RESEARCH PAPER

# Leveraging Continuous Prompt for Few-Shot Named Entity Recognition in Electric Power Domain with Meta-Learning

**Yang Yu, Wei He[†], Yu-meng Kang & You-lang Ji**

Metrology Center, State Grid Jiangsu Market Service Center Nanjing Jiangsu 210019, China

**Keywords:** Named Entity Recognition; Pre-trained model; Prompt-Tuning; Meta-Learning; Few-Shot

Citation: Yu, Y., He, W., Kang, Y.-M., et al.: Leveraging continuous prompt for Few-Shot Named Entity Recognition in electric power domain with meta-Learning. Data Intelligence 5(2), 494-509 (2023). doi: 10.1162/dint_a_00202
Submitted: October 2, 2022; Revised: December 12, 2022; Accepted: January 3, 2023

## ABSTRACT

Conventional named entity recognition methods usually assume that the model can be trained with sufficient annotated data to obtain good recognition results. However, in Chinese named entity recognition in the electric power domain, existing methods still face the challenges of lack of annotated data and new entities of unseen types. To address these challenges, this paper proposes a meta-learning-based continuous cue adjustment method. A generative pre-trained language model is used so that it does not change its own model structure when dealing with new entity types. To guide the pre-trained model to make full use of its own latent knowledge, a vector of learnable parameters is set as a cue to compensate for the lack of training data. In order to further improve the model's few-shot learning capability, a meta-learning strategy is used to train the model. Experimental results show that the proposed approach achieves the best results in a few-shot electric Chinese power named entity recognition dataset compared to several traditional named entity approaches.

## 1. INTRODUCTION

In recent years, China's electric power industry has developed rapidly and the scale of the industry has grown significantly. Under the influence of 5G, Internet of Things, and other high-tech, China's power industry has entered a new period of transformation and upgrading. The deep organic combination of

[†] Corresponding author: Wei He (E-mail: 562691978@qq.com; ORCID: 0000-0002-7993-4441).

cutting-edge artificial intelligence technology and traditional technology in the electric power industry has achieved good benefits. In the process of accelerating the production efficiency and economic transformation of the electric power industry, the knowledge graph has become an important way of electric power intelligence. In the whole construction process, named entity recognition (NER) of electric power text data is a fundamental part, and its purpose is to identify words or phrases in the text as different categories of entity labels [1], which provides the basis for the subsequent steps.

Pre-trained language models (PLM) [2] have been widely used for NER tasks. Existing PLM-based studies mainly consider NER as a sequentially labeled task with a label-specific CRF output layer added on top of the PLM representation. Although these approaches have achieved good results on open domain datasets, they still face the following two challenges in the Chinese electric power domain: (1) Lack of training instances. Due to the high cost of labeling, for an entity type, models often rely on less than one hundred instances for training. This few-shot scenario is a serious challenge to the generalization ability of the model. (2) Unseen new entity types. New entity types are very common due to the frequent expansion of business scenarios. Since the CRF output layers of existing methods are fixed, they have to retrain the models to predict new entity types. This not only does not meet the efficiency requirements of the electric power industry but also does not guarantee good recognition.

To address the above challenges, this paper proposes a Chinese NER model (Meta Continuous Prompt NER model, MCP-NER) based on meta-learning and continuous prompt-tuning. The model is based on a generative PLM in order to directly predict new types of electric power entities without modifying the model structure. To make up for the lack of training instances, MCP-NER applies prompt-tuning to mine the potential prior knowledge in PLM. Different from previous works using the manual prompt template[9], MCP-NER sets continuous prompt vectors to direct the attention flow of the transformer to avoid optimizing the entire parameters of the PLM, thus greatly improving the training efficiency. To further improve the generalization capability on unseen types, MCP-NER is trained by a meta-learning algorithm. It enforces the model to capture the common feature of different entity types by correcting the gradient between the support set and query set. We constructed a Chinese electric power NER dataset using real national grid data and conducted a comprehensive experiment using it. The experimental results show that our MCP-NER method outperforms all comparative methods and achieves state-of-the-art performance in few-shot NER scenarios.

The rest of this paper is organized as follows. In Section 2, we will introduce some related work of few-shot NER. In Section 3, we will formulate the few-shot NER problem. In Section 4, we will detail our proposed MCP-NER and its motivation. In Section 5, we will introduce our experimental setup and results. In Section 6, we will present our conclusions and look ahead to future work.

## 2. RELATED WORK

There are two important lines of research for few-shot NER: PLM-based approaches and meta-learning-based approaches.

### 2.1 PLM-based Approaches

PLM has recently had a significant impact on NER [2, 3], where Transformer-based models [2] are used as the core network structure, gaining widespread attention. The current mainstream methods [4, 5, 6] all regard NER as a sequence labeling problem. Yan et al. [5] proposed the Unified-NER model, which formulates the NER subtask as an entity-span generation task, leveraging hand-crafted one-to-one mappings from tokens in the vocabulary to entity types; however, it fails to generate complex entity types that correspond to, such as business requirements and fault exceptions in the electric power domain.

