Wuhan University
Journal of Natural Sciences

2023, Vol.28 No.4, 299-308

Article ID 1007-1202(2023)04-0299-10
DOI https://doi.org/10.1051/wujns/2023284299

EPT: Data Augmentation with Embedded Prompt Tuning for Low-Resource Named Entity Recognition

YU Hongfei, NI Kunyu, XU Rongkang, YU Wenjun, HUANG Yu†
College of Informatics, Huazhong Agricultural University, Wuhan 430070, Hubei, China

Abstract: Data augmentation methods are often used to address data scarcity in natural language processing (NLP). However, token-label misalignment, which refers to situations where tokens are matched with incorrect entity labels in the augmented sentences, hinders the data augmentation methods from achieving high scores in token-level tasks like named entity recognition (NER). In this paper, we propose embedded prompt tuning (EPT) as a novel data augmentation approach to low-resource NER. To address the problem of token-label misalignment, we implicitly embed NER labels as prompt into the hidden layer of pre-trained language model, and therefore entity tokens masked can be predicted by the finetuned EPT. Hence, EPT can generate high-quality and high-diverse data with various entities, which improves performance of NER. As datasets of cross-domain NER are available, we also explore NER domain adaption with EPT. The experimental results show that EPT achieves substantial improvement over the baseline methods on low-resource NER tasks.
Key words: data augmentation; token-label misalignment; named entity recognition; pre-trained language model; prompt
CLC number: TP183

0 Introduction

Named entity recognition (NER) is a fundamental natural language process (NLP) task that involves labeling named entities with predefined categories. Its accuracy has a significant impact on the performance of downstream tasks, such as information extraction[1], question answering[2], text summarization[3] and machine translation[4] etc. However, the high cost of obtaining labeled data has limited the amount of available data for most languages and domains. To address this issue, low-resource NER methods[5-8] have been proposed in recent years. Data augmentation is an effective method for expanding the training set by generating new data with preserved labels, which is especially useful in low-resource scenarios. Common methods of data augmentation for NLP include word-level modification[9-11] and back-translation[12,13].

While word-level modification and back-translation have shown satisfactory performance in sentence-level tasks, they can fail in token-level tasks such as NER due to token-label misalignment. For example, word-level modification may replace an entity with alternatives that are coherent in the sentence but entirely mismatched with the original label, such as synonym replacement[10], auto-regressive Pretrained Language Model (PLM)[11], Long Short Term Memory- Language Model (LSTM-LM)[14], and Masked Language Model (MLM)[15]. Simi-

Received date: 2023-03-10
Biography: YU Hongfei, male, Master candidate, research direction: reinforcement learning, natural language process. E-mail: hfyu@webmail.hzau.edu.cn
† To whom correspondence should be addressed. E-mail: yhuang@mail.hzau.edu.cn

larly, back-translation may rely on external word alignment tools[16,17], leading to fallibility. Some attempts have been made to address token-label misalignment, but they have not been entirely successful. One approach[7] randomly interchanges entities of the same category, which does not contribute to entity diversity and may damage the coherence of the context. Another attempt[18] only modifies non-entity context tokens with the Masked Sequence to Sequence (MASS)[19] method, keeping the entire entities unchanged. However, this method did not achieve notable improvement in pretrained LM-based NER tasks[20]. Building on prior research, Masked Entity Language Model (MELM)[21] attempts to generate augmented data through finetuning a language model on linearized labeled sequences[22,23], which injects NER labels into the sentence context and finetunes the model to predict masked entities. This approach achieved remarkable performance in NER evaluations.

Inspired by previous work, we introduce Embedded Prompt Tuning (EPT) as a novel approach to low-resource NER, which embeds NER labels as prompts to generate data with more diverse entities that perfectly align with their labels. Similar to MELM, EPT is based on a pretrained masked language model, which is then finetuned to augment data by randomly masking entity tokens in a training set. Besides, we add a few embedding layers to MLM for prompt embedding, which is a slight departure from the standard MLM approach. We note that it is insufficient to simply mask and replace entity tokens using the finetuned MLM, as has been observed in previous work[21]. Figure 1 demonstrates that the finetuned MLM can make predictions with token-label misalignment, such as "*Washington has*". In contrast, MELM can alleviate the misalignment and give more accurate predictions. However, the labels injected into the sequence can limit the number of original tokens, especially when there are multiple consecutive entities in the sentence, which can negatively impact the coherence of the context and the diversity of the entities. To address this issue, EPT introduces additional embedding layers that separately embed the label and the position of the token as inputs to the hidden layer of MLM. Therefore, the masked tokens are predicted based on both the context and the prompt of the tokens' labels and positions.

Our EPT approach has the potential to expand the scope of entity recognition across different domains with specialized entity categories. We applied EPT to the Cross-NER dataset[24], which comprises fully-labeled NER data from five domains with specialized and shared entity types. Most existing NER methods focus on a specific domain, resulting in suboptimal performance in domain adaptation[25]. By contrast, EPT's ability to extend entity categories allows us to leverage knowledge from other domains to increase entity diversity, even with limited labeled data. We finetuned EPT on the source domain and then on the training sentences of the target domain for data augmentation. Our results show that finetuning over the source domain leads to higher $F1$-scores.

