Wuhan University
Journal of Natural Sciences

2024, Vol.29 No.2, 125-133

Article ID 1007-1202(2024)02-0125-09 DOI https://doi.org/10.1051/wujns/2024292125

Cite this article: LIU Zhiwei, HUANG Bo, XIA Chunming *et al*. Few-Shot Named Entity Recognition with the Integration of Spatial Features[J]. *Wuhan Univ J of Nat Sci*, 2024, 29(2): 125-133.

Few-Shot Named Entity Recognition with the Integration of Spatial Features

□ **LIU Zhiwei[1], HUANG Bo[1†], XIA Chunming[1], XIONG Yujie[1], ZANG Zhensen[2], ZHANG Yongqiang[3]**

1. College of Electrical and Electronic Engineering, Shanghai University of Engineering Science, Shanghai 201620, China;
2. Shanghai Zhongyu Academy of Industrial Internet, Shanghai 201620, China;
3. AIoT Manufacturing Solutions Technology Co., Ltd., Hefei 230000, Anhui, China

Abstract: The few-shot named entity recognition (NER) task aims to train a robust model in the source domain and transfer it to the target domain with very few annotated data. Currently, some approaches rely on the prototypical network for NER. However, these approaches often overlook the spatial relations in the span boundary matrix because entity words tend to depend more on adjacent words. We propose using a multidimensional convolution module to address this limitation to capture short-distance spatial dependencies. Additionally, we utilize an improved prototypical network and assign different weights to different samples that belong to the same class, thereby enhancing the performance of the few-shot NER task. Further experimental analysis demonstrates that our approach has significantly improved over baseline models across multiple datasets.

Key words: named entity recognition; prototypical network; spatial relation; multidimensional convolution

CLC number: TP391.1

0 Introduction

Named Entity Recognition (NER) stands as a cornerstone in natural language processing and represents a fundamental undertaking. The principal objective revolves around the discernment of entity spans nestled within sentences and their subsequent categorization into precise classes. These classes encompass a spectrum of designations, notably encompassing but not limited to Person, Organization, and Location. As a traditional sequence labeling task, NER provides essential technical support for downstream applications such as information extraction, knowledge graphs, and text summarization.

The NER task has undergone several significant evolutions since its inception. In the early stages, the rule-based and dictionary-based approaches gained considerable traction. This method relies too heavily on domain experts to formulate rules and templates that may must be revised when dealing with complex linguistic

Received date: 2023-12-28

Foundation item: Supported by the Scientific and Technological Innovation 2030-Major Project of New Generation Artificial Intelligence (2020AAA0109300), Science and Technology Commission of Shanghai Municipality (21DZ2203100), and 2023 Anhui Province Key Research and Development Plan Project - Special Project of Science and Technology Cooperation (2023i11020002)

Biography: LIU Zhiwei, male, Master candidate, research direction: natural language processing. E-mail: zhiweiliu0208@gmail.com

† Corresponding author. E-mail: huangbosues@sues.edu.cn

expressions and diverse inputs. As machine learning has progressed, statistically based methods have emerged. As a quintessential statistical approach, conditional random fields[1] have demonstrated the capability to address intricate sequence annotation tasks by modeling the interdependencies among markers. However, despite this progress, these methods must be improved when effectively identifying intricate patterns. As the volume of trainable data grows and computer arithmetic capabilities improve, existing approaches have yielded promising results through deep learning. The researchers constructed the model using a more complex network structure, significantly improving model performance. In addition, existing supervised and unsupervised methods rely too heavily on the amount of annotated data. However, in real-world scenarios, NER systems frequently encounter the need to rapidly adapt to new entity types not encountered during training. This adaptation is typically accomplished through fine-tuning the original model, thereby enabling the system to perform effectively in the new domain.

Researchers have proposed few-shot learning to establish innovative concepts with a limited number of instances. In this approach, the model is initially trained within a richly-resource domain and transposed to a scarce-resource domain for specific missions. The model must quickly adapt to the data distribution within the target domain, relying on a sparse set of annotated data. Currently, few-shot learning is typically trained using the *N*-way *K*-shot pattern, where *N* represents the number of classes, and *K* represents the number of samples per class. Figure 1 illustrates an example of 2-way 1-shot instances in the target domain. Two samples of the target domain with labels, each containing only one entity type, were given. The objective is to recognize entities within the query example.

