r}_i^t) + (\mathcal{L}_i(\mathbf{r}_i^t, \mathbf{r}_i^s))/2 \tag{11}$$

where k is the number of structural words in the input sentence. We share the parameters of two fast encoders and two slow encoders, because of the following facts: 1) We focus on cross-lingual understanding tasks rather than passage retrieval, which mostly only needs one encoder; 2) Two separated fast and slow encoders could result in more computation and training time.

<en,es>	<en,ar>	<en,de>	<en,nl>	<en,hi>	Total
1M	0.8M	0.8M	0.7M	0.6M	3.9M

Table 1: Total parallel sentences used in pre-training.

3.3 Pre-training Strategy

Following the line of (Liu et al., 2019; Chi et al., 2021a), we also pre-train $\mathcal{E}$ with the mask language modeling (MLM) task. Concretely, we train the model in multi-task manner. The total objective in our pre-training can be defined as:

$$\mathcal{L} = \mathcal{L}_{scp} + \mathcal{L}_{mlm} \tag{12}$$

4 Experiment

In this section, we first introduce how we collect the pre-training data for the proposed SCP. Then we illustrate experiment settings for pre-training and fine-tuning. At last, we present our experimental results on various corss-lingual datasets, including baseline introduction and main results.

4.1 Pre-training Data

As aforementioned, our proposed task SCP requires parallel corpus. We choose MT dataset (Conneau and Lample, 2019) to construct our pre-training data. In contrast to earlier research (Chi et al., 2021a) that used billion-level corpora across about one hundred languages to generate training corpus, we only use six languages from the MT dataset, including English(en), Spanish(es), Arabic(ar), German(de), Holland(nl), and Vietamese(vi), demonstrating that our approach also makes significant gains in languages where we do not have data. Given the promising performance of off-the-shelf NER techniques (e.g., Spacy) in English, we choose English as our source language, with the remaining five languages serving as target languages in turn. As a result, we get 3.9 million pre-training parallel sentences after using the rules in Section 3.1. The amount of distribution for each language is reported in Table 1.

4.2 Evaluation

We evaluate XLM-SCP on three cross-lingual tasks: cross-lingual machine reading comprehension (xMRC), cross-lingual name entity recognition (xNER) and cross-lingual Part-of-Speech (xPOS). Concretely, we conduct experiments on five datasets: MLQA (Lewis et al., 2020), XQUAD (Artetxe et al., 2019b), CoNLL (Sang, 2002) and WikiAnn (Pan et al., 2017) and UPDOS (Zeman et al., 2019). We introduce each dataset and test languages in Appendix A.1.

We use a *zero-shot* configuration to fine-tune our model for all datasets, which means that we just use the English training set to optimize the model, and then test the final model on other target languages. Besides, we also test the *cross-lingual* transfer ability of XLM-SCP on these datasets, that is, we also validate the model performances on some target languages that are not included in our pre-training data.

We employ two evaluation measures for the xMRC task: Exact Match (EM) and span-level F1 score, which are commonly used for MRC model accuracy evaluation. The span overlap between the ground-truth answer and the model predictions is measured by span-level F1. If the forecast is precisely the same as the ground truth, the exact match (EM) score is 1, otherwise 0. In the case of the xNER challenge, we employ entity-level F1 scores to evaluate our model, which demands that the boundary and type between the prediction and the ground-truth entity be exactly matched. Similarly, we also use F1 score to validate the model performances in UPDOS.

4.3 Training Details

Model Structure To show the generalization of our approach, we initialize our model from two commonly used xPLMs encoders: XLM-R and Info-XLM. The resulting model is named **XLM-SCP** in our experiments. We use the base version checkpoints of the above two models from Hugging Face Transformers[2]. Our XLM-SCP contains 12 transformer layers, and the vector dimension size is set to 768.