Overall, this paper makes several contributions:

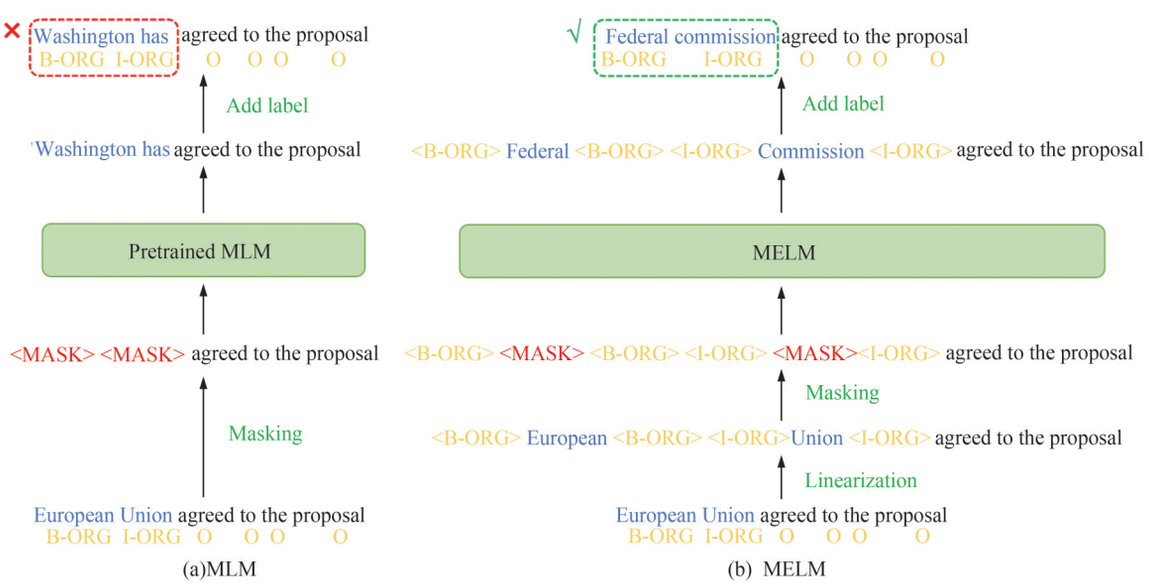


Fig.1 Comparison of different data augmentation methods

First, we propose a novel approach, EPT, to augment data for low-resource scenarios, which results in significant improvements in both monolingual and cross-domain NER. Second, we introduce implicit label embedding and novel position embedding, which effectively increases entity diversity while mitigating token-label misalignment. Third, we demonstrate the effectiveness of using training sentences from source domains to improve data augmentation in the domain adaptation area. These contributions provide new insights into the field of NER and offer potential solutions to common challenges faced in low-resource scenarios and cross-domain NER.

1 Method

The process of our data augmentation method is illustrated in Fig.2. We embed labels and positions of entity tokens into the hidden layers of our model (as described in Section 1.1). Then, the EPT model is fine-tuned (as outlined in Section 1.2) to generate augmented data from the NER training sentences that contain masked entities (as explained in Section 1.3). The generated data is used for training the NER model along with the original data after several loops of augmentation (as detailed in Section 1.4). Algorithm 1 shows the detailed process and steps of our method.

1.1 Embedding Prompt

1.1.1 Label embedding

To address the issue of token-label misalignment, we introduce additional embedding layers to the MLM model, which embeds the labels and positions of entity tokens into hidden vectors. As shown in Fig.2, these embeddings are added to the token embeddings with a coefficient α during the finetuning process. By doing so, label prompt is taken into consideration during the prediction of a masked token. Additionally, we initialize the weights of the embedding layers with special words, such as "location" for "B-LOC" and "I-LOC". Meanwhile the weights would also be initialized with normal words for a test, which means "B-label" and "I-label" are assigned random values.

1.1.2 Position embedding

The Relative Position Embedding (RPE) is designed to mark the positions of tokens within an entity word. While MELM does not require Position Embedding for the label tokens before and after each entity token, our EPT uses implicit label prompt, which can lead to consecutive entities with the same labels (e. g., multiple "I-ORG" labels in succession, as shown in Fig. 2), making it difficult for the model to predict masked tokens accurately. To address this issue, we introduce a trainable RPE that encodes position information for each sub-word of the entity. Since the number of sub-words in an entity can be various, our RPE is designed to be trainable and flexible as following, enable to handle entities of different lengths:

$$\forall x \in X = \{(s+1)*(l-1) \mid 0 \leqslant s < l \leqslant L,\ s, l \in \mathbb{N}\},\ P_x \in \mathbb{R}^D \tag{1}$$

