PMRC: Prompt-Based Machine Reading Comprehension for Few-Shot Named Entity Recognition

Jin Huang, Danfeng Yan*, Yuanqiang Cai

Beijing University of Posts and Telecommunications
Xitucheng Road 10, Beijing, China
jinhuang@bupt.edu.cn, yandf@bupt.edu.cn, caiyuanqiang@bupt.edu.cn

Abstract

The prompt-based method has been proven effective in improving the performance of pre-trained language models (PLMs) on sentence-level few-shot tasks. However, when applying prompting to token-level tasks such as Named Entity Recognition (NER), specific templates need to be designed, and all possible segments of the input text need to be enumerated. These methods have high computational complexity in both training and inference processes, making them difficult to apply in real-world scenarios. To address these issues, we redefine the NER task as a Machine Reading Comprehension (MRC) task and incorporate prompting into the MRC framework. Specifically, we sequentially insert boundary markers for various entity types into the templates and use these markers as anchors during the inference process to differentiate entity types. In contrast to the traditional multi-turn question-answering extraction in the MRC framework, our method can extract all spans of entity types in one round. Furthermore, we propose word-based template and example-based template that enhance the MRC framework's perception of entity start and end positions while significantly reducing the manual effort required for template design. It is worth noting that in cross-domain scenarios, PMRC does not require redesigning the model architecture and can continue training by simply replacing the templates to recognize entity types in the target domain. Experimental results demonstrate that our approach outperforms state-of-the-art models in low-resource settings, achieving an average performance improvement of +5.2% in settings where access to source domain data is limited. Particularly, on the ATIS dataset with a large number of entity types and 10-shot setting, PMRC achieves a performance improvement of +15.7%. Moreover, our method achieves a decoding speed 40.56 times faster than the template-based cloze-style approach.

Instruction

Named Entity Recognition (NER) is a fundamental problem in the field of Natural Language Processing (NLP). It aims to identify spans of text that correspond to named entities and assign them to predefined entity categories such as person, location, organization (Tjong Kim Sang and De Meulder 2003). Typically, NER tasks are regarded as sequence labeling problems (Ma and Hovy 2016), where each entity in the input sequence is assigned a specific label. However, manual annotation of a large corpus for NER requires substantial time and expertise from domain experts, making it a costly endeavor. Moreover, models trained using these methods need to adjust their structures and undergo retraining when faced with new data, limiting their ability to leverage knowledge learned from the original data.

To address these challenges, researchers have proposed the concept of few-shot NER (Wiseman and Stratos 2019; Yang and Katiyar 2020; Ziyadi et al. 2020). In few-shot NER, a pre-trained language model is used as the base model, and only a small amount of labeled data is utilized for fine-tuning in a specific domain. This approach significantly reduces the cost of annotation and enables the rapid construction of a well-performing NER model for new domains. However, existing few-shot NER models often require reconfiguration to adapt to new entity categories, which limits their performance, especially when labeled data is limited. Additionally, many existing models require the use of a unified label set during both training and testing, which results in poor recognition capability for newly encountered entity categories.

Recently, prompt-based fine-tuning (Liu et al. 2023; Li and Liang 2021; Lester, Al-Rfou, and Constant 2021) have emerged as an important new paradigm in the NLP community. Prompt-based methods aim to bridge the gap between pre-training tasks and downstream tasks by reconfiguring the input of pre-trained language models. This paradigm effectively harnesses the capabilities of language models. Prompt-based approaches have been widely adopted in the few-shot learning domain and have shown promising results. For example, in the NER field, (Cui et al. 2021) redefine the task as a cloze-style task, where prompts are used to guide the model to fill in the missing entity information. However, existing methods often require manual design of prompting templates, and in the case of limited samples, the model's sensitivity to templates makes the search for the optimal template time-consuming. Additionally, these methods require enumerating all possible entity spans in the original text during the training and prediction processes to predict entity types, resulting in computational complexity similar to that of n-gram models.

To address these challenges, we propose a prompt-based machine reading comprehension model (PMRC) for few-

*Corresponding author

shot NER. Inspired by (Shrimal et al. 2022), our model fills the template with boundary tokens for various entity types to predict entities. During the prediction process, by assigning different entity type start and end labels to the text sequence, PMRC is able to extract all entities in one round from the original text. Furthermore, we do not introduce additional parameters as classifiers, but rather use the dot product between the boundary tokens and the text tokens as the probability values for different entity types. This allows PMRC to be transferred to new domains without adjusting the model structure. Additionally, we investigate the impact of various templates that require minimal manual design, greatly enhancing the model's perception of entity start and end positions by incorporating label words or examples into the templates. PMRC achieves superior or competitive results on multiple datasets. In summary, PMRC consists of the following contributions:

- We introduce prompting into the MRC framework, where we sequentially insert boundary markers for various entity types into the templates. During inference, these markers are used as anchors to predict all entities in a single round, eliminating the need for enumerating entity spans and multiple rounds of QA.
- We propose word-based template and example-based template that do not require manual intervention, effectively enhancing the MRC framework's perception of entity start and end positions in few-shot scenarios. Moreover, in cross-domain settings, PMRC does not require redesigning the model structure and can continue training by simply replacing the templates. The knowledge learned from the source domain also contributes to a +2% improvement in average performance.
- We conduct extensive experiments on several benchmark datasets, and the research results demonstrate that PMRC achieves comparable performance in the standard supervised setting and outperforms the state-of-the-art models in low-resource scenarios. Specifically, in the few-shot setting without access to the source domain, PMRC achieves an average performance improvement of +5.2%. Additionally, PMRC exhibits faster training and inference speeds compared to SOTA models.

Related Work

Named Entity Recognition

Traditional approaches to NER have typically framed it as a sequence labeling problem, and with the emergence of pre-trained language models like BERT (Devlin et al. 2019), combining them with a linear classification layer has become a common solution for NER (Ma and Hovy 2016; Liu et al. 2019; Zhang et al. 2020; Liu et al. 2021). However, these methods often struggle to fully leverage the capabilities of language models in low-resource scenarios within a specific domain. Moreover, they require model restructuring and retraining when facing new domains, making them unsuitable for low-resource settings.

Currently, one major research direction in low-resource NER is prototype-based methods, where a prototype is constructed for each entity class, and entity types are assigned by measuring the distance between text sequences and prototypes (Fritzler, Logacheva, and Kretov 2019; Wiseman and Stratos 2019; Yang and Katiyar 2020; Henderson and Vulic 2021; Hou et al. 2020; Lin et al. 2019; Ji et al. 2022; Huang et al. 2022b). However, these methods heavily rely on support-set entities, and their performance deteriorates significantly as the discrepancy between the source and target domains increases.