Currently, few-shot NER methods can be broadly categorized into two main types. One-stage methods classify individual words in a sentence directly by analyzing the feature distribution of the constructed classes. Fritzler *et al*[2] represented class prototypes by averaging tokens with the same label and categorizing them based on the distance from the prototype. Yang *et al*[3] utilized the transfer matrix instead of retraining the conditional random fields (CRF) model of the target domain and classified it by the *k* nearest neighbor (*k*NN) algorithm. Figure 2 shows the traditional method based on *k*NN. The value of *k* significantly influences the classification decision. Das *et al*[4] optimized the distribution distance between tokens of the same category through contrastive learning and utilized Gaussian distribution embeddings to differentiate labeling categories. Unlike the one-stage approaches, the two-stage method places more emphasis on the recognition of entity spans, and most of this work is based on a prototypical network[5]. They assume that each entity type belongs to a prototype for training and uses the *k*NN method for classification. Wang *et al*[6] formulated the classification problem as a span-level matching problem and decomposed it into a series of span processes. Ma *et al*[7] utilized meta-learning to train the span detector, aiming to discover a universal parameter initialization that can swiftly adapt to new entity classes. Wang *et al*[8] introduced a global boundary matrix and adjusted span representations through prototypical learning. Li *et al*[9] take different combinations of type names and support samples as contrast and use type-aware filtering strategies to remove spans that are far from the target domain.

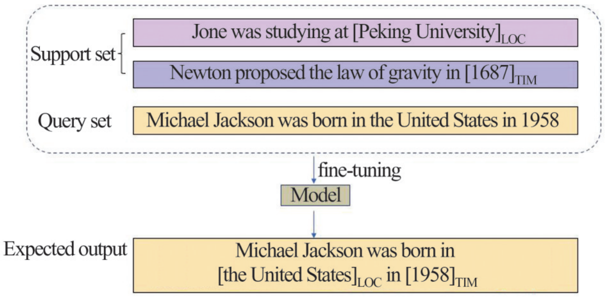
Support set
Jone was studying at [Peking University]$_{LOC}$
Newton proposed the law of gravity in [1687]$_{TIM}$
Query set
Michael Jackson was born in the United States in 1958
fine-tuning
Model
Expected output
Michael Jackson was born in [the United States]$_{LOC}$ in [1958]$_{TIM}$

Fig. 1 A 2-way 1-shot example in the target domain

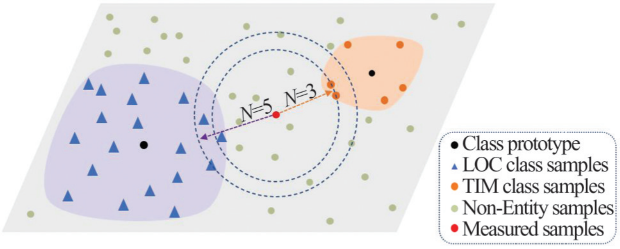

Fig. 2 The traditional classification method based on *k*NN

Despite remarkable advancements, current methods continue to grapple with challenges when confronted with few-shot NER. First, as with other sequence labeling issues, entity categories can be notably influenced by neighboring words, culminating in what is widely acknowledged as the short-range dependency issue. In practical terms, entity tokens seldom appear in isolation but manifest consecutively. There are also smoothing-based[10] methods used to address model overconfidence by spreading the probability of the span matrix over the span of neighboring entities. In pursuit of new solutions, we intend to capture the spatial intricacies of the boundary matrix via a multiscale convolutional approach and assign different convolutional kernel weights according to the actual situation. Subsequently, we aim to merge the results before and after convolution using a residual network[11]. This strategy is devised to discern and delineate a greater number of neighboring entities, thus enhancing the model's accuracy in NER. Referring to previous models, the selection of prototypes typically involves averaging samples belonging to the same class, assuming equal contribution from all samples to the prototype. However, in practical scenarios, different sample points contribute to the prototype to varying extents, thus requiring the allocation of distinct weights to each sample point.