where position $x \in \mathbb{N}$ is presented as pair $(s,\ l-1-s)$

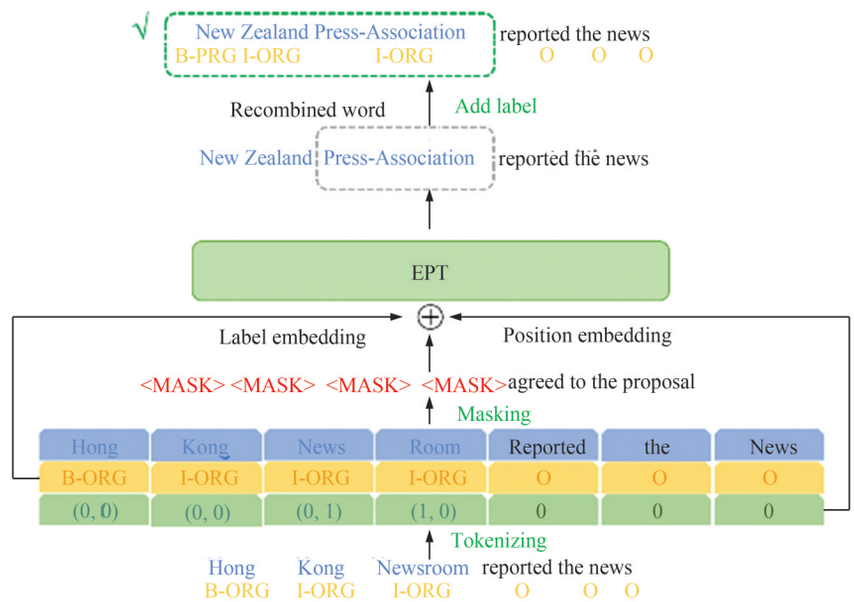


Fig.2 Embedded prompt tuning model

and X presents the whole position set; P_x is the x-th relative position vector; s is the relative position to the start of the entity; l is the length of the sub-words, and L is the maximum length of all the sub-words of entities and D is the number of the attention heads of EPT. Finally, to reduce the parameters to be trained, we broadcast the position vector multiplied by β to H dimension, which is the depth of the hidden layers. The relative position information of each token will be utilized to predict masked entity tokens.

1.2 Fine-Tuning EPT

Due to the invalidity of masking context mentioned in the introduction, there are only entity tokens to be masked during finetuning EPT. During the previous tokenization and embedding, the entity tokens have been randomly masked, and each sentence has several masked versions which are picked at the beginning of each epoch for diversity. The masking ratio of the tokens is identical to MELM's. EPT is trained to maximize the probability of the original sentence X given the masked sentence $\tilde{X}$ with prompt:

$$\max_{\theta} \log p_{\theta}\left(X|\tilde{X}+E_{1}+E_{\mathrm{p}}\right) \approx \sum_{i=1}^{n} m_{i} \log p_{\theta}\left(x_{i}|\tilde{X}+E_{1}+E_{\mathrm{p}}\right) \tag{2}$$

where E_1, E_p are embedding of labels and positions. θ represents the parameter of EPT and p_θ means the probability of original sentence given masked sentence under for EPT model with parameter θ. x_i is the i-th entity token while m_i is a Boolean value indicating whether x_i is masked or not, and upper bound of i is the account of entity tokens n. The prompt of labels and positions makes the prediction consistent with the original token in many spheres.

1.3 Data Augmentation

Generating novel augmented sentences poses a diversity problem of generating samples due to the potential repetition of data when the most probable token is selected directly. EPT employs the same augmentation method as MELM, which involves masking tokens with a rate drawn from a Gaussian distribution and randomly selecting replacements from the top k predicted items. As context masking has marginal effects in NER tasks[20], we focus only on masking entity tokens in sentences. This process then repeats R times for each sentence in the original dataset, resulting in R masking results. Importantly, EPT can generate more unique entities with semantic rationality, due to the RPE which prevents misplaced token recombination. By contrast, MELM tends to suffer from semantic fragmentation due to explicit label tokens and produces fewer novel entities.

1.4 Training NER Model

To check out the quality of the generated data, we do not manually filter the generated data. The augmented training set $\mathbb{D}_{\mathrm{aug}}$ is combined with the training set $\mathbb{D}_{\mathrm{train}}$ as the final training set for the NER task.

Algorithm 1 shows the detailed process and steps of our method.

Algorithm 1 Embedded Prompt Tuning (EPT)

Given $\mathbb{D}_{\mathrm{train}}$, $\mathcal{M}$ //Given gold training set $\mathbb{D}_{\mathrm{train}}$, and pretrained MLM $\mathcal{M}$

$\tilde{\mathcal{M}} \leftarrow \mathcal{M}+\alpha\,\mathcal{E}_{\mathrm{label}}+\beta\mathcal{E}_{\mathrm{pos}}$ //Add label embedding layer $\mathcal{E}_{\mathrm{label}}$ and position embedding layer $\mathcal{E}_{\mathrm{pos}}$ to $\mathcal{M}$, α and β are coefficients

$$\mathbb{D}_{\mathrm{masked}} \leftarrow \varnothing,\ \mathbb{D}_{\mathrm{aug}} \leftarrow \varnothing$$

for $\{X,\ Y\} \in \mathbb{D}_{\mathrm{train}}$ **do**

$\tilde{X},\ \tilde{Y}, P \leftarrow \mathrm{TOKENIZE}(X,\ Y)$ //Gain tokens $\tilde{X}$, labels $\tilde{Y}$ and positions P from tokenization

$\tilde{X} \leftarrow \mathrm{FINETUNEMASK}(\tilde{X},\ \eta)$ //Randomly mask entities for fine-tuning, η is the probability of masking

$$\mathbb{D}_{\mathrm{masked}} \leftarrow \mathbb{D}_{\mathrm{masked}} \cup \left\{(\tilde{X},\ \tilde{Y}, P)\right\}$$

end for

$\tilde{\mathcal{M}}_{\mathrm{finetune}} \leftarrow \mathrm{FINETUNE}\left(\tilde{\mathcal{M}},\ \mathbb{D}_{\mathrm{masked}}\right)$ //Fine-tune EPT on masked tokens and prompt

for $\{X, Y\} \in \mathbb{D}_{\mathrm{masked}}$ **do**

repeat R **times**:

$\tilde{X},\ \tilde{Y}, P \leftarrow \mathrm{TOKENIZE}(X,\ Y)$ //Gain tokens, labels and positions from tokenization

$\tilde{X} \leftarrow \mathrm{GENMASK}(\tilde{X},\ \mu)$ //Randomly mask entities for generation, μ is the probability of masking

$X_{\mathrm{aug}} \leftarrow \mathrm{RANDCHOICE}\left(\tilde{\mathcal{M}}_{\mathrm{finetune}}\left(\tilde{X},\ \tilde{Y}, P\right),\ \mathrm{Top}\,k\right)$ //Augment data with EPT, k represents the number of randomly selected tokens at each position

$$\mathbb{D}_{\mathrm{aug}} \leftarrow \mathbb{D}_{\mathrm{aug}} \cup \left\{X_{\mathrm{aug}}\right\}$$

end for

$\mathcal{N}_{\mathrm{NER}} \leftarrow \mathrm{FINETUNE}\left(\mathcal{N},\ \mathbb{D}_{\mathrm{train}} \cup \mathbb{D}_{\mathrm{aug}}\right)$ //Train NER model on training and generating dataset

2 Experiments

2.1 Dataset

Most of experiments are conducted on CoNLL NER dataset[26, 27] of three languages where $\mathbb{L}$= {English (En), Spanish (Es)}. Initially, N sentences are sampled as $\mathbb{D}^{l,N}_{\text{train}}$ from each language $l \in \mathbb{L}$, where $N \in \{200, 400, 600, 800\}$ for various low-resource scenarios. Then we gain development set $\mathbb{D}^{l,N}_{\text{dev}}$ with the same procedure. Finally, $\mathbb{D}^{l}_{\text{test}}$ is the full test set for each language in evaluation process. For monolingual experiments, we use $\mathbb{D}^{l,N}_{\text{train}}$ as the original data, $\mathbb{D}^{l,N}_{\text{dev}}$ as the development set and $\mathbb{D}^{l}_{\text{test}}$ as the test set.

We also conduct monodomain, cross-domain and multi-domain experiments on Cross NER dataset[24] of five domains, where $\mathbb{M}$= {AI, Literature, Music, Politics, Science}. $\mathbb{D}^{l}_{\text{train}}$, $\mathbb{D}^{l}_{\text{dev}}$ and $\mathbb{D}^{l}_{\text{test}}$ are constructed as the same way above, where $l \in \mathbb{L}$. For cross-domain experiments, we augment data $\mathbb{D}^{s}_{\text{aug}}$with the source train set $\mathbb{D}^{s}_{\text{train}}$ and the source development set $\mathbb{D}^{s}_{\text{dev}}$ as the source domain $s \in \mathbb{M}$. Thus $\mathbb{D}^{s}_{\text{aug}}$ is combined with the target train set $\mathbb{D}^{t}_{\text{train}}$ for the final NER training, where target domain $t \in \mathbb{M}$. The whole train sets, development sets and test sets are integrated into the multi-domain train set $\mathbb{D}^{\text{mix}}_{\text{train}} = \bigcup_{m \in \mathbb{M}} \mathbb{D}^{m}_{\text{train}}$, the multi-domain development set $\mathbb{D}^{\text{mix}}_{\text{dev}} = \bigcup_{m \in \mathbb{M}} \mathbb{D}^{m}_{\text{dev}}$ and the multi-domain test set $\mathbb{D}^{\text{mix}}_{\text{test}} = \bigcup_{m \in \mathbb{M}} \mathbb{D}^{m}_{\text{test}}$ for multi-domain experiments.

2.2 Experimental Setting

1) EPT Finetuning EPT parameters are initialized by XLM-RoBERTa-base[28] with a masked language modeling head. And EPT is finetuned for 20 epochs with Adam optimizer[29], using learning rate set to 1E−5 and batch size set to 16 with gradient accumulation for per 2 batch.

2) NER Model We use XLM-RoBERTa-base[28] with a head of a dropout layer and a linear layer as the NER model for our experiments. The same optimizer and learning rate are adopted as finetuning. It is generally trained for 10 epochs (20 epochs for Spanish corpus) before the best model is picked according to the performance over development set. The averaged Macro-F1 and Marco-F1 without context are reported over 3 runs while evaluating on test sets.

3) Hyperparameter Tuning The coefficient α of label embedding, the coefficient β for number of EPT augmentation and the round R is respectively set as 1E−2, 5E−3 and 3. We also tune embedding hyperparameters on the development set with grid search.

4) Computing Infrastructure Our experiments are conducted on NVIDIA 3090 GPU.

2.3 Baseline Methods

We compare our EPT with the following two methods to demonstrate its effectiveness:

1) Gold-Only[21] The NER model is trained only on the original gold training set.

2) MELM[21] We first linearize the sequences with labels and fine-tune MELM with them. Then MELM randomly masks entity tokens and predicts a masked entity token not only considering label information but also relying on the context words. MELM is enough to be the baseline method as its substantial improvement over other methods, such as Label-wise Substitution[7], Data Augmentation with a Generation Approach (DAGA) [23] and Multilingual Data Augmentation (MulDA)[13].