Recently, there have been studies aiming to enhance models' perception in few-shot domains by introducing external knowledge, such as label knowledge. However, these methods still face limitations. For example, EntLM (Ma et al. 2022b) incorporates high-frequency entity mentions obtained through distantly supervised datasets to construct prototypes for entity tokens. However, obtaining reliable entity mentions requires a robust distantly supervised method(Liang et al. 2020), and as unlabeled data increases, the entity mentions need to be adapted, which means the model needs to reset the word representations of special labels and undergo retraining. In contrast, our proposed method leverages semantic knowledge from labels effectively and exhibits excellent capabilities in cross-domain transfer.

MRC-Based NER

Recently, the paradigm of Machine Reading Comprehension (MRC) (Li et al. 2019, 2020; Liu et al. 2022) has been applied to the field of Named Entity Recognition (NER) with rich resources and has achieved remarkable results. Unlike sequence labeling, the MRC paradigm focuses only on the start and end of entities, and its input sequence is in the form of $(question, original\ \ text)$. The information contained in the question helps the model extract the corresponding entities. However, such methods require the construction of duplicate samples with different entity categories from the original text, which greatly increases the training and inference time of the model. To address this issue, NER-MQMRC (Shrimal et al. 2022) designs questions that encompass multiple categories and extracts various types of entities in parallel at the output layer. However, this approach is prone to exacerbating the token-level class imbalance, making it challenging to adapt to extremely resource-constrained few-shot scenarios. Unlike the aforementioned MRC methods, our approach not only demonstrates excellent few-shot capabilities but also fully leverages the knowledge transferability of the MRC paradigm. Furthermore, our method also excels in terms of inference speed.

Prompt Learning in NER

The concept of prompts originated from the idea of in-context learning in GPT-3 (Brown et al. 2020). Extensive research (Schick and Schütze 2021; Schick, Schmid, and Schütze 2020; Ben-David, Oved, and Reichart 2022; Chen et al. 2022b; Ding et al. 2022) has shown that compared to traditional fine-tuning approaches, prompt learning can effectively bridge the gap between pre-training and downstream tasks, resulting in superior performance in low-resource and cross-domain scenarios. Typically, prompt learning transforms downstream tasks into cloze-style tasks,

However, this method is not very friendly for NER tasks, as it requires dealing with the significant overhead of enumerating candidate entity spans (Cui et al. 2021). Moreover, designing suitable templates also requires a considerable amount of manual effort. Therefore, recent studies have addressed this issue by transforming NER into a sequence generation task, as demonstrated in LightNER (Chen et al. 2022a), UIE (Lu et al. 2022), (Chen et al. 2023).

Preliminaries

Few-Shot NER

Assuming we have a resource-rich NER dataset $\mathbb{H} = \left\{\left(\mathbf{X}_1^H, \mathbf{L}_1^H\right), \ldots, \left(\mathbf{X}_K^H, \mathbf{L}_K^H\right)\right\}$, where the input is a text sequence of length n, $\mathbf{X}^H = \left\{x_1^H, \ldots, x_n^H\right\}$, and we use $\mathbf{Y}^H = \left\{y_1^H, \ldots, y_n^H\right\}$ to represent the corresponding label sequence, and adopt $\mathcal{M}^H$ to represent the label set of the rich-resource dataset $\left(\forall y_i^H, y_j^H \in \mathcal{M}^H\right)$. Additionally, we have a low-resource NER dataset $\mathbb{L} = \left\{\left(\mathbf{X}_1^L, \mathbf{Y}_1^L\right), \ldots, \left(\mathbf{X}_J^L, \mathbf{Y}_J^L\right)\right\}$, where the number of labeled data is extremely limited compared to the resource-rich NER dataset (i.e., $J \ll K$). Regarding the low-resource domain, the target label set $\mathcal{M}^L$ $\left(\forall y_i^L, y_j^L \in \mathcal{M}^L\right)$ may be different from $\mathcal{M}^H$. Moreover, to ensure a truly low-resource setting, we eliminate the validation set setting used in previous works (e.g., TemplateNER (Cui et al. 2021), LightNER (Chen et al. 2022a)). This means that our model is trained on a small training set for a certain number of steps and directly tested on the test set, making our scenario more closely resemble real-world conditions.

MRC Model for NER

The NER task using the MRC approach involves extracting entities through a question-answering process. For each entity category, we construct a question and the model outputs the starting and ending positions of the entity in the input text sequence as the answer. This question-answering approach allows us to naturally identify potential entities in the text.

Formally, given an input sequence $X = \{x_1, x_2, \ldots, x_n\}$, where n represents the length of the sequence, and a predefined set of entity categories $C = \{c_1, c_2, \ldots, c_m\}$, where m represents the number of entity categories. First, we construct a natural language question for $\forall c_i \in C$, resulting in $Q = \{q_1, q_2, \ldots, q_m\}$. Then, we process the input into a question-answer format, resulting in:

$$X' = \{(q_1, X), (q_2, X), \ldots, (q_m, X)\}$$

The input $X'_r \in X'$ is passed through a pretrained language model and then fed into two fully connected layers to obtain the start probability sequence and end probability sequence, respectively. $W_{start} \in \mathbb{R}^n \times 2$ and $W_{end} \in \mathbb{R}^n \times 2$ are the weights to learn:

$$P_{\text{start}} = softmax(Encoder(X'_r) \cdot W_{\text{start}})$$
$$P_{\text{end}} = softmax(Encoder(X'_r) \cdot W_{\text{end}})$$

Finally, for the $c_r \in C$, we can get the entity start index sequence Y_{start} and end index sequence Y_{end} and use them to decode the predicted entities:

$$Y_{\text{start}} = \left\{i \mid \text{argmax}\left(P_{\text{start}}^{(i)}\right) = 1, i = 1, \cdots, n\right\}$$
$$Y_{\text{end}} = \left\{j \mid \text{argmax}\left(P_{\text{end}}^{(j)}\right) = 1, j = 1, \cdots, n\right\}$$

Method

In this section, we will describe how our method incorporates prompting into the MRC framework by discussing the model's input and inference components. An overview of PMRC is illustrated in Figure 1. Finally, we will explain how to construct templates using words and examples.