In summary, in this work, we design a two-stage framework. In the first stage, we pass through a biaffine layer to generate the entity boundary matrix, which aids in determining the position of the entities in the sentence. To extract span matrix spatial features, we use multiscale convolution to construct the spatial relations of the fractional matrix, and the label smoothing effect is also achieved, which can better identify nested entities. In the second stage, we improve the prototypical network and assign different weights to different samples based on the KL divergence between distributions.

Our contributions can be summarized as follows:

1) We propose a novel, robust framework to tackle the problem of NER in resource-constrained scenarios.

2) We utilize multiscale convolution for feature extraction on the spatial dimension of the bounding matrix and use a weighted prototypical network for categorizing.

3) The experimental results validate the framework's effectiveness in few-shot settings. Compared with the benchmark models, the F1 score of our framework shows a good improvement in different settings.

1 Related Work

1.1 Meta-Learning

Researchers have proposed the concept of few-shot learning to drive the application of machine learning in scenarios with extremely scarce sample data[12]. Meta-learning, a popular paradigm for few-shot learning, aims to discover an optimal set of parameters that enable the model to rapidly adapt to new tasks. Finn *et al*[13] redefined the gradient descent algorithm and designed a model-agnostic meta-learner. Li *et al*[14] concurrently trained initial parameters update direction and step size based on the foundation of model-agnostic meta-learning (MAML). Jiang *et al*[15] introduced an attention-based meta-learning approach for unknown tasks and applied it to the field of NLP. Subsequently, meta-learning has been widely applied to address problems with limited data, such as machine translation[16,17] and text classification[18-20].

1.2 Few-Shot NER

Hou *et al*[21] introduced a collapsed dependency transfer mechanism into CRF to transfer abstract label dependency patterns as transition scores. Ji *et al*[22] constructed a dispersed and distributed prototype-enhanced entity-level prototypical network. Chen *et al*[23] employed limited labeled samples for class-incremental learning and generated synthetic data for pre-existing classes using a source domain model. Wang *et al*[24] transformed data representation from a high-resource to a low-resource domain through data augmentation[25]. Zhou *et al*[26] utilized the high-quality augmented data generated by the model to provide rich knowledge of entity regularities. Zhang *et al*[27] utilized prompt templates containing entity category information to construct labeling prototypes, enhancing the model's suitability for migration.

2 Method

Figure 3 illustrates the framework diagram of our approach. The model is first trained to generate a span matrix on the support set and then classified it using class prototypes. We first introduce the preliminaries. Then, we discuss how to obtain a boundary matrix with multiscale convolution and use a weight prototypical network to classification.

2.1 Preliminaries

In this stage, we formulate a few-shot named entity recognition as a span-based sequence labeling task. Given an input sequence $X=\{x_i\}_{i=1}^{L}$ of length L, we aim to identify all entity spans $M=\{\left(s_j,e_j\right)_{j=1}^{L'}\}$ and classify them into corresponding labels $Y=\{y_t\}_{t=1}^{n}$, where, x_i is the i-th token, s_j/e_j denotes the start/end position for the j-th span, L' is the number of spans in the sentence, and