2.4 Experimental Results

2.4.1 Monolingual NER

We use the average F1 score of recognition of different entities to measure the performance of the methods. Table 1 shows the results of monolingual low-resource NER**.** From it we can see our proposed EPT achieves the highest average macro-F1 scores across various levels of low-resource NER, highlighting its effectiveness on monolingual tasks. In comparison to the best-performing method MELM, EPT achieves average macro F1-score improvements of 1.6%, 5.9%, 1.2%, and 3.1% at 200, 400, 600, and 800 levels which means the number of the original sentences, respectively. It is worth noting that EPT achieves a monolingual macro-F1 of 56.8%, while the Gold-Only method fails to predict any entities given only 200 sentences. EPT, and its variations initialized with different weights, consistently achieve the highest F1 scores on the English corpus at different low-resource levels, even though MELM outperforms methods without data augmentation by a considerable margin. Furthermore, the varying performance of EPT initialized with different weights is attributed to the value of initialization, and we will discuss this in more details in Section 2.4.3.

As the neural network failed to converge after 10 epochs, an additional 10 epochs of training were conducted on the Spanish corpus. Remarkably, the proposed EPT outperformed the baseline methods and its variety in terms of the marco-F1 scores. Notably, EPT achieved

even more significant gains in the context of the English corpus at each low-resource level. We hypothesize that this could be attributed to the fact that entities in Spanish are easier to truncate into sub-words, which can be reorganized randomly into novel entities and provide substantial improvements to NER models. To support this assumption, we present statistical results about recombined words in Section 3.1, which provides convincing evidence.

Specifically, at a low-resource level of 400 English training samples, EPT outperforms the baseline method MELM by 8.3% in terms of *F*1 score. We attribute this improvement to the special tokens that are injected into the sentences, which separates entities from their contexts and from other entities, thereby leading to less diversity of entities. Rather than inserting label tokens into sentences, EPT focuses on implicitly embedding label and position information as prompt, which ensures the diversity and quality of augmented entities across different low-resource levels and languages.

Table 1 Macro-*F*1 of monolingual low-resource NER %

Level	Method	En	Es	Avg
200	Gold-Only	19.85/10.28	39.62/32.25	29.74/21.27
	MELM	56.01/50.70	66.43/62.33	61.22/56.52
	EPT*	49.96/43.88	63.06/58.54	56.51/51.21
	EPT	56.80/51.58	68.92/65.12	62.86/58.35
400	Gold-Only	47.84/41.47	61.45/56.69	54.65/49.08
	MELM	54.22/48.64	70.76/67.18	62.49/57.91
	EPT*	71.68/68.28	69.10/65.31	70.39/66.80
	EPT	62.53/57.98	74.18/71.03	68.36/64.51
600	Gold-Only	61.80/57.15	64.74/60.39	63.27/58.77
	MELM	73.45/70.28	74.55/71.44	74.00/70.86
	EPT*	74.87/71.89	68.69/64.82	71.77/68.36
	EPT	75.30/72.38	75.15/72.12	75.23/72.25
800	Gold-Only	75.73/72.84	67.29/63.24	71.51/68.04
	MELM	79.74/77.32	73.59/70.47	76.67/73.90
	EPT*	80.14/77.77	79.79/77.33	79.97/77.55
	EPT	79.67/77.24	79.94/77.50	79.81/77.37

Note: En and Es mean English data and Spanish data. The numbers separated by slashes represent macro-*F*1 scores with and without non-entity class, respectively. EPT initializes the weights of the embedding layers with special words, such as "location" for "B-LOC" and "I-LOC", while EPT* initializes them with random values

2.4.2 Cross-domain NER

In our experiments on cross-domain low-resource NER, we first applied EPT directly on the training set of each domain and the concatenation of them. We then utilized the augmented data generated by this step to fine-tune NER models using the training sets and development sets of target domains. Finally, we evaluated the NER models on the test sets of target domains. Since expanding the method MELM to multiple domains was costly, we evaluated the performance of the Gold-Only baseline. As illustrated in Fig. 3(a), EPT achieved substantial improvement over the Gold-Only baseline. Compared with the baseline, EPT achieved absolute gains of 19.6, 22.4, 28.9, 8.6, and 24.9 on the AI, literature, music, politics, and science domains, respectively, demonstrating its effectiveness on monodomain NER (Fig. 3(b)).

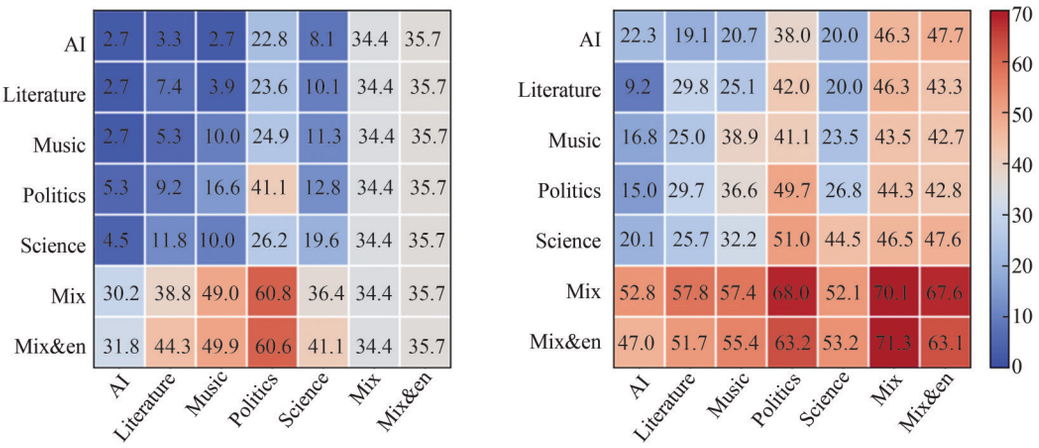
In addition to the results of monodomain NER, the results of cross-domain NER are depicted in Fig. 3, where it is evident that EPT leads to significant improvement on cross-domain NER tasks with different source and target domains. This highlights the importance of data augmentation techniques like EPT for cross-domain NER, even when the augmented data is obtained from a source domain that is quite distinct from the target domain.