Prompting the MRC Model

Inspired by NER-MQMRC (Shrimal et al. 2022), we have developed a prompt-based MRC model for few-shot NER, as shown in the Figure 1. We leverage the semantic information or contextual information provided by the prompts to enhance the model's sensitivity to entity spans. During the training phase, the special tokens in the prompts serve as anchors for the entity's start and end positions. In the inference phase, we employ a dot-product-based metric to match the labels.

PMRC Input Firstly, in the NER-MQMRC task, the questions for each entity are separated by a special token [ENT]. For PMRC, we use both [ENT_START] and [ENT_END] tokens to delimit each entity category. These tokens are added to the vocabulary of the PLM, such as BERT. Following the setup in COPNER (Huang et al. 2022a), we manually construct a label vocabulary mapping $\mathcal{M}$, where we assign a semantically meaningful word to each entity label belonging to the predefined label set L. We use the entity class name as LW. For example, in most NER datasets, "PER" is used as the label for person entities, and then the specific word "person" is assigned to $\mathcal{M}(\text{PER}) = \text{person}$ for subsequent person entities. Then, for each specific dataset, we design prompts as follows: $P = \{p_1, p_2, ..., p_m\}$, where $p(l_i) = \{\text{[ENT\_START]}\mathcal{M}(l_i)\,\text{[ENT\_END]}\}, \forall l_i \in L$.

Next, the generated prompts are appended to each input sentence X to form an extended input sequence $X' = \{x_1, x_2, \ldots, x_t, p_1, \ldots, p_{m+1}\}$, where t is the length of the original input sentence, and m is the number of entity classes. Additionally, an extra label word is added to represent non-entity classes. Subsequently, X' is input to the $Encoder$ to generate contextualized representations. PMRC treats the final hidden layer output as the representation for each token: $H = Encoder([x_1, ..., x_t, p_1, ..., p_{m+1}]) = [h_1, \ldots, h_t, s_1, h'_1, e_1, \ldots, s_m, h'_m + 1, e_{m+1}]$, where h_i, h'_i, s_i and e_i are the embedding of origin text token, label word token, [ENT_START] and [ENT_END], respectively. Next, we utilize MLP layers to enhance the model's awareness of the start and end positions:

$$H_{\text{start}} = W_{\text{start}} \cdot H$$
$$H_{\text{end}} = W_{\text{end}} \cdot H$$

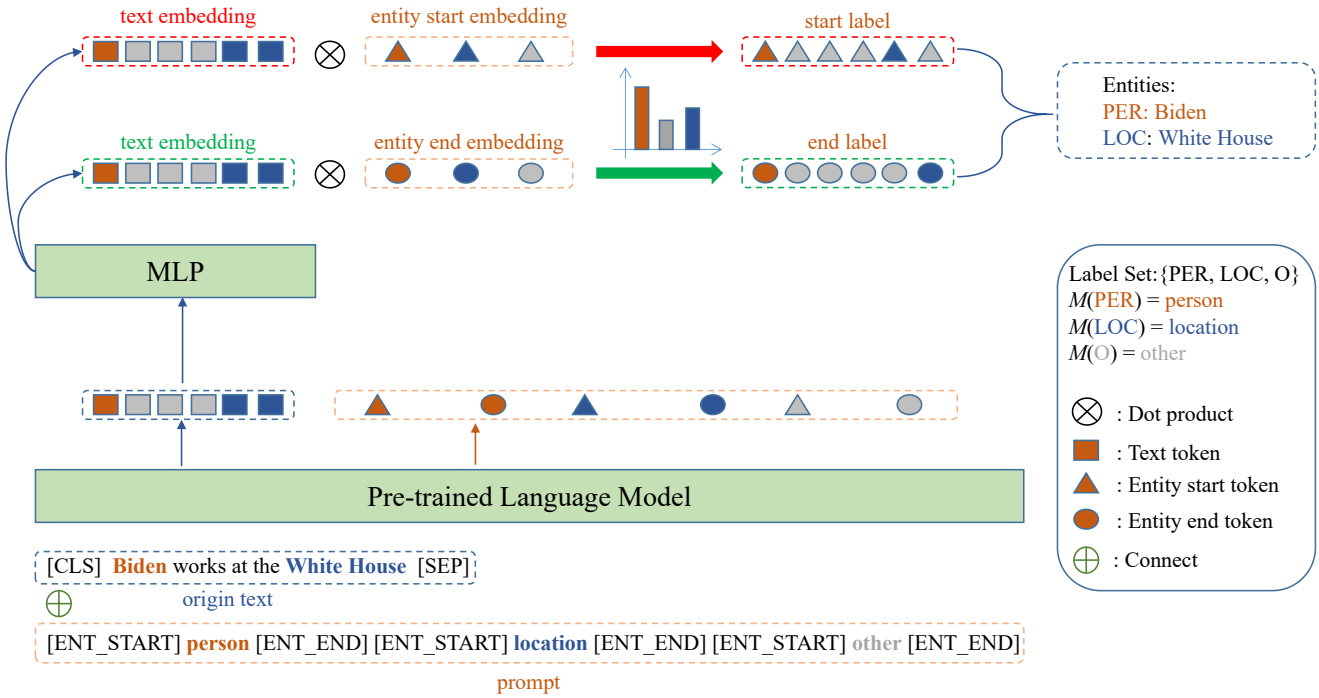

Figure 1: An overview of PMRC.The figure illustrates how the representation of entity start and end markers is generated and how we use them to compute the final model predictions.

The representations of the special tokens in the prompts can serve as prototypes for each entity class, which will be further elaborated in the next section.