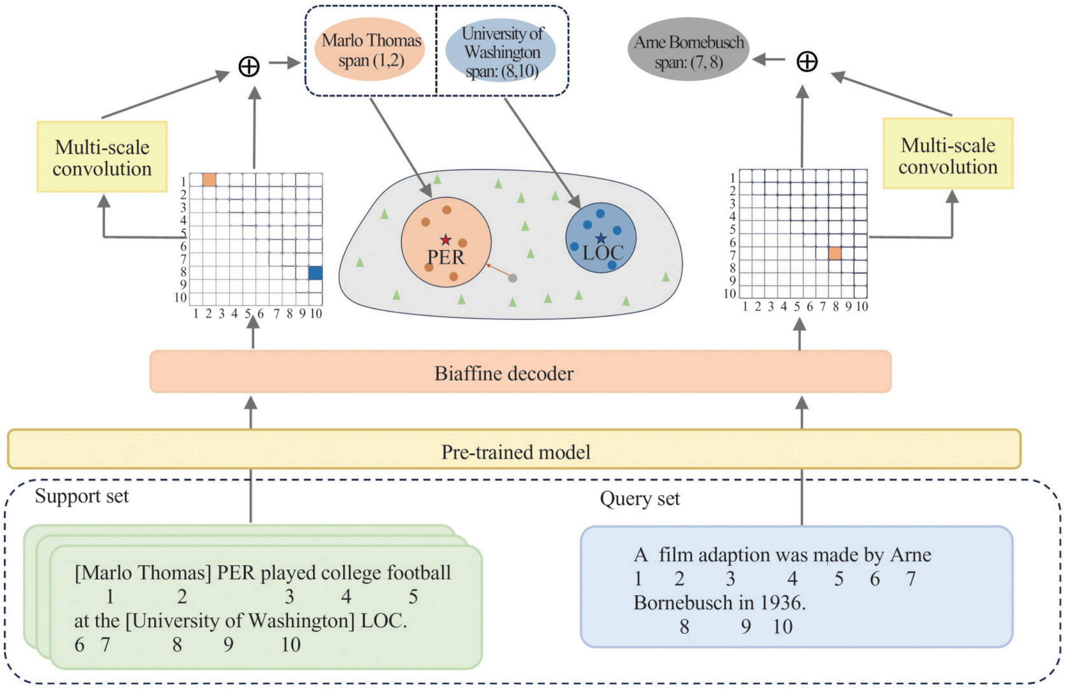

Fig.3 The framework of our proposed

y_t is the t-th entity type in the label set Y. We use standard N-way K-shot settings and divide data in the source domain as training episodes $\varepsilon_{\text{train}}=\{(S_{\text{train}},Q_{\text{train}},T_{\text{train}})\}$, where $S_{\text{train}}=(X_s,M_s,Y_s)$ denotes the support set, $Q_{\text{train}}=(X_Q,M_Q,Y_Q)$ denotes the query set, and $T_{\text{train}}=Y_{\text{train}}\cup O$ is the corresponding type set. We use a similar method to construct the target domain data for the testing process to validate the model's performance on the novel domain. Given some novel episodes $\varepsilon_{\text{novel}}=\{(S_{\text{novel}},Q_{\text{novel}},T_{\text{novel}})\}$, where $S_{\text{novel}},Q_{\text{novel}}$ represent the support and query sets in the novel domain, $T_{\text{novel}}$is the novel-type set. We expect to use a few support sets S_{novel} to fine-tune the model and make predictions on the query set Q_{novel}. In general, $T_{\text{train}}\cap T_{\text{novel}}=\varnothing$.

2.2 Entity Span Extractor

As a classic two-stage approach, we only extract all candidate entity spans from the sentences without classifying them in this stage. Given an input sequence $X=\{x_i\}_{i=1}^{L}$ from the support set S_{train}, we first utilize a pre-trained model to encode the input tokens into well-initialized embeddings $H=\{h_i\}_{i=1}^{L}$.

$$[h_1,h_2,\cdots,h_L]=\text{PLM}([x_1,x_2,\cdots,x_L]) \tag{1}$$

where $H\in\mathbb{R}^{L\times h'}$ denotes the hidden layer output of pre-trained encoder, h' denotes the hidden size. After obtaining the contextual representation, we use two separate feedforward neural networks to create different representations h_j^s/h_j^e for the start/end positions of the j-th span and then adopt a Biaffine Layer[10,28] to get the predicted score matrix:

$$P_x=h_j^s W_a h_j^e+W_b(h_j^s\oplus h_j^e)+b_m \tag{2}$$

where, W_a, W_b are the trainable parameters, b_m denotes the bias.

Considering that the labels in the support set are visible, we use a global boundary matrix to represent the ground truth of the training process.

$$\boldsymbol{\Omega}_{s_j,e_j}=\begin{cases}1, & s_j\leqslant e_j\wedge(s_j,e_j)\in M\\ 0, & s_j\leqslant e_j\wedge(s_j,e_j)\notin M\\ -\text{inf}, & s_j>e_j\end{cases} \tag{3}$$

where s_j/e_j denotes the start/end position for the j-th span, $\boldsymbol{\Omega}_{s_j,e_j}$ is the score of the span (s_j,e_j), M denotes the spans in a sentence that belong to entity types.