Besides, the results of multidomain NER indicate that the NER models fine-tuned on mixed training sets achieve considerably higher *F*1 scores. For instance, the politics corpus achieves an *F*1 score of 68.0 with the domain-mixed training set, while it achieves only 49.7 with the politics training set. However, it is also observed that having more training data does not always lead to better evaluation scores. In fact, in most cases, the *F*1 scores decrease when English augmented data is mixed into the training set, which could be attributed to the noise in the roughly labeled corpus interfering with the prediction of specific fields.

2.4.3 Ablation study

The results presented in Table 2 demonstrate the efficacy of the position and label embedding used in EPT. Specifically, when fine-tuning EPT without label-embedding (EPT without LE) or without position-embedding (EPT without PE), the *F*1 scores drop considerably across different low-resource levels. This indicates that the position and label embedding layers play a crucial role in improving the performance of EPT.

Furthermore, the comparison of EPT without LE



(a) Results of cross-domain low-resource NER with Gold-Only data

(b) Results of cross-domain low-resource NER with data augmented by our EPT

Fig.3 Results of cross-domain low-resource NER

The data on the horizontal axis represents the domain of the training set, and the data on the vertical axis represents the domain of the development set and the testing set, where "mix" represents the mixed data of the five domains, and "mix&en" represents the mixed data of 5 domains and the English dataset[26] mentioned above. Thus, the data on the diagonal shows the performance of methods on monodomain NER task, and the other data shows the ability of methods on cross-domain NER task and multi-domain NER task

and EPT without PE with the Gold-Only baseline shows that the label and position information embedded as prompts in EPT indeed help generate diverse and adequate entities from the original data, leading to significantly better NER performance.

It is important to note that the initialization of the label embedding layer can have a significant impact on the performance of the EPT model. The experiments in Table 3 show that the choice of initialization weights can affect the model's stability and performance. In this case, the EPT model initialized with the embedding weights of special tokens is the most stable of all the models tested in Table 1.

Table 2 Macro-*F*1 of the ablation experiments %

Level	Method	Macro-*F*1	Macro-*F*1 without others
200	Gold-Only	19.85	10.28
	without LE	56.32	51.04
	without PE	54.85	49.39
	EPT	56.80	51.58
400	Gold-Only	47.84	41.47
	without LE	53.25	47.55
	without PE	56.89	51.65
	EPT	62.53	57.98
600	Gold-Only	61.80	57.15
	without LE	72.94	69.70
	without PE	73.68	70.53
	EPT	75.30	72.38
800	Gold-Only	75.73	72.84
	without LE	79.02	76.52
	without PE	78.63	76.10
	EPT	79.67	77.24

Note: Macro-*F*1 without others means the macro-*F*1 value without non-entity class

Table 3 Macro-*F*1 of EPT with different initial weights %

Initialization	Macro-*F*1	Macro-*F*1 without others
with LE random	77.41	74.73
with LE normal avg	78.96	76.45
with LE normal max	80.14	77.77
with LE special	79.67	77.24

Note: Macro-*F*1 without others means the macro-*F*1 value without non-entity class

2.4.4 Hyperparameter tuning

To determine the optimal hyperparameters for label embedding coefficient α and position embedding coefficient β, we conduct a grid search in {5E–3, 1E–2, 2E–2} and {2.5E–3, 5E–3, 1E–2}. EPT is finetuned to generate English augmented data on CoNLL dataset. Then a NER tagger is trained on the data and evaluated on English development set for its performance. As shown in Table 4, the best development set *F*1 is achieved when α=1E–2

and β=5E−3, and we adopt it for the rest of this work.

Table 4 Development set *F*1 for label embedding and position embedding

%

β	α		
	5E−3	1E−2	2E−2
2.5E−3	77.16	77.96	77.78
5E−3	78.80	**79.76**	76.85
1E−2	78.73	78.71	77.34

3 Analysis and Discussion

The progress made by EPT in the NER mission is clearly related to the label and position embedding we proposed. We suggest that label embedding can produce more valuable samples, and our position embedding reduces samples containing location errors, both of which have been responsible for the success of EPT. Next the correctness of our ideas is shown through data.

3.1 Number of Recombined Entities

As illustrated in Fig. 4, our proposed EPT consistently outperforms the baseline method MELM in generating more recombined entities across various low-resource levels and languages. These recombined entities are formed by meaningfully combining parts of different entity words during the prediction of masked entity tokens. The red lines in the figure represent the ratio of recombined entities generated by augmentation methods to the original samples, providing further evidence of the superior diversity of augmented entities achieved by EPT. Interestingly, we observe that more recombined entities are generated on the Spanish corpus than the English corpus, regardless of the augmentation method used.