Pre-training Details Our training codes are based on PyTorch 1.11 and Transformers 4.10.0. Along the line of the research (Devlin et al., 2018), we randomly mask 15% tokens of the input sequence[3] to implement MLM. In pre-training, we optimize our model using the Adam optimizer and a batch size of 128 for a total of 4 epochs. Moreover, learning rate is set to 1e-6 with 1.5K warmup steps. The max input sequence length is set to 128. Experimentally, τ in Eq.10 is set to 0.05 and the queue size of Q^s and Q^t are both 20k. And λ is

[2] https://github.com/huggingface/transformers

[3] The structural words in SCP will not be masked to avoid missing labels.

Model	xMRC		xNER		xPOS	Average
	MLQA	XQUAD	CoNLL	WikiAnn	UPDOS	
M-BERT	57.80/42.40	69.63/53.72	78.20	62.21	70.31	67.63
XLM	61.70/44.20	70.93/53.18	79.00	61.22	70.12	68.58
XLM-R	63.24/45.88	73.54/57.55	78.48	61.35	74.21	70.16
XLM-SCP*	**65.14/47.20**	**75.35/59.20**	**80.35**	**63.39**	**75.20**	**71.89**
Info-XLM	65.25/47.63	75.79/59.50	79.52	63.01	74.71	71.66
XLM-SCP$^{\heartsuit}$	**67.01/48.90**	**76.93/60.75**	**80.94**	**64.77**	**75.60**	**73.05**

Table 2: Average evaluation results on five datasets. The results of our model are averaged over 5 runs. * denotes the model build upon of XLM-R. $^{\heartsuit}$ refers to model based on Info-XLM. The results of each language are represented in the Appendix B. We highlight the highest numbers among models with the same xPLM encoder. Here, we average the F1 scores on these datasets.

set to 0.99. We pre-train our model using 8×V100-32G GPUs for about one day. Fine-tuning details can be seen in Appendix A.2.

4.4 Results

Baselines We compare our model with the following xPLM-based baselines: (1) M-BERT (Devlin et al., 2018) pre-trained with MLM and NSP tasks on Wikipedia data over 104 languages; (2) XLM (Conneau and Lample, 2019) is jointly optimized with MLM and TLM tasks in 100 languages during pre-training; (3) XLM-R (Conneau et al., 2019), a multilingual version of Roberta which is pre-trained with MLM in large-scale CC-100 dataset; (4) Info-XLM (Chi et al., 2021a), another popular and effective xPLM which initializes from XLM-R with the proposed pre-training task XLCO in 94 languages.

xMRC Results Table 2 compares our method to that of typical systems on five datasets. On two xMRC datasets, our models outperform these baselines by an interesting amount. For instance, ours built on XLM-R achieves 65.14%/47.20% (vs. 63.24%/45.88%) in terms of F1/EM score on MLQA. Similarly, we also obtain 1.81%/1.65% gains on XQUAD dataset. We can also draw another interesting conclusion: When compared to Info-XLM, which is both built on top of XLM-R and continues to be pre-trained on 130 million data across 94 languages, our model initialized from XLM-R performs comparably. Nevertheless, XLM-SCP only needs 3.9 million parallel corpora from six languages, demonstrating the efficacy of our proposed approaches (3.9M≪130M).

Model	WikiAnn	XQUAD	MLQA
XLM-R	60.41	73.24/57.01	64.89/44.99
XLM-SCP	**61.91**	**74.56/58.50**	**66.24/46.57**

Table 3: Model performances under zero-shot cross-lingual transfer. In the experiments, We initialize XLM-SCP from XLM-R.

xNER Results As shown in Table 2, when compared with XLM-R, our XLM-SCP yields 1.87%/2.04% F1 score improvements on the CoNLL and WikiAnn datasets, separately. Importantly, when compared to Info-XLM on top of XLM-R, ours still outperform on xNER tasks. In other words, our approach has demonstrated its full potential using less than 4% of the corpus. Moreover, XLM-SCP initialized from Info-XLM also outperforms on these two datasets: 80.92% (vs. 79.52%) and 64.69% (vs. 63.01%).

xPOS Results We further test our model on xPOS tasks across 37 languages. Results from Table 2 show our model also obtains consistent gains of about 1% score on UPDOS dataset. Using Info-XLM as the basic encoder, ours can achieve the best results 75.60%. Overall, our experimental results on three tasks demonstrate the efficacy and generalizability of our proposed approach.