Since neighboring cells in the span matrix affect each other, so we use CNN with three-dimensional convolutional kernels for spatial modeling. Considering the effect of distance on span labels, we assign different weighting factors to these convolutions.

$$C_{x1}=\text{GeLU}(\text{LayerNorm}(\text{Conv2d}(P_x))) \tag{4}$$

$$C_x=C_{x1}\lambda_1+C_{x2}\lambda_2+C_{x3}\lambda_3 \tag{5}$$

where $\{\lambda_1,\lambda_2,\lambda_3\}$ represent the proportion of results for three different scales of convolution, C_x indicates the final summed result.

Considering that most of the words in the sentence belong to nonentities and the categories are imbalanced among other entities, we followed Wang *et al*[8] to use the span-based cross-entropy loss function to constrain the boundary information on each training support set. The aim is to encourage the model to be more focused on hard-to-classify samples during training by reducing the weights of easy-to-classify samples.

$$L_{\text{span}} = \log\left(1 + \sum_{1 \leqslant e_j \leqslant s_j \leqslant L} \exp\left((-1)^{\boldsymbol{\Omega}_{s_j, e_j}} (P_x + C_x)\right)\right) \tag{6}$$

3.3 Span Classification

In this phase, our objective is to categorize the spans generated earlier. The traditional prototypical network averages all samples belonging to the same category to obtain the prototype[5, 29]. Considering that different sample points have different degrees of contribution to the class prototype, we design a new prototypical calculation method. Given a test set $S=\{s_t\}_{t=1}^{T}$, where s_t represents a collection of all samples belonging to the same class, x_i represents one of the samples. We measure the difference in distribution between samples x_i and the s_t using KL divergence, where the weight of sample x can be measured by the distribution changes when the sample is not present in the test set.

$$D_{\text{KL}}(x_i) = D_{\text{KL}}[s_i \| s_t - x_i |] \tag{7}$$

We use the KL divergence as the weight of the sample x. When the KL divergence between all samples s_t and the sample distribution without x_i is smaller, it proves that the sample point contributes less to the prototype, and the corresponding weight is smaller.

$$W(x_i) = D_{\text{KL}}(x_i) \tag{8}$$

Class prototypes can be calculated by the product of weights and sample points as follows:

$$c_k = \frac{\sum_{i=1}^{|s_k|} W(x_i) f_\phi(x_i)}{\sum_{i=1}^{|s_k|} W(x_i)} \tag{9}$$

where c_k represents the prototype of class k, and $f_\phi(x_i)$ describes the sample features mapped to a high-dimensional space.

Finally, we optimize the model by the cross-entropy loss function.

$$L_{\text{dis}} = -\log \frac{1}{T} \sum_{i=1}^{T} \frac{\exp\left(-d\left(f_\phi(\hat{x}), c_k\right)\right)}{\sum_k \exp\left(-d\left(f_\phi(\hat{x}), c_k\right)\right)} \tag{10}$$

where $\hat{x}$ indicates a new sample to be tested.

3 Experiments

In this section, we present a comparison of our method with the existing few-shot NER framework. Detailed descriptions of the training settings and the final results are provided in the subsequent sections.

3.1 Settings

1) Datasets

To evaluate the generalization effect of the model in different domains, we conduct experiments on several public NER datasets, and split them into two groups. Table 1 presents the summary statistics of the datasets.

Few-NERD[30] is a novel NER dataset created using data from Wikipedia and designed for few-shot learning scenarios. Unlike previous datasets, it is annotated with a hierarchy of 8 coarse-grained and 66 fine-grained entity types. To validate the impact at different entity granularities, the researchers further divided the data into two categories, i.e., Inter and Intra.

Cross-NER contains four datasets from different fields, including the CoNLL-03 [31] dataset from the news domain, the WNUT-17 [32] dataset from the social domain, the OntoNotes [33] dataset from the general domain, and the GUM [34] dataset from the Wiki domain.