Prompt-based optimization methods [3, 4] have received a great deal of attention since the advent of GPT-3. GPT-3 shows that large-scale language models can achieve superior performance in few-shot scenarios through fine-tuning and context learning. Schick et al. [7] argue that small-scale language models can also achieve good performance using prompt optimization. While most research has been done on text classification tasks, some work extends the impact of prompt optimization to other tasks, such as relation extraction. In addition to various downstream tasks, cues are used to probe knowledge from PLMs and are thus a promising learning paradigm [15, 16, 17, 18, 19, 33, 34].

Recently, Cui et al. [9] proposed a prompt-based Template-BART model for few-shot NER, which enumerates all possible spans in a sentence and populates them into hand-crafted templates. Then the output of the model is to classify each candidate entity span according to the corresponding template score. Unlike their approach, the MCP-NER model proposed in this paper does not rely on template engineering but uses learnable prompt embeddings to guide attention to guide knowledge in PLM more efficiently. In addition, the MCP-NER model only updates lightweight prompt vectors during training, which greatly improves the efficiency of PLM-based methods. Furthermore, some works in related fields have used soft prompts [28, 29, 30, 31, 32] achieve the good performance. Inspired by them, we apply this technique to the NER task.

### 2.2 Meta-learning-based Approaches

Meta-learning has become a popular method in research on few-shot learning. This type of method [11, 12] utilizes the nearest neighbor criterion to assign entity types. Typically, this depends on the similarity patterns of entities between the source and target domains without updating the network parameters for the NER task. However, this also makes them unable to improve neural representations of cross-domain instances. Other studies have utilized transfer learning methods [13, 14] for knowledge transfer across languages or domains to enhance few-shot learning. MetaNER [20] incorporates meta-learning and adversarial training strategies to encourage robust, general, and transferable representations for sequence labeling. de et al. [21] propose a task generation scheme for converting classical NER datasets into the few-shot setting, for both training and evaluation. Ma et al. [22] proposes MAML-enhanced prototypical networks to find a good embedding space that can better distinguish text span representations from different entity classes. Li et al. [23] propose FewNER separates the entire network into a task-independent part and a task-specific part, in order to cope with them separately. Unlike them, our approach builds on prompt-tuning, where the operations in meta-learning are applied to prompt vectors, which are more lightweight and flexible.

## 3. PROBLEM FORMULATION

Given an input text sequence $X = \{x^0, x^1, \ldots, x^{n-1}\}$, the goal of NER is to output a set $\mathrm{E} = \{\mathrm{e}^1, \mathrm{e}^2, \ldots, e^m\}$, where $n$ is the total number of words, $m$ is the number of recognized entities, $x^i$ represents the $i$-th word in the text, and $e^i = (u^i_{start}, u^i_{end}, c^i)$ represents the recognized $i$-th entity. Here, $u^i_{start} \in [1, n]$ and $u^i_{end} \in [u^i_{start}, n]$ represent the start index and end index referred to $e^i$ in $X$ respectively, $c^i \in C$ represent the type label of $e^i$. $C$ is the set of all type labels. The training dataset for NER often consists of pair-wise data $D_{train} = \{(X^1, E^1), \ldots, (X^l, E^l)\}$, where $l$ is the number of training instances. Traditional NER systems are trained in the standard supervised learning paradigms, which usually require a large number of pairwise examples, i.e., $l$ is large. In real-world applications, the more favorable scenarios are that only a small number of labeled examples are given for each entity type ($l$ is small) because expanding labeled data increases annotation cost and decreases customer engagement. This yields a challenging task few-shot NER.

## 4. THE PROPOSED APPROACH

### 4.1 Modeling Process

In order to enable the model to deal with unseen entity types during testing without changing its own structure. For sequence $X = \{x^0, x^1, \ldots, x^{n-1}\}$, we first convert the original target $E$ into sequence $T = \{t^0, t^1, \ldots, t^{r-1}\}$, where $r$ is the total number of indexes, each index $t^i$ corresponds to a unique token $\tilde{t}^i$ :

$$\tilde{t}^i = \begin{cases} X^{t^i}, & 0 < t^i < n \\ c^{t^i - n}, & n \le t^i < n + |C| \end{cases} \tag{1}$$

Here $C$ denotes the set of all entity types. When $0 < t^i < n$, $\tilde{t}^i$ is the $t^i$th word in the input text $X$, which is used to form the identified entity mention; when $n \le t^i < n + |C|$, $\tilde{t}^i$ is the $t^i - n$th entity type label $c^{t^i - n}$, which is the type corresponding to the identified entity mention; suppose $X$ = {已,检,查,电,表,无,故,障,何,时,可,以,复,电} (The energy meter has been checked. When can the electric power supply be restored) which contains two entities: "电表 (Energy meter)" and "复电 (Restoring electric power supply)", then the target index sequence is $T = \{3,4,14,12,13,17\}$. Among them, 3 and 4 represent the indexes of the two words "电" (electric power) and "表" (meter) in $X$ respectively, 14 represents the sum of the index 0 in $C$ and the text length $n = 14$ of the entity type [machine equipment] of "电表 (Energy meter)"; 12 and 13 respectively Indicate the index of "复" (restore) and "电" (electric power) in $X$, and 17 represents the sum of the index 3 in $C$ and the text length $n = 14$ of the type "复电 (Restoring electric power supply)" [Business requirements].