Zero-shot Cross-lingual Transfer Results We further test out the method under the setting of zero-shot cross-lingual transfer in other unseen targeted languages in pre-training such as Arabic (ar), Afrikaans (af). Concretely, we conduct experiments to validate the resulting model's performances on the selected test sets in other languages from WikiAnn, XQUAD and MLQA that are not

Algorithms	WikiAnn	XQUAD
XLM-SCP	**63.39**	**75.35/59.20**
w/o SCP	62.11	74.02/58.01
w/o CL-MoCo	62.65	74.50/58.46
w/o MLM	62.58	74.44/58.11

Table 4: Ablation study of pre-training schemes on WikiAnn and XQUAD datasets. In the experiments, We initialize XLM-SCP from XLM-R.

included during pre-training. From Table 3, we can observe that XLM-SCP also achieves about 1.5% improvements on three datasets under the zero-shot cross-lingual transfer setting. In general, the results in Table 2 and Table 3 prove that our approach not only improves the performance in the languages that included in our SCP pre-training but also has better transferability capabilities in other low-resource languages.

5 Analysis

Aside from the high performances achieved by our proposed approaches, we are still concerned about the following questions: $\mathbf{Q_1}$: What are the effects of each key component in our XLM-SCP? $\mathbf{Q_2}$: Is CL-MoCo really superior to MoCo in cross-lingual understanding tasks? $\mathbf{Q_3}$: Does the size of the queue in CL-MoCo affect the performance of our model? $\mathbf{Q_4}$: What are the model performances with different τ in Eq.10? (Seen in Appendix C, Figure 5) $\mathbf{Q_5}$: Within the chosen objects, verbs, objects, and entities in structural words, which part has the biggest effect on our XLM-SCP's performance? (Seen in Appendix C, Table 10) In this section, we conduct extensive experiments to answer the above questions.

Answer to $\mathbf{Q_1}$: Experiments are carried out to confirm the independent contributions of each component in our proposed pre-training scheme. Table 4 shows the model performances by removing each key component on WikiAnn and XQUAD. From the table, we can see that SCP plays the most important role in our architecture. Removing SCP decreases the model performances from 63.39% to 61.35% on WikiAnn. Meanwhile, we can see that our pre-training system as a whole is effective since each part, including MLM and CL-MoCo, helps the model perform better. Notice that removing CL-MoCo means we only construct negative pairs from in-batch negatives.

Answer to $\mathbf{Q_2}$: We further conduct analysis to verify the effectiveness of **CL-MoCo vs. MoCo** on cross-lingual understanding tasks. We conduct ablation experiments on three tasks across four datasets and show the results in Figure 3. We can find that our proposed CL-MoCo can achieve better results on all these datasets when compared with the original MoCo. The results further prove CL-MoCo has a stronger ability to learn effective cross-lingual representations.

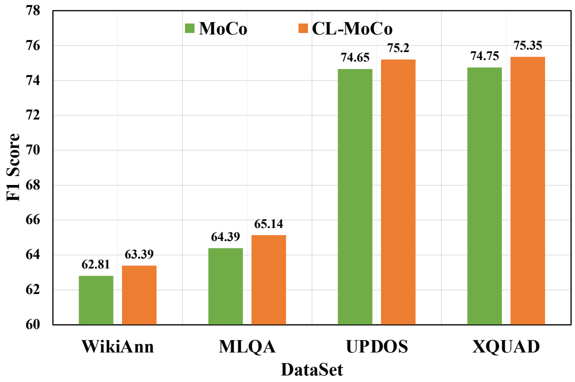

Figure 3: **CL-MoCo vs. MoCo** across four datasets, and F1 score is used for evaluation. In the experiments, We initialize XLM-SCP from XLM-R. The results are averaged of five runs.