Table 1 Summary statistics of each dataset

Datasets		Domain	Type	Sentence/10^3
Few-NERD		Wikipedia	66	188.2
Cross-NER	GUM	Wiki	11	3.5
	OntoNotes	General	18	76.7
	CoNLL-03	News	4	20.7
	WNUT-17	Social	6	5.7

2) Hyperparameters

We used the BERT-base [35] as the backbone encoder to initialize the word vector. The Baffine decoder with the affine layers of hidden size 150 and dropout rate 0.2. The learning rate was searched between 2E−5 and 5E−6 on the randomly initialized weights. We chose AdamW [36] as our optimizer with a linear warm-up in the first 10% steps and a weight decay of 0.1. The batch size is set to 8, and the max sequence length is set to 128. We have chosen $\{3, 5, 7\}$ as the convolution kernel size of the boundary matrix, and the corresponding weights of the three types of convolutions are $\{0.6, 0.3, 0.1\}$. We chose PyTorch as our development environment with version 1.8, and the model was trained on an RTX 3090 GPU.

3) Baselines

We compared existing competitive few-shot NER models, such as ProtoBERT[5], Matching Network[37], StructShot and NNShot[3], ESD[6], CONTaiNER[4], L-TapNet+CDT[21], DecomMeta[7], SpanProto[8], and TadNER[9].

3.2 Main Results

Table 2 compares our model with other baseline models on the Few-NERD dataset.

1) Our model significantly outperforms TadNER in both Inter and Intra tasks. Notably, the performance in the Inter task surpasses that in the Intra task, indicating that Few-shot NER presents more significant challenges under coarse-grained conditions.

2) Across all experimental results, the performance of 1-2 shots are worse than that of 5-10 shots, mainly because fewer samples are more accessible to selection bias. The model will show a good classification effect when the selected sample points are closer to the real class prototype. However, this uncertainty of the sample point makes it difficult for the model to find that point in most cases.

3) All span-based methods outperform token-based methods in our experiments.

Table 3 displays the model's performance on Cross-NER. The results indicate that our model also performs well in cross-domain data and exhibits a 1.35% and 1.48% improvement compared to the baseline. This underscores the strong adaptability of our approach.

Table 2 F1 scores with standard deviations on Few-NERD for both inter and intra settings

Model	Intra				Inter			
	1-2-shot		5-10-shot		1-2-shot		5-10-shot	
	5-way	10-way	5-way	10-way	5-way	10-way	5-way	10-way
ProtoBERT	23.45	19.76	41.93	34.61	44.44	39.09	58.80	53.97
NNShot	31.01	21.88	35.74	27.67	54.29	46.98	50.56	50.00
StructShot	35.92	25.38	38.83	26.39	57.33	49.46	57.16	49.39
CONTaiNER	40.43	33.84	53.70	47.49	55.95	48.35	61.83	57.12
ESD	41.44	32.29	50.68	42.92	66.46	59.95	74.14	67.91
DecomMeta	52.04	43.50	63.23	56.84	68.77	63.26	71.62	68.32
SpanProto	52.19	44.03	67.76	59.97	71.30	**65.24**	77.47	73.94
TadNER	**60.78**	**55.44**	67.94	60.87	64.83	64.06	72.12	69.94
Ours	52.14	46.83	**69.21**	**61.74**	**71.84**	64.54	**78.67**	**75.69**

Table 3 F1 scores with standard deviations on Cross-NER

Model	1-shot				5-shot			
	CoNLL-03	GUM	WNUT-17	OntoNotes	CoNLL-03	GUM	WNUT-17	OntoNotes
Matching Network	19.50	4.73	17.23	15.06	19.85	5.58	6.61	8.08
ProtoBERT	32.49	3.89	10.68	6.67	50.06	9.54	17.26	13.59
L-TapNet+CDT	44.30	12.04	20.80	15.17	45.35	11.65	23.30	20.95
DecomMeta	46.09	17.54	25.14	34.13	58.18	31.36	31.02	45.55
SpanProto	46.92	16.40	27.67	35.86	58.59	34.86	30.22	46.97
Ours	**48.57**	**17.56**	**28.92**	**37.22**	**60.38**	**35.97**	**32.48**	**47.73**

Figure 4 shows the impact of the number of fine-tuning steps on the F1 score. It can be observed that the model already performed well without fine-tuning. As the fine-tuning steps increase, the model's performance continues to improve, which indicates that our model has strong domain transfer capabilities.