For any input $X$, its corresponding generation probability of $T$ can be modeled as:

$$P(T \mid X)=\prod_{i=1}^{l} P(t^{i} \mid X, t^{0}, t^{1}, \ldots, t^{i-1}) \tag{2}$$

where $P(t^i \mid X, t^0, t^1, \ldots, t^{i-1})$ represents the probability of generating $t^i$ by combining $X$ and the generated $t^0, t^1, \ldots, t^{i-1}$.

The structure of the MCP-NER model proposed in this paper is shown in Figure 1. It uses the PLM model BART to model $P(t^i \mid X, t^0, t^1, \ldots, t^{i-1})$, which consists of an Encoder and a Decoder, and autoregressively generates an index sequence $T$, as shown below:

$$\boldsymbol{H}_{seq}=\text{Encoder}(X), \boldsymbol{H}_{type}=\text{Encoder}(C) \tag{3}$$

$$\boldsymbol{h}^{i}=\text{Decoder}(\boldsymbol{H}_{seq}, \tilde{t}^{0}, \tilde{t}^{1}, \ldots, \tilde{t}^{i-1}) \tag{4}$$

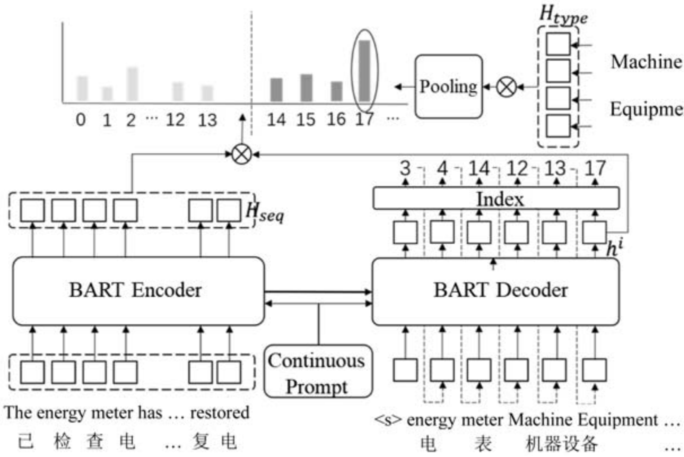

**Figure 1.** Architecture of the proposed MCP-NER. NER is modeled as a generation task, and the span position of the entity and the corresponding entity type are generated through the vector output by the decoder at each moment. The soft cue vector is added to the encoding and decoding process of BART to guide the attention flow in the transformer.

Among them, $\mathbf{H}_{seq} \in \mathbb{R}^{n \times d}$ and $\mathbf{H}_{type} \in \mathbb{R}^{|C| \times d}$ represent the semantic matrix of $X$ and $C$ respectively, and $\mathbf{h}^i \in \mathbb{R}^d$ is the hidden state of the moment $i$ in the decoding process, which is used to calculate the probability distribution $P^i$ of the index $t^i$:

$$P^{i}=\text{softmax}([P_{seq}^{i}; P_{type}^{i}]) \tag{5}$$

$$P_{seq}^{i}=\boldsymbol{H}_{seq} \otimes \boldsymbol{h}^{i}, P_{type}^{i}=\boldsymbol{H}_{type} \otimes \boldsymbol{h}^{i} \tag{6}$$

Among them, $P_{seq}^{i} \in \mathbb{R}^{n}$ and $P_{type}^{i} \in \mathbb{R}^{|C|}$ represent the probability distribution vectors of entity reference and entity type generated at moment $i$, respectively. $[P_{seq}^{i}; P_{type}^{i}]$ represents the concatenation of vectors $P_{seq}^{i}$ and $P_{type}^{i}$ to obtain the overall probability distribution $P^i \in \mathbb{R}^{n+|C|}$ as output.

Since there is no output layer with a fixed number of categories, for the new entity type $c'$ in the prediction process, only need to add its label to the set $C$, MCP-NER can calculate the new probability distribution $P^i$, and avoid the retraining caused by adjusting the output layer .

### 4.2 Continuous Prompt Optimization

The PLM model contains prior knowledge learned from massive corpora, which can effectively make up for the lack of training data. However, the huge number of parameters in PLM makes its training time too long and the optimization efficiency is low. In order to use PLM to alleviate the few-shot challenge in Chinese electric power NER while ensuring the training efficiency of the model, MCP-NER proposes to incorporate vectorized prompt into the self-attention layer of the BART model to guide the distribution of attention. During model optimization, only these prompt vectors are updated, not the parameters of the PLM.

Specifically, MCP-NER provides two sets of trainable parameters $\phi$ ={$\phi^1$, $\phi^2$, …, $\phi^N$} for the encoder and the decoder respectively, where $\phi^i = [\phi^i_K ; \phi^i_V] \in \mathbb{R}^{2\times Z\times d}$ , and $N$ represents the number of layers of the encoder or decoder, and $Z$ is an adjustable hyperparameter representing the number of prompt vectors in each layer (set to 10 in the experiment). In fact, some related works have proposed other ways to apply continuous prompt, such as using prompt as a prefix or suffix for the input word vector of the encoder. Here we only catenate the soft embeddings at the end of K and V because we want the continuous prompt to be directly involved in the attention stream of the encoder and decoder to achieve a more fine-grained prompt.