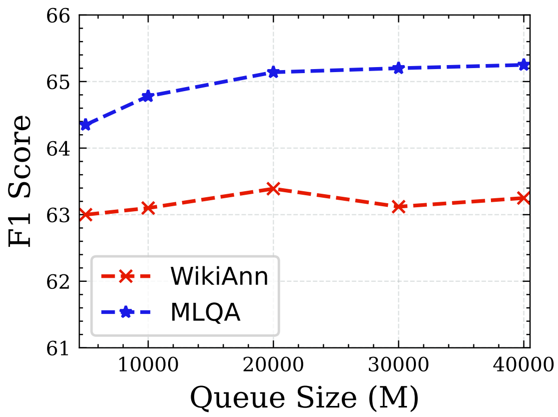

Figure 4: Queue size sensitivity experiments across two datasets, and F1 score is used for evaluation. In the experiments, We initialize XLM-SCP from XLM-R.

Answer to $\mathbf{Q_3}$: The main assumption of CL-MoCo is that the size of negative samples is important in contrastive learning. Here we empirically study this assumption in cross-lingual understanding tasks via varying the queue size of keeping negative pairs. As shown in Figure 4, we validate XLM-SCP with M $\in \{5k, 10k, 20k, 30k, 40k\}$ on WikiAnn and MLQA datasets. We can draw the conclusion that the model performs slightly better as the queue size increases initially, especially for xMRC tasks. Interestingly, the model achieves best results on WikiAnn when M is equal to 20k, and its performances slightly decrease when M passes 20k. One possible explanation is that larger size of the queue may introduce some "false negative samples", which could have a more obvious side effect on xNER tasks. In light of the fact that the queue size has a negligible effect on training speed and memory use, we have chosen a queue size of 20k for all downstream datasets.

6 Conclusion

In this paper, we observe that misalignment of crucial structural words occurs in the parallel sentences of current xPLMs. We propose a new pretraining task called Structural Contrastive Pretraining (SCP) to alleviate this problem, enabling the model to comprehend the cross-lingual representations more accurately. We further incorporate momentum contrast into cross-lingual pre-training, named CL-MoCo. In particular, CL-MoCo employs two sets of fast/slow encoders to jointly learn the source-to-target language and target-to-source language cross-lingual representations. Because of this, the resulting model is better for cross-lingual transfer. Extensive experiments and analysis across various datasets show the effectiveness and generalizability of our approach. As an extension of our future work, we will apply our method to other natural language understanding tasks and find a proper way to reduce data preprocessing costs.

Limitations

The main target of this paper is to utilize structural knowledge for cross-lingual comprehension. We present a new pre-training task named SCP in the hope of bridging the misalignment of structural words in the parallel corpus. More generally, we expect the proposed method can facilitate the research of cross-lingual understanding. Admittedly, the main limitation of this work is that we rely on off-the-shelf tools to extract and align words in different languages, which would result in some mistakes at some situations. For example, GIZA++ only achieves 80%-85% accuracy in aligning the corresponding words in another language. Currently, no tech can achieve this goal in 100% accuracy. As a result, some bias data in pre-training calls for further research and consideration when utilizing this work to build xPLMs.

Acknowledgement

This research was supported by NSFC Grant No. 62206067, and Guangzhou-HKUST(GZ) Joint Funding Scheme 2023A03J0673.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019a. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019b. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Nuo Chen, Linjun Shou, Min Gong, Jian Pei, and Daxin Jiang. 2021a. From good to best: Two-stage training for cross-lingual machine reading comprehension.

Nuo Chen, Linjun Shou, Ming Gong, Jian Pei, Bowen Cao, Jianhui Chang, Daxin Jiang, and Jia Li. 2023a. Alleviating over-smoothing for unsupervised sentence representation. *arXiv preprint arXiv:2305.06154*.