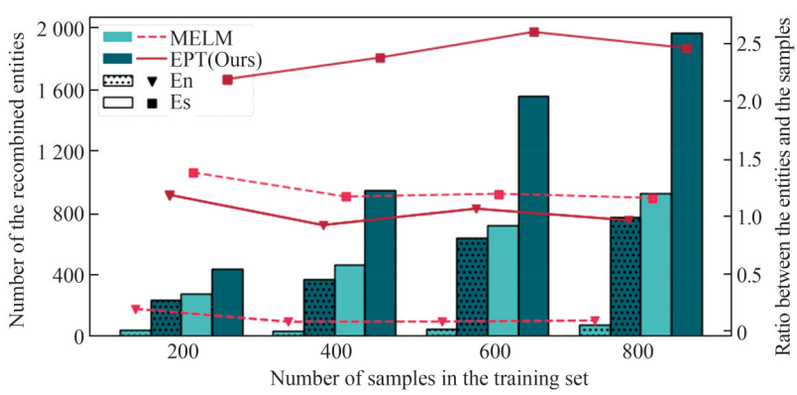


Fig.4 Numbers of recombined entities

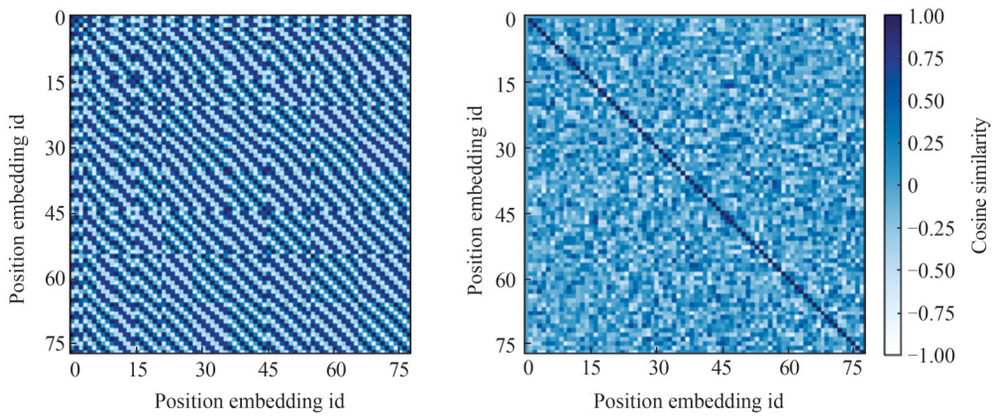
Bars present the accounts of the recombined entities generated by MELM and our EPT. Lines present the ratios of the recombined entities to the numbers of the original samples at each low-resource level

3.2 Role of Position Embedding

In addition to the demonstrated effectiveness of position embedding (PE) in Section 2.4.3, the underlying reasons for its success are still not clear. To investigate this, we computed the cosine similarity between the PE weights of different PE methods and present the results in Fig. 5. Our analysis revealed that the absolute PE (APE), shown as Fig. 5(a), encodes positional information with translation invariance, monotonicity, and symmetry [30], whereas our relative PE (RPE), shown as Fig. 5(b), learns position embeddings with orthogonality. This implies that masked tokens will be predicted with entity tokens of the same position, leading to better entity recognition. As can be seen from Table 5, our learnable APE achieves a remarkable *F*1 score among these PE methods.

4 Conclusion

In this work, we have introduced EPT as a novel approach for data augmentation in low-resource NER. By leveraging label and position embeddings, EPT can predict masked entity tokens based on contextual information, thereby generating augmented data with diverse and novel entities while avoiding the issue of label-token misalignment. Furthermore, we have demonstrated the effectiveness of EPT on various NER datasets, including monolingual, monodomain, cross-domain,



(a) Cosine similarity between the PE weights of APE (b) Cosine similarity between those of our learnable RPE

Fig.5 Comparison of different position embedding methods

Table 5 Macro-*F*1 for different PE methods %

PE methods	Macro-*F*1
Learnable APE	48.48
APE	46.90
Learnable RPE	**52.78**

and multidomain scenarios.

However, one limitation of our approach is that we have only evaluated it in English and Spanish corpora. Further research could investigate its performance across a wider range of languages and extremely low-resource NER tasks. Additionally, while our experiments demonstrate significant improvements in performance, there may be further optimization of hyperparameters that could lead to even better results.

Acknowledgements

We would like to express our gratitude to Professor Wei Xiaomei for inspiring my interest in NLP tasks and for providing invaluable guidance and feedback throughout the entire process. Her insightful comments and expertise have greatly contributed to the success of this research.

References

[1] Hamdi A, Carel E, Joseph A, *et al*. Information extraction from invoices[J]. *Document Analysis and Recognition – ICDAR*, 2021, **2**: 699-714.

[2] Alwaneen T H, Azmi A M, Aboalsamh H A, *et al*. Arabic question answering system: A survey[J]. *Artificial Intelligence Review*, 2022: 1-47.

[3] Kouris P, Alexandridis G, Stafylopatis A. Abstractive text summarization based on deep learning and semantic content generalization[C]// *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* Florence: Association for Computational Linguistics, 2019: 5082-5092.

[4] Li Z, Qu D, Xie C J, *et al*. Language model pre-training method in machine translation based on named entity recognition[J]. *International Journal on Artificial Intelligence Tools*, 2020, **29**(7n08): 2040021.

[5] Cotterell R, Duh K. Low-resource named entity recognition with cross-lingual, character-level neural conditional random fields[J]. *International Joint Conference on Natural Language Processing*, 2017, **2**: 91-96.