As shown in Figure 2, the input vector sequence $\mathbf{X}^i \in \mathbb{R}^d$ of the $i$th layer is first projected into three dimensions: query, key, and value.

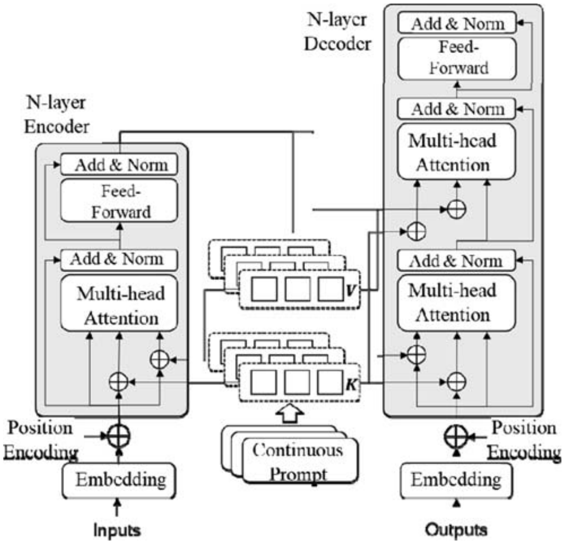

**Figure 2.** Architecture of our proposed continuous prompt optimization.

$$\boldsymbol{Q}^i = \boldsymbol{X}^i\boldsymbol{W}_Q, \boldsymbol{K}^i = \boldsymbol{X}^i\boldsymbol{W}_K, \boldsymbol{V}^i = \boldsymbol{X}^i\boldsymbol{W}_V \tag{7}$$

$\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V \in \mathbb{R}^{d\times d}$ is the original parameter matrix in BART, which is fixed during training. Subsequently, MCP-NER modifies the attention mechanism and introduces prompt vectors as shown below.

$$Attention^i = \text{softmax}(\frac{\boldsymbol{Q}^i[\boldsymbol{K}^i;\phi_K^i]^T}{\sqrt{d}})[\boldsymbol{V}^i;\phi_V^i] \tag{8}$$

Among them, $\phi_K^i \in \mathbb{R}^{Z\times d}$ and $\phi_V^i \in \mathbb{R}^{Z\times d}$ represents the parameter vector added to the key and value respectively. These vectors serve as prompt, concatenated with $\mathbf{K}^i$ and $\mathbf{V}^i$ to participate in the attention mechanism. MCP-NER aggregates and computes attention scores to guide the final attention flow.

In the experiment, the parameter size of the prompt vector is much smaller than that of the whole BART, so the model training time can be greatly shortened. At the same time, since these prompts are deeply involved in the attention flow, the prior knowledge in the BART parameters can be fully utilized to ensure the model's few-shot learning ability.

### 4.3 Meta-learning-based Training

In order to further improve the few-shot learning ability of the model and enable it to quickly adapt to new entity types that have not been seen during training, this paper proposes a Meta-Learning strategy to train MCP-NER.

First, the few-shot NER training set is divided into several tasks, where each task $G$ consists of several training samples $\{(X_1, E_1), \ldots, (X_{|G|}, E_{|G|})\}$. The samples in the same task cover N entity classes, and each entity class has K samples. During training, these samples are divided into two sets, called support set $S$ and query set $Q$ respectively. Subsequently, for each $G$, MCP-NER updates its prompt vector parameter $\phi$ through the following process:

1) Input each sample in the support set $S$ into the model $M_\phi$, calculate the support loss $L_S(M_\phi(X),T)$, and then get the gradient of the prompt parameter $\nabla_\phi L_S$;
2) Use the gradient descent method to get the temporary prompt parameter $\theta = \phi - \alpha\nabla_\phi L_S$, where $\alpha$ is the learning rate;
3) Input each sample in the query set $Q$ into the model, and calculate the query loss $L_Q(M_\theta(X),T)$;
4) Combine the loss of the support set and the query set to get the overall loss $L_T = \sigma L_S + (1-\sigma)L_Q$, where $\sigma$ is the weight hyperparameter;
5) Use the SGD optimization algorithm and the learning rate $\beta$ to optimize the original prompt parameters $\phi$ to get the final prompt parameters $\phi^* = \phi - \beta\nabla_\phi L_T$.

Here, $M_\phi(X)$ represents the predicted output of the MCP-NER model for the input sequence $X$, $T$ is the corresponding target output, and all loss functions $L$ use cross-entropy loss. In order for the model to have the ability to recognize new types of entities, the training process ensures that the entity types in the support set $S$ and query set $Q$ are disjoint to simulate this situation. According to this split, the model undergoes a two-stage gradient update during the training process for each task. The first stage is to make the model fit the entity type in $S$, and get the optimal parameters $\theta$ on $S$. This parameter is used as a possible optimization direction, but the original parameters are not updated at this time. In the second stage, $Q$ is used to rectify $\theta$, by optimizing the joint loss $L_T$ to make the model focus on common features in $Q$ and $S$ to adapt to new entity types not seen in $Q$. Finally get updated $\phi^*$, which is better at capturing common features in different entity types, so that the sample size can be smaller or even missing.