Inference of the PMRC In the inference stage, the special tokens in the prompts are considered as metric references for calculating distances with each token. Following the setup in the (Ma et al. 2022a), we utilize dot product as the metric. Formally, the representation of the start token is $S = [s_1, s_2, \ldots, s_{m+1}]$, and the representation of the end token is $E = [e_1, e_2, \ldots, e_{m+1}]$. $\forall x_i$, PMRC can find the nearest metric referent s_j or e_j in the PLM representation space:

$$P_{\text{start}} = softmax(H_{\text{start}} \bigotimes S)$$
$$P_{\text{end}} = softmax(H_{\text{end}} \bigotimes E)$$

In the label sequence of the NER task, there is an extreme imbalance between the number of non-entity tokens and entity boundary tokens. We employ Focal Loss (Lin et al. 2017) to alleviate the performance loss caused by this imbalance. Therefore, for the prediction of the beginning index and the end index, we have the following two losses, where α_l and γ are hyperparameters:

$$\text{FL}_{\text{start}} = -\alpha_l \left(1 - p_l\right)^{\gamma} \log\left(p_l\right), p_l \in P_{\text{start}}$$
$$\text{FL}_{\text{end}} = -\alpha_l \left(1 - p_l\right)^{\gamma} \log\left(p_l\right), p_l \in P_{\text{end}}$$

The overall training objective to be minimized is as follows:

$$\text{FL} = (\text{FL}_{\text{start}} + \text{FL}_{\text{end}})/2$$

In the decoding stage, we adopt a commonly used approach in MRC tasks. Given a starting position s_{ij} for an entity, we find the closest rightmost ending position e_{rj}. The entity span $(x_i, \ldots, x_r)$ is then assigned the entity label $l_j \in L$.

Prompt Construction

The work of TemplateNER demonstrates that manually crafted templates can have a certain impact on the models. In this work, we designed two types of Prompt templates: word-based template and example-based template. The former requires manual selection of label words, while the latter does not require manual intervention. However, when there are a large number of entity classes, the example-based templates can become excessively long. Therefore, in this work, we default to using word-based templates. Figure 2 illustrates examples of these methods, and they are described in detail as follows:

- Word-based template: We first arrange the words in LW. Additionally, we include "other" in LW as a label word for non-entity categories. Then, we insert [ENT_START] and [ENT_END] tokens on both sides of each word to provide separation.
- Example-based template: Randomly select some samples from the training set until these samples contain mentions of various entity classes. Then, concatenate these samples together. Finally, insert [ENT_START] and [ENT_END] tokens on both sides of each entity mention.

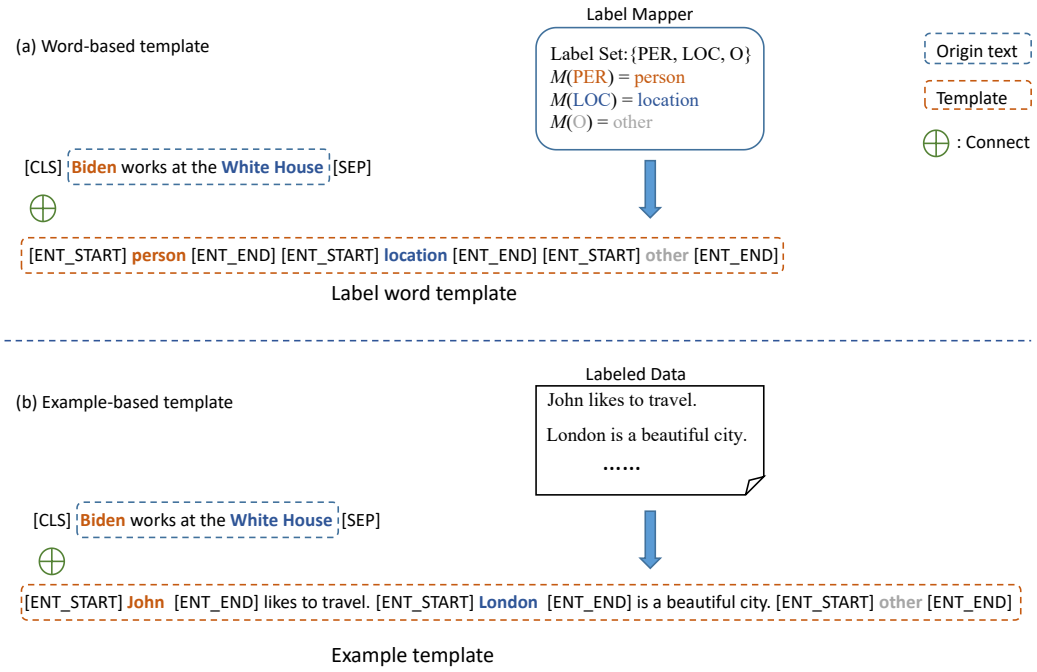

Figure 2: The illustration of prompt construction. (a) We use the label words to fill the template. (b) We will concatenate examples containing different entity types as templates.

Experiments

To validate the effectiveness of PMRC in various settings, we conducted extensive experiments in rich-resource NER and low-resource NER, including in-domain few-shot settings and cross-domain few-shot settings.

Implementation Details

If no specific model is mentioned, we use BERT-base-uncased (Devlin et al. 2019) as the backbone model. This section provides a detailed description of the training process and hyperparameters for each dataset. Considering the instability of few-shot learning, we run each experiment five times with different random seeds and report the average performance. The experiments were conducted using PyTorch on a single Nvidia 3090 GPU. All optimizations were performed using the AdamW optimizer with a linear warm-up schedule in the Standard Supervised Setting and a cosine_with_restarts schedule in the few-shot scenarios. The focal loss function was used with $\alpha = 0.25$ and $\gamma = 2$. Additionally, a weight decay of 0.01 was applied to all non-bias parameters. The details of the training hyperparameters will be described in the following section.

Rich-Resource Setting We fixed the batch size at 16 and set the learning rate to 2e-5. The model was trained for 30 epochs, and evaluation was conducted after 20 epochs. We selected the model with the best performance on the validation set and evaluated it on the test set.

Low-Resource Setting For all experiments, we fixed the batch size at 4, and the model was trained for 30 epochs before being directly tested. The learning rate is set to 1e-4 for the MIT Movie (Liu et al. 2013) and ATIS (Hakkani-Tür et al. 2016) datasets, and 5e-5 for the MIT Restaurant (Liu et al. 2013) dataset.

Models	P	R	F
(Yang, Liang, and Zhang 2018)	-	-	90.77
(Ma and Hovy 2016)	-	-	91.21
(Yamada et al. 2020)	-	-	**94.30**
(Gui et al. 2020)	-	-	92.02
(Li et al. 2020)	92.47	93.27	92.87
(Yu, Bohnet, and Poesio 2020)	**92.85**	92.15	92.50
BERTTagger	91.93	91.54	91.73
Few-Shot Friendly Models	**P**	**R**	**F**
(Wiseman and Stratos 2019)	-	-	89.94
TemplateNER (Cui et al. 2021)	90.51	93.34	91.90
LightNER (Chen et al. 2022a)	92.39	**93.48**	92.93
PMRC	92.42	91.31	91.86

Table 1: Results in the standard supervised NER setting.