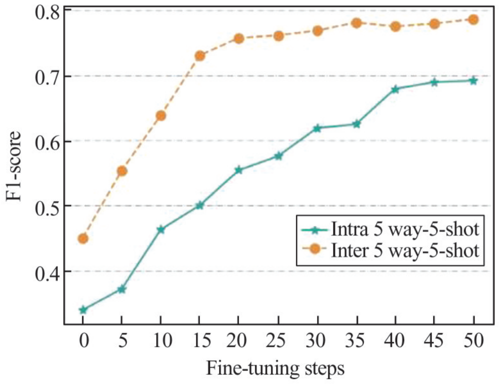

Fig. 4 The effectiveness of fine-tuning

3.3 Ablation Study

To verify the role of each module in the model, we design the following ablation experiment.

1) w/o. Multiscale Convolution, where we remove the multidimensional convolution module and directly use the span range matrix generated by the biaffine module for subsequent work.

2) w/o. Entity Span Extractor, where we do not extract entity spans but employ a traditional token-based prototypical network to train the model.

3) w/o. WeightProto Learning, where we use the *k*NN algorithm to classify the candidate span.

As depicted in Table 4, each component positively contributes to the model's performance. The removal of the multiscale Convolution module leads to a 2.57% decrease in the model's F1 score, underscoring the significance of spatial characterization within the boundary matrix. Furthermore, the span-based model surpasses the token-based approach in terms of efficacy, aligning with the comparative effectiveness observed across various domains. Finally, we opted for a weight-based prototype model. During the initialization phase, we embed instances randomly and assign different weights to multiple instances through model training. Experimental results demonstrate that our approach yields promising outcomes.

Table 4 F1 score for ablation study over different components on Cross-NER datasets with 5-way 1-shot setting

Method	CoNL L-03	GUM	WNU T-17	Onto Notes
Ours	**48.57**	**17.56**	**28.92**	**37.22**
1) w/o. Multiscale Convolution	46.84	15.29	25.34	34.51
2) w/o. Entity Span Extractor	33.64	10.44	23.26	25.37
3) w/o. WeightProto Learning	31.67	16.02	27.39	32.48

3.4 Visualization

Considering that the above experimental results cannot visualize the distribution of each entity class after model training, we use t-distributed Stochastic Neighbor Embedding (t-SNE)[38] to downsize the high-dimensional vectors. It is evident from Fig. 5 that our method makes the distribution of spans belonging to the same entity class more concentrated and the class spacing clearer, which also reflects the superiority of our framework.

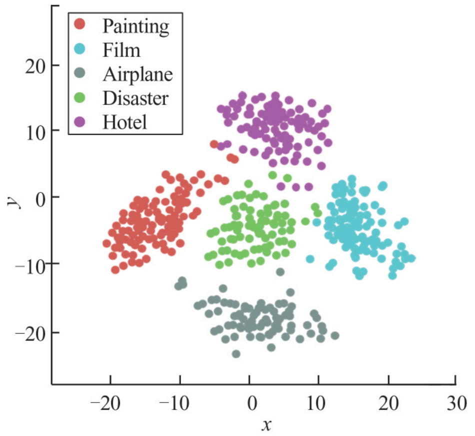

Fig. 5 t-SNE visualization of our framework on the Few-NERD dataset with 5-way 5-10-shot settings

4 Conclusion

We have introduced a comprehensive framework with the aim of addressing the challenge of identifying a limited set of named entities within a particular domain. Our empirical evaluations indicate that the two-stage methodology demonstrates superior performance compared to prevailing one-stage techniques. To thoroughly explore the spatial correlations among neighboring spans, we employ a multiscale convolution mechanism to facilitate the rationalization of spatial information within the entity span matrix. This information is subsequently integrated with the original data through a residual module, thereby enhancing the model's capacity to discern short-range dependencies. Considering that different samples have different degrees of contribution to the prototype, we propose an improved prototype calculation method to measure the importance of each sample by the KL divergence of the sample distribution. Extensive experimentation validates the efficacy of our proposed method by substantially outperforming the baseline.