## 5. EXPERIMENTS AND ANALYSIS

### 5.1 Experimental Environment

The hardware environment of the experiment in this paper: the CPU adopts Intel® Core 7700, the memory is 32GB, and the GPU adopts Nvidia RTX-2080ti. Software environment: ubuntu 18.04, Python 3.6.8, deep learning framework adopts Pytorch 1.4.0.

### 5.2 Datasets

In this article, we collected data of 18,000 work orders from the customer service of State Grid Jiangsu Company in China between May and June 2021. We hired outsourcers to label the "User Acceptance Content" part for all of the data according to the pre-defined NER specification, and finally constructed the named entity recognition dataset ENER. In this dataset, we define the following 13 types of entities:

**Table 1.** Types of entities in ENER.

| Dataset | Types of entities |
|---|---|
| ENER | Machine equipment, Electricity Price, Business Demand, Fault Exception, Financial Bill, Electronic Channel, User Information, Document Regulation, Marketing Activity, Identity, Company, Illegal Act, Professional Vocabulary |

The training set, validation set and test set contain 1,358, 1,059 and 2,032 instances, respectively. For each entity type, the number of the training instances is less than 100.

In addition, we use the CoNLL 2003 [27] as an open-domain dataset to evaluate the performance of our model on conventional benchmarks.

### 5.3 Evaluation Metrics

In this paper, the accuracy rate (P), recall rate (R) and $F_1$ score ($F_1$) are used as the evaluation indicators of model performance to evaluate the entity recognition results. The calculation methods are:

$$P = T_{TP} / (T_{TP} + F_{TP}) \times 100\% \tag{9}$$

$$R = T_{TP} / (T_{TP} + F_{FN}) \times 100\% \tag{10}$$

$$F1 = 2PR / (P + R) \times 100\% \tag{11}$$

Among them, $T_{TP}$ represents the number of entities correctly recognized by the model; $F_{TP}$ represents the number of entities recognized but actually wrong; $F_{FN}$ represents the number of entities that are actually correct but not recognized by the model.

### 5.4 Compared Methods and Ablation Settings

In this paper, several tradtional baseline models commonly used in NER are used as the comparison methods for MCP-NER: BiGRU, BiLSTM-CNN, and BiLSTM-CRF. These methods use a recurrent neural network (RNN) as the encoder and a CRF or convolutional neural network (CNN) as the output classifier. In addition, this paper compares with PLM-based methods that have performed well in recent years: Sequence Labeling-BERT (SL-BERT) [2], Sequence Labeling-BART (SL-BART) [24]. We also compare our method with few-shot learning methods: FewNER [25], Model Fusion [26]. They utilize a linear layer to classify the hidden vectors at each input location, resulting in NER labels. Finally, this paper compares with the latest prompt template optimization method Template-BART [9]. This method applies templated-based prompt to utilize the PLM.

In order to test the contribution of continuous prompt optimization and meta-learning training strategy in MCP-NER to the overall effect, this paper designs the following two groups of ablation experiments: (1) MCP-NER (w/o prompt): The prompt vector of the model species is deleted, and only the Fine-tune BART generation results, at which point all parameters in BART are no longer fixed. (2) MCP-NER (input prefix prompt): The prompt vector is directly prefixed to the encoder's input, not at the end of K and V. (3) MCP-NER (w/o ML): Use the Mini-Batch strategy commonly used in machine learning instead of the meta-learning strategy to train the model.

### 5.5 Overall Experimental Results

First, all methods are trained using the full training set of ENER. The experimental results are shown in Table 2. Among them, SL-BERT and SL-BART, as strong baselines, show good results. Nonetheless, MCP-NER still achieves the best results, demonstrating the effectiveness of this paper in handling few-shot scenarios. Furthermore, although both MCP-NER and Template-BART are based on BART, the former achieves (2.42% vs. 0.82%) and (3.47% vs. 2.12%) higher F1 scores than these two methods, respectively, which proves that continuous prompt are more effective than prompt templates. Correspondingly, ablation experiments also demonstrate the contribution of prompt optimization and meta-learning strategies. In contrast, meta-learning improves the ability of few-shot learning even more.

**Table 2.** Results by all models.

| Model | ENER | | | CoNLL-2003 | | |
|---|---|---|---|---|---|---|
| | P (%) | R (%) | $F_1$ (%) | P (%) | R (%) | $F_1$ (%) |
| BiGRU | 62.15 | 68.65 | 65.28 | 89.71 | 90.97 | 90.24 |
| BiLSTM-CNN | 62.78 | 69.99 | 66.45 | 91.89 | 91.48 | 91.62 |
| BiLSTM-CRF | 63.92 | 70.13 | 66.94 | 92.05 | 91.62 | 91.78 |
| SL-BERT | 66.37 | 72.67 | 69.01 | 91.93 | 91.54 | 91.73 |
| SL-BART | 65.56 | 73.20 | 68.98 | 89.60 | 91.63 | 90.60 |
| Template-BART | 65.15 | 74.28 | 70.03 | 90.51 | **93.34** | 91.90 |
| FewNER | 64.82 | 73.78 | 69.54 | 90.94 | 92.59 | 91.32 |
| Model-Fusion | 64.18 | 73.92 | 69.37 | 89.91 | 91.89 | 90.47 |
| MCP-NER | **68.03** | **75.34** | **72.45** | **92.48** | 93.08 | **92.72** |
| w/o prompt | 67.41 | 74.11 | 71.28 | 91.87 | 92.42 | 92.05 |
| input prefix prompt | 67.32 | 74.47 | 71.39 | 92.03 | 92.79 | 92.49 |
| w/o ML | 65.69 | 72.26 | 69.59 | 91.32 | 91.97 | 91.66 |

Note: Here P, R, and F1 denote Precision, Recall, and F1 score, respectively.