Standard Supervised NER Setting

We evaluated the performance of PMRC in the rich-resource setting on the CoNLL03 (Sang and Meulder 2003) dataset. We selected the most recent and competitive models as our baselines, including BERTTagger (Devlin et al. 2019), MRC, TemplateNER (Cui et al. 2021), and LightNER (Chen et al. 2022a). As shown in Table 1, although PMRC was de-

Models	PER	ORG	LOC*	MISC*	Overall
BERTTagger	76.25	75.32	61.55	59.35	68.12
TemplateNER	84.49	72.61	71.98	73.37	75.59
LightNER	90.96	76.88	81.57	**82.08**	78.97
PMRC	**96.36**	**82.15**	**82.29**	66.38	**84.66**

Table 2: Results in the in-domain few-shot NER setting. * indicates the low-resource entity type.

signed for few-shot NER, it performs competitively in the rich-resource environment. This suggests that our model has excellent capabilities in extracting entity spans and their corresponding categories from input texts.

In-Domain Few-Shot NER Setting

Datasets and Baselines: Following LightNER, we constructed a low-resource scenario dataset based on CoNLL03. We performed undersampling on specific entity types while retaining the full data for other types, ensuring a low-resource scenario across entity types within the same domain. Specifically, we selected "LOC" and "MISC" as the low-resource entities and set "PER" and "ORG" as the rich-resource entities. The rich-resource entity categories and the low-resource entity categories share the same text domain. In particular, we undersampled the training set of CoNLL03 and obtained 4,001 training examples, including 2,496 "PER", 3,763 "ORG", 100 "MISC", and 100 "LOC" entities. We evaluated the performance of PMRC under the in-domain few-shot NER setting on this dataset. We selected BERTTagger, the low-resource-friendly model TemplateNER, and LightNER as our baselines.

Results: As shown in Table 2, compared to BERTTagger, we achieved significant performance improvements on both low-resource and rich-resource entity types. Specifically, we achieved an average improvement of **+13.89%** on the low-resource types. Compared to LightNER, PMRC performed better on the low-resource "LOC" type and overall, but slightly weaker on the "MISC" type. We speculate that the MISC entity category includes various types of entities, which may make it challenging for PMRC to learn the boundary information of MISC entities from a small number of samples. Importantly, we achieved an average improvement of **+5.34%** on the rich-resource categories, demonstrating that our approach maintains excellent performance on low-resource types without compromising the performance on rich-resource entity categories. This once again validates the competitiveness of our method designed for few-shot scenarios in the rich-resource setting.

Cross-Domain Few-Shot NER Setting

Datasets and Baselines: In this section, following the setup of LightNER, we evaluated the performance of the model with and without the use of source domain data. Specifically, we employed the CoNLL03 dataset, a general domain dataset, as a resource-rich source domain data. We randomly sampled a subset of training instances from the MIT Movie, MIT Restaurant, and ATIS datasets to serve as the training data for the target domain. We randomly sample a fixed number of instances for each entity type (10, 20, 50, 100, 200, 500 instances per entity type for MIT Movie and MIT Restaurant, and 10, 20, 50 instances per entity type for ATIS). If the number of instances for a particular type is less than the fixed quantity, we use all available instances for training. We selected several competitive methods with the same experimental setup as our baselines, including Neigh.Tag (Wiseman and Stratos 2019), Example-based (Ziyadi et al. 2020), Multi-prototype + NSP (referred to as MP-NSP) (Huang et al. 2020), BERTTagger, TemplateNER, BERT-MRC (Yu, Bohnet, and Poesio 2020), and LightNER.

Results Without Source Data: When training from scratch on the target domain without any available source domain data, we cannot use prototype-based methods. Table 3 shows that PMRC outperforms all SOTA models significantly on the MIT Movie and ATIS datasets. In particular, PMRC achieves an F1 score of 58.2% in the extremely low-shot 10-shot setting on the MIT Movie dataset, surpassing LightNER's results in the 20-shot setting and TemplateNER in the 100-shot setting. Furthermore, in the 10-shot setting on the ATIS dataset, PMRC achieves a **+15.7%** improvement compared to LightNER, even performing comparably to it in the 50-shot setting. PMRC also achieves competitive performance on the MIT Restaurant dataset. Overall, compared to previous state-of-the-art models, PMRC achieves an average improvement of **+5.2%** in all cross-domain few-shot settings.

Results With Source Data: We first train the model extensively on the resource-rich CoNLL03 dataset and then fine-tune it in the low-resource scenario. Table 3 shows that on all three target domain datasets, PMRC achieves performance comparable to LightNER and significantly outperforms other baselines. Compared to the results without source data, there was a **+2%** improvement. This indicates that PMRC can transfer the broad knowledge learned in the resource-rich domain to resource-limited domains. Additionally, we observe that on the ATIS dataset, PMRC without source domain data clearly outperforms the source-trained LightNER.

Analysis

Ablation Study

We conducted ablation experiments on the 10-shot, 20-shot, and 50-shot scenarios in the MIT Movie dataset to analyze the effects of different modules in PMRC. From the experimental results Table 4, it can be observed that PMRC experiences varying degrees of performance degradation when these three components are missing. The most significant performance drop occurs when we do not use label words at all in the constructed templates, particularly with a 5.6% decrease in the 10-shot setting with source domain. Therefore, we can conclude that incorporating label semantic knowledge into templates effectively improves the model's performance in extremely limited data scenarios. Additionally, using an MLP in PMRC to enhance the model's ability to distinguish entity start and end positions also leads to cer-