Nuo Chen, Linjun Shou, Ming Gong, Jian Pei, and Daxin Jiang. 2022. Bridging the gap between language models and cross-lingual sequence labeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1909–1923, Seattle, United States. Association for Computational Linguistics.

Nuo Chen, Linjun Shou, Ming Gong, Jian Pei, Chenyu You, Jianhui Chang, Daxin Jiang, and Jia Li. 2023b. Bridge the gap between language models and tabular understanding. *arXiv preprint arXiv:2302.09302*.

Nuo Chen, Chenyu You, and Yuexian Zou. 2021b. Self-supervised dialogue learning for spoken conversational question answering. *CoRR*, abs/2106.02182.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021a. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. In *NAACL-HLT*, pages 3576–3588. Association for Computational Linguistics.

Zewen Chi, Li Dong, Bo Zheng, Shaohan Huang, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2021b. Improving pretrained cross-lingual language models via self-labeled word alignment. 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 3418–3430, Online. Association for Computational Linguistics.

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Saksham Singhal, Payal Bajaj, Xia Song, and Furu Wei. 2021c. XLM-E: cross-lingual language model pre-training via ELECTRA. *CoRR*, abs/2106.16138.

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

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *NeurIPS*, pages 7057–7067.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *CoRR*, abs/2104.08821.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR (2)*, pages 1735–1742. IEEE Computer Society.

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.

Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. 2018. Dureader: a chinese machine reading comprehension dataset from real-world applications. In *QA@ACL*, pages 37–46. Association for Computational Linguistics.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701.

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 *EMNLP/IJCNLP (1)*, pages 2485–2494. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *ICLR*. OpenReview.net.

Patrick S. H. Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: evaluating cross-lingual extractive question answering. In *ACL*, pages 7315–7330. Association for Computational Linguistics.

Jia Li, Yongfeng Huang, Heng Chang, and Yu Rong. 2022a. Semi-supervised hierarchical graph classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jiajin Li, Jianheng Tang, Lemin Kong, Huikang Liu, Jia Li, Anthony Man-Cho So, and Jose Blanchet. 2022b. Fast and provably convergent algorithms for gromov-wasserstein in graph learning. *arXiv preprint arXiv:2205.08115*.

Shining Liang, Linjun Shou, Jian Pei, Ming Gong, Wanli Zuo, and Daxin Jiang. 2021. Calibrenet: Calibration networks for multilingual sequence labeling. In *WSDM*, pages 842–850. ACM.

Shining Liang, Linjun Shou, Jian Pei, Ming Gong, Wanli Zuo, Xianglin Zuo, and Daxin Jiang. 2022. Multi-level contrastive learning for cross-lingual spoken language understanding. *CoRR*, abs/2205.03656.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *EMNLP (1)*, pages 6008–6018. Association for Computational Linguistics.

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. *CoRR*, abs/1907.11692.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lin Pan, Chung-Wei Hang, Haode Qi, Abhishek Shah, Saloni Potdar, and Mo Yu. 2021. Multilingual BERT post-pretraining alignment. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 210–219, Online. Association for Computational Linguistics.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *ACL*

(1), pages 1946–1958. Association for Computational Linguistics.

Shichao Pei, Lu Yu, Guoxian Yu, and Xiangliang Zhang. 2020. REA: robust cross-lingual entity alignment between knowledge graphs. In *KDD*, pages 2175–2184. ACM.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

Erik F. Tjong Kim Sang. 2002. Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *CoNLL*. ACL.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. 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 1599–1613, Minneapolis, Minnesota. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A continual pre-training framework for language understanding. In *AAAI*, pages 8968–8975. AAAI Press.

Jianheng Tang, Weiqi Zhang, Jiajin Li, Kangfei Zhao, Fugee Tsung, and Jia Li. 2023. Robust attributed graph alignment via joint structure learning and optimal transport. *ICDE*.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022. ESimCSE: Enhanced sample building method for contrastive learning of unsupervised sentence embedding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3898–3907, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020. CLEAR: contrastive learning for sentence representation. *CoRR*, abs/2012.15466.