### *5.6 Ablation Study on Training Time*

Figure 3 lists the training time of MCP-NER and several typical comparison methods when using the full training set of ENER. Among them, the abscissa is the comparison method, and the ordinate is the training time (minutes). To ensure fairness, all methods are trained until the model converges. PLM-based methods require a long training time due to their huge amount of parameters. Among them, Template-BART takes the longest because it also needs to enumerate entity spans and entity types. On the contrary, the MCP-NER in this paper only needs to optimize a small part of the prompt parameters, thus greatly reducing the time overhead of training, even close to the BiLSTM-CRF model that does not use a pre-trained model.

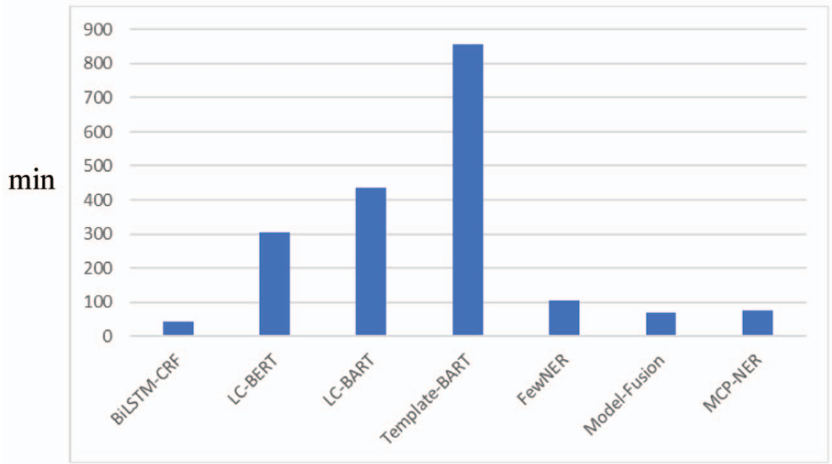

**Figure 3.** Comparison of training time.

### 5.7 Experiment Results of Different Shots

In order to further explore the influence of the number of samples on MCP-NER, for each entity type, {10, 20, 50} training instances were randomly sampled as $D_{10}$, $D_{20}$ and $D_{50}$ from the original training set of ENER respectively. Following the N-way-K-shot settings, we also change the ways (i.e., the number of the entity types) for different settings of the shots. Train on these three subsets and test the effect on the original test set of ENER. The experimental results are shown in Table 3. The fewer training samples, the more obvious the improvement of MCP-NER compared to other methods. This is because prompt optimization aligns PLM pre-training with the goal of NER, which can guide rich prior knowledge to cope with few-shots. In addition, when the training samples are small enough, the improvement brought by meta-learning is very obvious, which fully proves its effectiveness.

**Table 3.** Results of different shots.

| Model | 5-Way | | | 10-Way | | |
|---|---|---|---|---|---|---|
| | 10 | 20 | 50 | 10 | 20 | 50 |
| SL-BERT | 21.32 | 29.99 | 40.06 | 31.97 | 41.25 | 62.81 |
| SL-BART | 18.45 | 27.31 | 39.08 | 32.23 | 39.78 | 60.17 |
| Template-BART | 30.82 | 45.32 | 53.29 | 40.51 | 54.39 | 65.28 |
| FewNER | 29.74 | 43.78 | 54.13 | 39.79 | 53.85 | 64.28 |
| Model-Fusion | 30.17 | 44.02 | 53.73 | 41.32 | 53.81 | 63.92 |
| MCP-NER | **40.25** | **48.20** | **58.12** | **51.22** | **62.79** | **69.53** |
| w/o prompt | 38.73 | 46.20 | 55.03 | 48.23 | 60.08 | 67.35 |
| input prefix prompt | 39.44 | 47.23 | 55.98 | 49.47 | 61.36 | 67.98 |
| w/o ML | 36.84 | 48.14 | 57.24 | 48.14 | 59.07 | 66.27 |

### 5.8 Recognition Performance on New Types of Entities

In order to explore the transferability of MCP-NER's recognition ability to unseen entities in the training set, this paper samples from the original training set of ENER and constructs a source domain training set $D_S$, and three target domain test sets $D_T^1$, $D_T^2$, and $D_T^3$. Among them, $D_S$ contains all the training samples of the following 10 entity types: Financial Bills, Electronic Channels, User Information, Document Regulations, Marketing Activities, Identity, Company, Illegal Acts, Professional Vocabulary, Electricity price; while $D_T^1$, $D_T^2$, and $D_T^3$ contain test data of types Machine Equipment, Business Requirements and Fault Exceptions, respectively, which are used to simulate the data of new types of entities. In this experiment, all models are trained on $D_S$, and then tested on the test sets of $D_L^1$, $D_L^2$, and $D_L^3$ respectively.