Source	Methods	MIT Movie						MIT Restaurant						ATIS			Avg.
		10	20	50	100	200	500	10	20	50	100	200	500	10	20	50	
None	BERTTagger	25.2	42.2	49.6	50.7	59.3	74.4	21.8	39.4	52.7	53.5	57.4	61.3	44.1	76.7	90.7	53.2
	TemplateNER	37.3	48.5	52.2	56.3	62.0	74.9	46.0	57.1	58.7	60.1	62.8	65.0	71.7	79.4	92.6	61.6
	BERT-MRC	18.7	48.3	55.5	62.5	80.2	82.1	12.3	37.1	53.5	63.9	65.5	70.4	35.3	63.2	90.2	55.9
	LightNER	41.7	57.8	73.1	78.0	80.6	84.8	48.5	58.0	62.0	70.8	75.5	**80.2**	76.3	85.3	92.8	71.0
	PMRC	**58.2**	**75.2**	**78.7**	**82.0**	**83.7**	**85.7**	**49.5**	**58.2**	**65.4**	**71.8**	**75.5**	77.8	**92.0**	**94.4**	**95.3**	**76.2**
CoNLL03	Neigh.Tag.	0.9	1.4	1.7	2.4	3.0	4.8	4.1	3.6	4.0	4.6	5.5	8.1	2.4	3.4	5.1	3.7
	Example.	29.2	29.6	30.4	30.2	30.0	29.6	25.2	26.1	26.8	26.2	25.7	25.1	22.9	16.5	22.2	26.4
	MP-NSP	36.4	36.8	38.0	38.2	35.4	38.3	46.1	48.2	49.6	49.6	50.0	50.1	71.2	74.8	76.0	49.2
	BERTTagger	28.3	45.2	50.0	52.4	60.7	76.8	27.2	40.9	56.3	57.4	58.6	75.3	53.9	78.5	92.2	56.9
	TemplateNER	42.4	54.2	59.6	65.3	69.6	80.3	53.1	60.3	64.1	67.3	72.2	75.7	77.3	88.9	93.5	68.3
	BERT-MRC	20.2	50.8	56.3	62.9	81.5	82.3	15.8	39.5	54.8	65.8	68.8	73.5	40.5	66.7	91.8	58.1
	LightNER	62.9	75.6	78.8	**82.2**	**84.5**	85.7	58.1	**67.4**	**69.5**	**73.7**	**78.4**	**81.1**	86.9	89.4	93.9	77.9
	PMRC	**65.6**	**76.6**	**79.4**	82.0	84.4	**86.0**	**58.4**	62.5	69.1	73.0	75.3	77.8	**92.5**	**94.7**	**95.4**	**78.2**

Table 3: Results in the cross-domain few-shot NER setting. 10 indicates 10 instances for each entity types.

Source	Methods	MIT Movie		
		10	20	50
None	DEFAULT	**58.2**	**75.2**	**78.7**
	w/o MLP	56.0	74.5	77.6
	w/o label words	52.7	73.9	78.2
	w/o Focal Loss	57.7	74.0	78.6
CoNLL03	DEFAULT	**65.6**	**76.6**	**79.4**
	w/o MLP	62.7	75.4	76.6
	w/o label words	60.0	73.4	78.2
	w/o Focal Loss	61.2	75.9	79.2

Table 4: Ablation Study: (1) *w/o MLP*: We omitted the MLP layer and directly performed dot product between the text embeddings outputted by BERT and the token embeddings. (2) *w/o label words*: We completely excluded label words from the prompt templates, constructing the templates only with start and end markers. (3) *w/o Focal Loss*: We employed cross-entropy loss instead of focal loss for loss calculation.

tain performance improvements.

Analysis of Prompt Templates

Previous studies have shown that prompt-based model is more sensitive to templates, and different templates may have a large performance gap. We experimented on the MIT Movie dataset using the two templates mentioned earlier, where the random word template is words chosen at random to populate a word-based template. Table 5 shows that, the label word-based template performs better than random word template. However, as the amount of data increases, the performance differences among these templates become less significant. Additionally, we observe that the example-based template performs comparably to the label word-based template and does not require manual intervention for design. However, a drawback is that when there are numerous entity categories, the template can become excessively long, causing the input to exceed the maximum encoding length and leading to performance degradation. This indicates that word prompts can effectively enhance model performance with extremely limited data. Furthermore, PMRC exhibits strong robustness in terms of templates, thanks to our model design that effectively leverages the model's machine reading comprehension capabilities.

Prompt Template	5	10	20	50	100	200
Random word Template	33.3	55.4	74.2	78.3	81.2	83.7
Example Template	**44.1**	**58.8**	74.9	78.5	82.1	83.9
Label word Template	41.5	58.2	**75.2**	**78.7**	**82.2**	**84.0**

Table 5: Results of Different Templates on the MIT Movie Dataset. The "Random word Template" refers to the template where random words are used to fill in the prompts.

Inference Efficiency

From the perspective of model complexity, the complexity of the encoder in the backbone Transformer model (Vaswani et al. 2017) is $O\left(n^2d + nd^2\right)$, where d represents the model's dimension. The Transformer decoder performs multi-head attention over the outputs of the encoder for t autoregressive steps, resulting in a complexity of $O\left(\left(n^2d + nd^2\right) \times t\right)$. For models like BART (Lewis et al. 2020), which use both an encoder and an autoregressive decoder, the overall complexity becomes $O\left(\left(n^2d + nd^2\right) \times (t+1)\right)$. In terms of method complexity, PMRC has a complexity of $O\left(1\right)$, TemplateNER has a complexity of $O\left(N^2\right)$, and QaNER (Liu et al. 2022) has a complexity of $O\left(C\right)$. Therefore, in theory, PMRC has certain advantages over the baselines in terms of both model complexity and method complexity. We conducted experiments on the test set of CoNLL03 dataset to evaluate the model's inference efficiency. The results in Table 6 demonstrate that PMRC achieves the fastest inference efficiency, being $40.56\times$ faster than TemplateNER, $4.59\times$ faster than QaNER, and $2.44\times$ faster than LightNER.

Model	Complexity of Model	Complexity of Method	SpeedUp
TemplateNER	$O((n^2d + nd^2) \times (t + 1))$	$O(N^2)$	1.00×
QaNER (Liu et al. 2022)	$O(n^2d + nd^2)$	$O(C)$	8.83×
LightNER	$O((n^2d + nd^2) \times (t + 1))$	$O(1)$	16.65×
PMRC	$O(n^2d + nd^2)$	$O(1)$	40.56×

Table 6: Results of Inference Efficiency on the CoNLL03 Dataset.

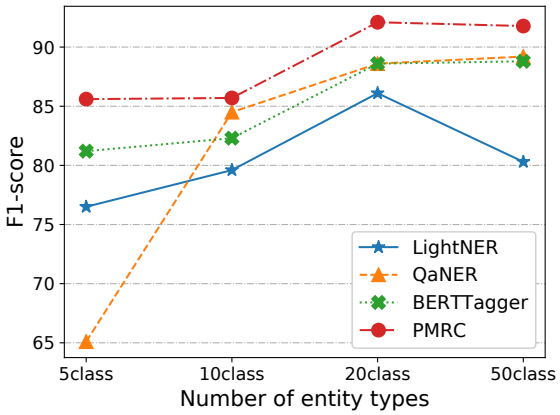

Figure 3: F1-scores under different number of entity types.