Nan Yang, Furu Wei, Binxing Jiao, Daxing Jiang, and Linjun Yang. 2021. xMoCo: Cross momentum contrastive learning for open-domain question answering. 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 6120–6129, Online. Association for Computational Linguistics.

Chenyu You, Nuo Chen, and Yuexian Zou. Mrd-net: Multi-modal residual knowledge distillation for spoken question answering.

Chenyu You, Nuo Chen, and Yuexian Zou. 2021. Self-supervised contrastive cross-modality representation learning for spoken question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 28–39.

Daniel Zeman, Joakim Nivre, and Mitchell Abrams. 2019. Universal dependencies 2.5. lindat/clariah-cz digital library at the institute of formal and appliedlinguistics (ufal). *Faculty of Mathematics and Physics. Charles University*.

A Training details

A.1 Fine-tuning Dataset

Cross-Lingual Machine Reading Comprehension MLQA and XQUAD are two popular xMRC benchmarks, which share the same training set from SQUA and consists of different test sets in low-resource languages. In this work, we evaluate our methods on six languages: including *English, Arabic, German, Spanish, Hindi, Vietnamese*.

Cross-lingual Name Entity Recognition CoNLL and WikiAnn are commonly-used xNER benchmarks. We evaluate CoNLL on four language test sets: *Spanish, Dutch, English, German*. As for the WikiAnn challenge, we evaluate the model with 48 languages.

Cross-lingual Part-of-Speech Tagging UPDOS is a typical dataset of POS in multilingual. Of note, UPDOS contains 37 languages, which all of them are used to test our model performances.

A.2 Fine-tuning details

We use the official codes from Hugging Face Examples[4] to fine-tune and test our models. The detailed hyper-parameter setups are presented in Table 5.

B Main Results

In this section, we present the model's performances on each language across five datasets.

xMRC Results Table 6 and Table 7 show the model performances on MLQA and XQUAD datasets.

xNER Results Table 8 shows the model performances on WikiAnn dataset.

xMRC Results Table 9 represents the model performances on UDPOS dataset.

C Analysis

Answer to Q_4: Intuitively, it is essential to study the sensitivity analysis of the temperature τ in our SCP. Thereafter, we further conduct experiments to verify the impact of different τ on our model performances. We test out our XLM-SCP with $\tau \in \{0.01, 0.05, 0.1, 0.5\}$ on XQUAD, MLQA and WikiAnn datasets. From the Figure 5, we can observe that changing τ could cause the model to improve and decrease. Concretely, ours achieve best results when $\tau = 0.05$.

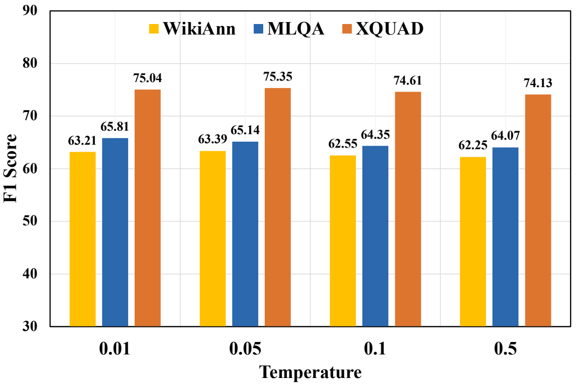

Figure 5: Temperature sensitivity experiments across three datasets, and F1 score is used for evaluation. In the experiments, We initialize XLM-SCP from XLM-R.

Answer to Q_5: We further conduct analysis to find that which part of the chosen nouns, verbs, objects, and entities in structural words has the most impact on how well our model works? Hence, we remove each S-V-O and entity word in turn and test out the model's performances on xNER tasks and xPOS tasks. As the Table 10 shows, each component in the selected structural word has different impact on our XLM-SCP. Interestingly, the model's performance drops significantly on the WikiAnn dataset without entity while very somewhat on the UDPOS dataset without entity. The possible reason is that xNER tasks require the model has a stronger ability of entity-level understanding while xPOS tasks need more fine-grained understanding on token-level.