The experimental results are shown in Table 4. The MCP-NER model has the best performance in recognizing entities of unseen types during training, validating the effectiveness of the generative framework and meta-learning approach. Among them, the effect is the best on the Business Requirement type, which may be because the Business Requirement is closer to the entity type in the source domain, and the domain of Machine Equipment and Fault Exception are more different, so the improvement is relatively small. In addition, regardless of the entity type, MCP-NER(-prompt) and MCP-NER(-weight) always have a left-right effect gap compared with the overall model, which proves that continuous prompt optimization and meta-learning strategy are also effective in cross-type transfer learning.

**Table 4.** Results of cross-domain test.

| Model | Target domain | | |
|---|---|---|---|
| | Machine equipment | Business requirements | Fault exceptions |
| SL-BERT | 45.25 | 48.44 | 47.25 |
| SL-BART | 44.09 | 49.18 | 45.39 |
| Template-BART | 52.94 | 58.30 | 59.94 |
| MCP-NER | **53.85** | **62.91** | **64.83** |
| w/o prompt | 49.21 | 57.69 | 60.15 |
| input prefix prompt | 48.16 | 58.35 | 59.01 |
| w/o ML | 50.78 | 60.12 | 62.37 |

## 6. CONCLUSION AND FUTURE WORK

In this paper, a method based on meta-learning and continuous prompt optimization is proposed for the few-shot NER task in the Chinese electric power domain. Different from traditional approaches based on PLM models, this method uses trainable parameter vectors as prompts instead of fine-tuning the entire model.

The entire model is based on pre-trained BART and uses autoregressive decoding to generate entities and corresponding types. Only parameter vectors are added as prompts between the encoder and decoder to guide the optimization of the entire model. Furthermore, to cope with new types of entities that have not been seen before, the model is trained with a meta-learning-based strategy.

Experiments show that, our method achieves the best results in few-shot scenarios. Moreover, as the number of samples decreases, the improvement of the proposed method is more significant than that of the comparison method. In future work, we consider combining in-context learning with prompt-tuning, using the training data as a direct prompt.

## ACKNOWLEDGEMENT

This work is supported by the Science and Technology Project (No.J2021151) of State Grid Jiangsu Co., Ltd.

## AUTHOR CONTRIBUTION STATEMENT

Yang Yu (nuaa20322@126.com, 0000-0003-4180-7970) was responsible for researching and proposing the overall methodology, conducting the main experiments, and writing the article.

Wei He (562691978@qq.com, 0000-0002-7993-4441) assisted in some of the experiments and was responsible for writing and revising the article.

Yu-meng Kang (919804529@qq.com, 0000-0003-3538-0085) assisted in some of the experiments, and was responsible for statistical analysis and article writing.

You-lang Ji (jyl_sq@163.com, 0000-0002-2736-3317) assisted the experiments, and was responsible for writing and revising the article.

## REFERENCES

[1] Sang, E.T.K., De Meulder, F.: Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In: Proceedings NAACL, pp. 142–147 (2003)

[2] Devlin, J., Chang, M.W., Lee, K., et al.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings NAACL, pp. 4171–4186 (2019)

[3] Zheng, H., Wen, R., Chen, X., et al.: PRGC: Potential Relation and Global Correspondence Based Joint Relational Triple Extraction. In: Proceedings ACL, pp. 6225–6235 (2021)

[4] Liu, Y., Meng, F., Zhang, J., et al.: GCDT: A Global Context Enhanced Deep Transition Architecture for Sequence Labeling. In: Proceedings ACL, pp. 2431–2441 (2019)

[5] Zhang, N., Deng, S., Bi, Z., et al.: OpenUE: An Open Toolkit of Universal Extraction from Text. In: Proceedings EMNLP, pp. 1–8 (2020)

[6] Brown, T.B., Mann, B., Ryder, N., et al.: Language Models are Few-Shot Learners. In: arXiv preprint arXiv:2005.14165 (2020)

[7] Schick, T., Schmid, H., Schütze, H.: Automatically Identifying Words That Can Serve as Labels for Few-Shot Text Classification. In: Proceedings COLING, pp. 5569–5578 (2020)

[8] Shin, T., Razeghi, Y., Logan IV, R.L., et al.: AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In: Proceedings EMNLP, pp. 4222–4235 (2020)

[9] Cui, L., Wu, Y., Liu, J., Yang, S., et al.: Template-Based Named Entity Recognition Using BART. In: Findings ACL-IJCNLP 2021, pp. 1835–1845 (2021)

[10] Yan, H., Gui, T., Dai, J., et al.: A Unified Generative Framework for Various NER Subtasks. In: Proceedings ACL, pp. 5808–5822 (2021)

[11] Fritzler, A., Logacheva, V., Kretov, M.: Few-shot classification in named entity recognition task. In: Proceedings ACM/SIGAPP, pp. 993–1000 (2019)