Class Increment Experiment

During the experimental process, we observed that PMRC exhibited greater performance improvement on the ATIS dataset, which has a larger number of entity types, compared to the other two datasets. This suggests that our method may have an advantage when dealing with a larger number of entity types. To further investigate this, we conducted experiments on the 10-shot ATIS dataset. Firstly, we performed negative sampling on the dataset by randomly selecting K entity types from the training set. Then, we set the entity labels as "O" for the unselected entity types in both the training and test dataset, and removed examples that did not contain any entity types, resulting in the creation of 5class, 10class, 20class, and 50class datasets. Additionally, we repeated the negative sampling experiment five times and averaged the results to reduce experimental errors caused by random selection. Figure 3 shows that our method maintains a stable performance improvement even when the number of entity categories significantly increases, while LightNER's performance shows a noticeable decline under the 50class scenario.

Conclusion

In this paper, we propose PMRC, a prompt-based MRC paradigm for low-resource NER. PMRC enhances performance in extremely limited data scenarios by leveraging prompts that contain label words or examples. Additionally, PMRC demonstrates excellent knowledge transfer capabilities by assigning entity start and end markers from the prompts to each token in the input text. We conducted extensive experiments in various settings, and PMRC achieved competitive results in resource-rich environments while surpassing state-of-the-art performance in resource-constrained environments.

Acknowledgments

This work has been partially supported by Beijing University of Posts and Telecommunications Basic Research Fund with No. 2022RC12, and State Key Laboratory of Networking and Switching Technology with No. NST20220303.

References

Ben-David, E.; Oved, N.; and Reichart, R. 2022. PADA: Example-based Prompt Learning for on-the-fly Adaptation to Unseen Domains. *Trans. Assoc. Comput. Linguistics*, 10: 414–433.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Chen, J.; Lu, Y.; Lin, H.; Lou, J.; Jia, W.; Dai, D.; Wu, H.; Cao, B.; Han, X.; and Sun, L. 2023. Learning In-context Learning for Named Entity Recognition. *CoRR*, abs/2305.11038.

Chen, X.; Li, L.; Deng, S.; Tan, C.; Xu, C.; Huang, F.; Si, L.; Chen, H.; and Zhang, N. 2022a. LightNER: A Lightweight Tuning Paradigm for Low-resource NER via Pluggable Prompting. In Calzolari, N.; Huang, C.; Kim, H.; Pustejovsky, J.; Wanner, L.; Choi, K.; Ryu, P.; Chen, H.; Donatelli, L.; Ji, H.; Kurohashi, S.; Paggio, P.; Xue, N.; Kim, S.; Hahm, Y.; He, Z.; Lee, T. K.; Santus, E.; Bond, F.; and Na, S., eds., *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, 2374–2387. International Committee on Computational Linguistics.

Chen, X.; Zhang, N.; Xie, X.; Deng, S.; Yao, Y.; Tan, C.; Huang, F.; Si, L.; and Chen, H. 2022b. KnowPrompt: Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction. In Laforest, F.;

Troncy, R.; Simperl, E.; Agarwal, D.; Gionis, A.; Herman, I.; and Médini, L., eds., *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, 2778–2788. ACM.

Cui, L.; Wu, Y.; Liu, J.; Yang, S.; and Zhang, Y. 2021. Template-Based Named Entity Recognition Using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 1835–1845.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics.

Ding, N.; Chen, Y.; Han, X.; Xu, G.; Wang, X.; Xie, P.; Zheng, H.; Liu, Z.; Li, J.; and Kim, H. 2022. Prompt-learning for Fine-grained Entity Typing. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 6888–6901. Association for Computational Linguistics.

Fritzler, A.; Logacheva, V.; and Kretov, M. 2019. Few-shot classification in named entity recognition task. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 993–1000.

Gui, T.; Ye, J.; Zhang, Q.; Li, Z.; Fei, Z.; Gong, Y.; and Huang, X. 2020. Uncertainty-Aware Label Refinement for Sequence Labeling. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 2316–2326. Association for Computational Linguistics.

Hakkani-Tür, D.; Tür, G.; Celikyilmaz, A.; Chen, Y.; Gao, J.; Deng, L.; and Wang, Y. 2016. Multi-Domain Joint Semantic Frame Parsing Using Bi-Directional RNN-LSTM. In Morgan, N., ed., *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, 715–719. ISCA.

Henderson, M.; and Vulic, I. 2021. ConVEx: Data-Efficient and Few-Shot Slot Labeling. In Toutanova, K.; Rumshisky, A.; Zettlemoyer, L.; Hakkani-Tür, D.; Beltagy, I.; Bethard, S.; Cotterell, R.; Chakraborty, T.; and Zhou, Y., eds., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, 3375–3389. Association for Computational Linguistics.

Hou, Y.; Che, W.; Lai, Y.; Zhou, Z.; Liu, Y.; Liu, H.; and Liu, T. 2020. Few-shot Slot Tagging with Collapsed Dependency Transfer and Label-enhanced Task-adaptive Projection Network. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 1381–1393. Association for Computational Linguistics.

Huang, J.; Li, C.; Subudhi, K.; Jose, D.; Balakrishnan, S.; Chen, W.; Peng, B.; Gao, J.; and Han, J. 2020. Few-Shot Named Entity Recognition: A Comprehensive Study. *CoRR*, abs/2012.14978.

Huang, Y.; He, K.; Wang, Y.; Zhang, X.; Gong, T.; Mao, R.; and Li, C. 2022a. COPNER: Contrastive Learning with Prompt Guiding for Few-shot Named Entity Recognition. In Calzolari, N.; Huang, C.; Kim, H.; Pustejovsky, J.; Wanner, L.; Choi, K.; Ryu, P.; Chen, H.; Donatelli, L.; Ji, H.; Kurohashi, S.; Paggio, P.; Xue, N.; Kim, S.; Hahm, Y.; He, Z.; Lee, T. K.; Santus, E.; Bond, F.; and Na, S., eds., *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, 2515–2527. International Committee on Computational Linguistics.

Huang, Y.; Lei, W.; Fu, J.; and Lv, J. 2022b. Reconciliation of Pre-trained Models and Prototypical Neural Networks in Few-shot Named Entity Recognition. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 1793–1807. Association for Computational Linguistics.