[4] https://github.com/huggingface/transformers/examples

Parameter	MLQA	XQUAD	WikiAnn	CoNLL	UPDOS
Batch size	32	32	32	16	16
Learning Rate	3e^{-5}	3e^{-5}	2e^{-5}	2e^{-5}	2e^{-5}
Epoch	5	5	5	5	5
Warm Up	10%	10%	10%	10%	10%
Max Length	384	384	128	128	128

Table 5: Hyper-parameters setup during fine-tuning.

Models	en	ar	de	vi	hi	es	Avg.
Ours(XLM-R)	79.74/65.93	53.80/35.12	61.59/45.65	67.98/47.00	60.97/42.11	66.35/45.01	65.14/47.20
Ours(Info-XLM)	80.84/67.95	53.84/35.35	60.90/45.14	66.57/46.70	60.86/44.48	66.70/45.88	67.01/48.90

Table 6: The performance of our models on MLQA datasets.

Models	en	es	de	ar	hi	vi	Avg.
Ours(XLM-R)	78.82/63.91	74.63/60.41	74.34/59.92	67.57/49.23	68.11/50.67	72.72/50.82	75.35/59.20
Ours(Info-XLM)	79.65/67.30	76.12/60.05	73.21/60.89	70.31/52.98	69.10/51.33	72.42/50.34	76.93/60.75

Table 7: The performance of our models on XQUAD datasets.

Model	ar	he	vi	id	jv	ms	tl	eu	ml	ta	te	af	nl	en	de	el	bn	hi	mr	ur	fa	fr	it	pt
Ours	54.8	52.7	67.6	47.6	60.4	68.0	69.0	61.3	61.6	54.3	47.3	76.3	80.4	82.4	74.2	74.7	69.5	68.0	62.9	62.0	53.7	77.4	77.8	79.2

es	bg	ru	ja	ka	ko	th	sw	yo	my	zh	kk	tr	et	fi	hu	qu	pl	uk	az	It	pa	gu	ro	Avg.
75.1	77.7	62.4	19.4	66.6	48.7	2.2	66.2	48.7	56.5	69.1	40.6	75.0	71.2	75.6	77.8	59.2	78.2	77.6	62.9	72.4	52.3	57.8	76.3	62.8

Table 8: Results on WikiAnn named entity recognition.

Model	af	ar	bg	de	el	en	es	et	eu	fa	fi	fr	he	hi	hu	id	it	ja	kk
XLM-SCP	88.0	68.5	89.6	88.8	86.5	95.8	88.8	86.3	67.7	69.6	85.8	87.5	67.9	68.7	82.7	72.6	89.5	28.9	76.0

Model	ko	mr	nl	pt	ru	ta	te	th	tl	tr	ur	vi	yo	zh	It	pl	uk	ro	Avg.
XLM-SCP	52.3	81.6	89.3	88.2	89.5	62.3	83.2	48.0	89.2	74.3	60.3	58.2	25.4	39.6	84.4	85.4	85.4	84.8	75.20

Table 9: Results on part-of-speech tagging.

Algorithms	WikiAnn	UPDOS
XLM-SCP	**63.39**	**75.20**
w/o subject	63.12	74.72
w/o verb	63.01	74.84
w/o object	63.08	74.82
w/o entity	62.88	75.01

Table 10: Ablation study of structural words on WikiAnn and UDPOS datasets. In the experiments, We initialize XLM-SCP from XLM-R.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section limitations

☑ A2. Did you discuss any potential risks of your work?
section limitations

☑ A3. Do the abstract and introduction summarize the paper's main claims?
abstract and section i1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section Experiments

☑ B1. Did you cite the creators of artifacts you used?
Section Experiments

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section Experiments

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section Experiments

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section Experiments

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section Experiments

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section Experiments

C ☑ Did you run computational experiments?

Section Experiments

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section Experiments

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section Experiments

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section Experiments

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section Experiments

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.