[12] Wiseman, S., Stratos, K.: Label-Agnostic Sequence Labeling by Copying Nearest Neighbors. In: Proceedings ACL, pp. 5363–5369 (2019)

[13] Bao, Z., Huang, R., Li, C., et al.: Low-Resource Sequence Labeling via Unsupervised Multilingual Contextualized Representations. In: Proceedings EMNLP, pp. 1028–1039 (2019)

[14] Wang, Y., Mukherjee, S., Chu, H., et al.: Meta self-training for few-shot neural sequence labeling. In: Proceedings ACM SIGKDD, pp. 1737–1747 (2021)

[15] Zhang, Y., Chen, H., Zhao, Y., et al.: Learning tag dependencies for sequence tagging. In: Proceedings IJCAI, pp. 4581–4587 (2018)

[16] Cui, L., Zhang, Y.: Hierarchically-Refined Label Attention Network for Sequence Labeling. In: Proceedings EMNLP, pp. 4115–4128 (2019)

[17] Chiu, J.P., Nichols, E.: Named Entity Recognition with Bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics 4(1), 357–370 (2016)

[18] Huang, Y., He, K., Wang, Y., et al.: Copner: Contrastive learning with prompt guiding for few-shot named entity recognition. In: Proceedings COLING, pp. 2515–2527 (2022)

[19] Chen, J., Hu, Y., Liu, J., et al.: Deep short text classification with knowledge powered attention. In: Proceedings AAAI, pp. 6252–6259 (2019)

[20] Li, J., Shang, S., Shao, L.: Metaner: Named entity recognition with meta-learning. In: Proceedings of The Web Conference, pp. 429–440 (2020)

[21] De Lichy, C., Glaude, H., Campbell, W. Meta-learning for few-shot named entity recognition. In: Proceedings ACL, pp. 44–58 (2021)

[22] Ma, T., Jiang, H., Wu, Q., et al.: Decomposed Meta-Learning for Few-Shot Named Entity Recognition. In: Findings ACL, pp. 1584–1596 (2022)

[23] Li, X., Li, Z., Zhang, Z., et al.: Effective Few-Shot Named Entity Linking by Meta-Learning. In: Proceedings ICDE, pp. 178–191 (2022)

[24] Lewis, M., Liu, Y., Goyal, N., et al.: BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In: Proceedings ACL, pp. 7871–7880 (2020)

[25] Das, S.S.S., Katiyar, A., Passonneau, R.J., et al.: CONTaiNER: Few-Shot Named Entity Recognition via Contrastive Learning. In: Proceedings ACL, pp. 6338–6353 (2022)

[26] Gong, Y., Mao, L., Li, C.: Few-shot learning for named entity recognition based on BERT and two-level model fusion. Data Intelligence, 3(4), 568–577 (2021)

[27] Fritzler, A., Logacheva, V., Kretov, M.: Few-shot classification in named entity recognition task. In: Proceedings ACM/SIGAPP, pp. 993–1000 (2019)

[28] Gao, T., Fisch, A., Chen, D.: Making Pre-trained Language Models Better Few-shot Learners. In: Proceedings ACL, pp. 3816–3830 (2021)

[29] Chen, X., Zhang, N., Xie, X., et al.: Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In: Proceedings ACM Web Conference, pp. 2778–2788 (2022)

[30] Ye, H., Zhang, N., Deng, S., et al.: Ontology-enhanced Prompt-tuning for Few-shot Learning. In: Proceedings ACM Web Conference, pp. 778–787 (2022)

[31] Lester, B., Al-Rfou, R., Constant, N.: The Power of Scale for Parameter-Efficient Prompt Tuning. In: Proceedings EMNLP, pp. 3045–3059 (2021)

[32] Chen, X., Li, L., Deng, S., et al.: LightNER: a lightweight tuning paradigm for low-resource NER via pluggable prompting. In: Proceedings COLING, pp. 2374–2387 (2022)

[33] Liu, K., Fu, Y., Tan, C., et al.: Noisy-Labeled NER with Confidence Estimation. In: Proceedings NAACL, pp. 3437–3445 (2021)

[34] Yu, H., Zhang, N., Deng, S., et al.: Bridging Text and Knowledge with Multi-Prototype Embedding for Few-Shot Relational Triple Extraction. In: Proceedings COLING, pp. 6399–6410 (2020)

## AUTHOR BIOGRAPHY

Yang Yu received the BS degree in Information Engineering from Nanjing University of Aeronautics and Astronautics in 2009. He is currently working as a deputy senior engineer in the State Grid Group, focusing on power marketing business, customer relationship management and marketing strategy.

Wei He received her BS and MS degrees from Nanjing University of Aeronautics and Astronautics in 2014 and 2017, respectively. She is currently working as a engineer in the State Grid Group, focusing on power market theory and strategy and customer service.

Yu-meng Kang received her BS degree from Hefei University of Technology in 2020. She is currently working as a assistant engineer in the State Grid Group, focusing on power marketing and customer service.

You-lang Ji received the MS degree in Electrical Engineering and Automation from Nanchang Engineering College in 1997. He is currently working as a senior engineer in the State Grid Group, focusing on electric power marketing business, customer relationship management and marketing strategy.