Ji, B.; Li, S.; Gan, S.; Yu, J.; Ma, J.; Liu, H.; and Yang, J. 2022. Few-shot Named Entity Recognition with Entity-level Prototypical Network Enhanced by Dispersedly Distributed Prototypes. In Calzolari, N.; Huang, C.; Kim, H.; Pustejovsky, J.; Wanner, L.; Choi, K.; Ryu, P.; Chen, H.; Donatelli, L.; Ji, H.; Kurohashi, S.; Paggio, P.; Xue, N.; Kim, S.; Hahm, Y.; He, Z.; Lee, T. K.; Santus, E.; Bond, F.; and Na, S., eds., *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, 1842–1854. International Committee on Computational Linguistics.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 7871–7880. Association for Computational Linguistics.

Li, X.; Feng, J.; Meng, Y.; Han, Q.; Wu, F.; and Li, J. 2020. A Unified MRC Framework for Named Entity Recognition. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 5849–5859. Association for Computational Linguistics.

Li, X.; Yin, F.; Sun, Z.; Li, X.; Yuan, A.; Chai, D.; Zhou, M.; and Li, J. 2019. Entity-Relation Extraction as Multi-Turn Question Answering. In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 1340–1350. Association for Computational Linguistics.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. 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)*, 4582–4597. Online: Association for Computational Linguistics.

Liang, C.; Yu, Y.; Jiang, H.; Er, S.; Wang, R.; Zhao, T.; and Zhang, C. 2020. BOND: BERT-Assisted Open-Domain Named Entity Recognition with Distant Supervision. In Gupta, R.; Liu, Y.; Tang, J.; and Prakash, B. A., eds., *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, 1054–1064. ACM.

Lin, H.; Lu, Y.; Han, X.; and Sun, L. 2019. Sequence-to-Nuggets: Nested Entity Mention Detection via Anchor-Region Networks. In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 5182–5192. Association for Computational Linguistics.

Lin, T.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2017. Focal Loss for Dense Object Detection. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2999–3007. IEEE Computer Society.

Liu, A. T.; Xiao, W.; Zhu, H.; Zhang, D.; Li, S.-W.; and Arnold, A. 2022. QaNER: Prompting question answering models for few-shot named entity recognition. *arXiv preprint arXiv:2203.01543*.

Liu, J.; Pasupat, P.; Cyphers, S.; and Glass, J. R. 2013. Asgard: A portable architecture for multilingual dialogue systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, 8386–8390. IEEE.

Liu, K.; Fu, Y.; Tan, C.; Chen, M.; Zhang, N.; Huang, S.; and Gao, S. 2021. Noisy-Labeled NER with Confidence Estimation. In Toutanova, K.; Rumshisky, A.; Zettlemoyer, L.; Hakkani-Tür, D.; Beltagy, I.; Bethard, S.; Cotterell, R.; Chakraborty, T.; and Zhou, Y., eds., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, 3437–3445. Association for Computational Linguistics.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9): 1–35.

Liu, Y.; Meng, F.; Zhang, J.; Xu, J.; Chen, Y.; and Zhou, J. 2019. GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling. In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 2431–2441. Association for Computational Linguistics.

Lu, Y.; Liu, Q.; Dai, D.; Xiao, X.; Lin, H.; Han, X.; Sun, L.; and Wu, H. 2022. Unified Structure Generation for Universal Information Extraction. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 5755–5772. Association for Computational Linguistics.

Ma, J.; Ballesteros, M.; Doss, S.; Anubhai, R.; Mallya, S.; Al-Onaizan, Y.; and Roth, D. 2022a. Label Semantics for Few Shot Named Entity Recognition. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, 1956–1971. Association for Computational Linguistics.

Ma, R.; Zhou, X.; Gui, T.; Tan, Y.; Li, L.; Zhang, Q.; and Huang, X. 2022b. Template-free Prompt Tuning for Few-shot NER. In Carpuat, M.; de Marneffe, M.; and Ruíz, I. V. M., eds., *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, 5721–5732. Association for Computational Linguistics.

Ma, X.; and Hovy, E. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1064–1074. Berlin, Germany: Association for Computational Linguistics.

Sang, E. F. T. K.; and Meulder, F. D. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In Daelemans, W.; and Osborne, M., eds., *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, 142–147. ACL.

Schick, T.; Schmid, H.; and Schütze, H. 2020. Automatically Identifying Words That Can Serve as Labels for Few-Shot Text Classification. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Schick, T.; and Schütze, H. 2021. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In Merlo, P.; Tiedemann, J.; and Tsarfaty, R., eds., *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, 255–269. Association for Computational Linguistics.

Shrimal, A.; Jain, A.; Mehta, K.; and Yenigalla, P. 2022. NER-MQMRC: Formulating Named Entity Recognition as Multi Question Machine Reading Comprehension. In *Proceedings of the 2022 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track, 230–238. Hybrid: Seattle, Washington + Online: Association for Computational Linguistics.

Tjong Kim Sang, E. F.; and De Meulder, F. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 142–147.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

Wiseman, S.; and Stratos, K. 2019. Label-Agnostic Sequence Labeling by Copying Nearest Neighbors. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5363–5369. Florence, Italy: Association for Computational Linguistics.

Yamada, I.; Asai, A.; Shindo, H.; Takeda, H.; and Matsumoto, Y. 2020. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 6442–6454. Association for Computational Linguistics.

Yang, J.; Liang, S.; and Zhang, Y. 2018. Design Challenges and Misconceptions in Neural Sequence Labeling. In Bender, E. M.; Derczynski, L.; and Isabelle, P., eds., *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, 3879–3889. Association for Computational Linguistics.

Yang, Y.; and Katiyar, A. 2020. Simple and Effective Few-Shot Named Entity Recognition with Structured Nearest Neighbor Learning. In Webber, B.; Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 6365–6375. Association for Computational Linguistics.

Yu, J.; Bohnet, B.; and Poesio, M. 2020. Named Entity Recognition as Dependency Parsing. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 6470–6476. Association for Computational Linguistics.

Zhang, N.; Deng, S.; Bi, Z.; Yu, H.; Yang, J.; Chen, M.; Huang, F.; Zhang, W.; and Chen, H. 2020. OpenUE: An Open Toolkit of Universal Extraction from Text. In Liu, Q.; and Schlangen, D., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, 1–8. Association for Computational Linguistics.

Ziyadi, M.; Sun, Y.; Goswami, A.; Huang, J.; and Chen, W. 2020. Example-Based Named Entity Recognition. *CoRR*, abs/2008.10570.