[6] Feng X, Feng X, Qin B, *et al*. Improving low resource named entity recognition using cross-lingual knowledge transfer[J]. *IJCAI*, 2018, **1**: 4071-4077.

[7] Dai X, Adel H. An analysis of simple data augmentation for named entity recognition[EB/OL]. [2022-12-18]. https://arxiv.org/abs/2010.11683.

[8] Chen S Q, Pei Y J, Ke Z W, *et al*. Low-resource named entity recognition via the pre-training model[J]. *Symmetry*, 2021, **13**(5): 786.

[9] Arkhipov M, Trofimova M, Kuratov Y, *et al*. Tuning multilingual transformers for language-specific named entity recognition[C]// *Proceedings of the 7th Workshop on Balto-Slavic Natural Language Processing.* Stroudsburg: Association for Computational Linguistics, 2019: 89-93.

[10] Wei J, Zou K. EDA: Easy data augmentation techniques for boosting performance on text classification tasks[EB/OL]. [2022-11-06]. https://arxiv.org/abs/1901.11196.

[11] Kumar V, Choudhary A, Cho E. Data augmentation using pre-trained transformer models[EB/OL]. [2021-12-26]. https://arxiv.org/abs/2003.02245.

[12] Chen J, Wang Z, Tian R, *et al*. Local additivity-based data

augmentation for semi-supervised NER[EB/OL]. [2022-10-26]. https://arxiv.org/abs/2010.01677.

[13] Liu L, Ding B, Bing L, *et al*. MulDA: A multilingual data augmentation framework for low-resource cross-lingual NER[C]// *The* 59*th Annual Meeting of the Association for Computational Linguistics and the* 11*th International Joint Conference on Natural Language Processing.* Stroudsburg: Association for Computational Linguistics, 2021, **1**: 5834-5846.

[14] Kobayashi S. Contextual augmentation: Data augmentation by words with paradigmatic relations[EB/OL]. [2022-10-26]. https://arxiv.org/abs/1805.06201.

[15] Wu X, Lv S W, Zang L J, *et al*. Conditional BERT contextual augmentation[C]// *International Conference on Computational Science*. Cham: Springer-Verlag, 2019, **4**(19): 84-95.

[16] Tsai C T, Mayhew S, Roth D. Cross-lingual named entity recognition via wikification[C]// *Proceedings of the* 20*th SIGNLL Conference on Computational Natural Language Learning*. Stroudsburg: Association for Computational Linguistics, 2016: 219-228.

[17] Li X, Bing L, Zhang W, *et al*. Unsupervised cross-lingual adaptation for sequence tagging and beyond[EB/OL]. [2022-10-26]. https://arxiv.org/abs/ 2010.12405.

[18] Li K, Chen C B, Quan X J, *et al*. Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation[C]// *Proceedings of the* 58*th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg: Association for Computational Linguistics, 2020: 7056-7066.

[19] Song K T, Tan X, Qin T, *et al*. MASS: Masked sequence to sequence pre-training for language generation[EB/OL]. [2022-12-21]. https://arxiv.org/abs/1905.02450.

[20] Lin H Y, Lu Y J, Tang J L, *et al*. A rigorous study on named entity recognition: Can fine-tuning pretrained model lead to the promised land?[C]// *Proceedings of the* 2020 *Conference on Empirical Methods in Natural Language Processing* (*EMNLP*). Stroudsburg: Association for Computational Linguistics, 2020: 7291-7300.

[21] Zhou R, Li X, He R, *et al*. MELM: Data augmentation with masked entity language modeling for low-resource NER [C]// *The* 60*th Annual Meeting of the Association for Computational Linguistics.* Stroudsburg: Association for Computational Linguistics, 2022, **1**: 2251-2262.

[22] Ding B S, Liu L L, Bing L D, *et al*. DAGA: Data augmentation with a generation approach for low-resource tagging tasks[C]// *Proceedings of the* 2020 *Conference on Empirical Methods in Natural Language Processing* (*EMNLP*). Stroudsburg: Association for Computational Linguistics, 2020: 6045-6057.

[23] Liu Y H, Gu J T, Goyal N, *et al*. Multilingual denoising pre-training for neural machine translation[J]. *Transactions of the Association for Computational Linguistics*, 2020, **8**: 726-742.

[24] Liu Z H, Xu Y, Yu T Z, *et al*. CrossNER: Evaluating cross-domain named entity recognition[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, **35**(15): 13452-13460.

[25] Fu J L, Liu P F, Zhang Q. Rethinking generalization of neural models: A named entity recognition case study[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, **34**(5): 7732-7739.

[26] Sang T K, Erik F. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition[J]. *Conference on Natural Language Learning.* Stroudsburg: Association for Computational Linguistics, 2002.

[27] Tjong Kim Sang E F, De Meulder F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition[C]// *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL* 2003. Morristown: Association for Computational Linguistics, 2003: 142–147.

[28] Conneau A, Khandelwal K, Goyal N, *et al*. Unsupervised cross-lingual representation learning at scale[C]// *The* 58*th Annual Meeting of the Association for Computational Linguistics.* Stroudsburg: Association for Computational Linguistics, 2020: 8440-8451.

[29] Kingma D P, Ba J. Adam: A method for stochastic optimization[EB/OL]. [2022-11-28]. https://arxiv.org/abs/ 1412.6980.

[30] Wang B Y, Shang L F, Lioma C, *et al*. On position embeddings in bert[C]// *International Conference on Learning Representations.* Vienna: ICLR2021, 2021.

□