References

[1] Lafferty J, McCallum A, Pereira F C N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data[C]//*Proceedings of the* 18*th Conference of the International Conference on Machine Learning*. Washington D C: AAAI Press, 2001: 282-289.

[2] Fritzler A, Logacheva V, Kretov M. Few-shot classification in named entity recognition task[C]//*Proceedings of the* 34*th ACM/SIGAPP Symposium on Applied Computing*. New York: ACM SIGGRAPH, 2019: 993-1000.

[3] Yang Y, Katiyar A. Simple and effective few-shot named entity recognition with structured nearest neighbor learning [C]//*Proceedings of the* 2020 *Conference on Empirical Methods in Natural Language Processing*. Stroudsburg: EMNLP, 2020: 6365-6375.

[4] Das S S S, Katiyar A, Passonneau R J, *et al*. CONTaiNER: Few-shot named entity recognition via contrastive learning [C]//*Proceedings of the* 60*th Annual Meeting of the Association for Computational Linguistic*. Stroudsburg: ACL, 2022: 6338-6353.

[5] Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning[C]//*Proceedings of the* 2020 *Conference in Neural Information Processing Systems*. Cambridge: NIPS, 2020: 4077-4087.

[6] Wang P Y, Xu R X, Liu T Y, *et al*. An enhanced span-based decomposition method for few-shot sequence labeling[C]// *Proceedings of the* 2022 *Conference of the North American Chapter of the Association for Computational Linguistics*. Stroudsburg: NAACL, 2022: 5012-5024.

[7] Ma T T, Jiang H Q, Wu Q H, *et al*. Decomposed meta-learning for few-shot named entity recognition[C]// *Proceedings of the* 2022 *Conference in Annual Meeting of the Association for Computational Linguistic*. Stroudsburg: ACL, 2022: 1584-1596.

[8] Wang J N, Wang C Y, Tan C Q. SpanProto: A two-stage span-based prototypical network for few-shot named entity recognition[C]//*Proceedings of the* 2022 *Conference on Empirical Methods in Natural Language Processing*. Stroudsburg: EMNLP, 2022: 3466-3476.

[9] Li Y Q, Yu Y, Qian T Y. Type-aware decomposed framework for few-shot named entity recognition [EB/OL]. [2023-10-16]. https://arxiv.org/pdf/2302.06397.pdf.

[10] Zhu E W, Li J P. Boundary smoothing for named entity recognition[C]//*Proceedings of the* 60*th Annual Meeting of the Association for Computational Linguistics*. Stroudsburg: ACL, 2022: 7096-7108.

[11] He K M, Zhang X Y, Ren S Q, *et al*. Deep residual learning for image recognition[C]//*Proceedings of the* 2016 *IEEE conference on Computer Vision and Pattern Recognition*. New York: IEEE, 2016: 770-778.

[12] Qiao S Y, Liu C X, Shen W, *et al*. Few-shot image recognition by predicting parameters from activations[C]// *Proceedings of the* 2018 *IEEE conference on Computer Vision and Pattern Recognition*. New York: IEEE, 2018: 7229-7238.

[13] Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks[C]//*Proceedings of the* 34*th International Conference on Machine Learning*. New York: ICML, 2017: 1126-1135.

[14] Li Z G, Zhou F W, Chen F, *et al*. Meta-SGD: Learning to learn quickly for few-shot learning [EB/OL]. [2017-09-28]. https://arxiv.org/pdf/1707.09835.pdf.

[15] Jiang X, Havaei M, Chartrand G, *et al*. On the importance of attention in meta-learning for few-shot text classification [EB/OL]. [2018-06-03]. https://arxiv. org/pdf/1806.00852. pdf.

[16] Gu J T, Wang Y, Chen Y, *et al*. Meta-learning for low-resource neural machine translation[C]//*Proceedings of the* 2018 *Conference on Empirical Methods in Natural Language Processing*. Stroudsburg: EMNLP, 2018: 3622-3631.

[17] Zhan R Z, Liu X B, Wong D F, *et al*. Meta-curriculum learning for domain adaptation in neural machine translation [C]//*Proceedings of the AAAI Conference on Artificial Intelligence*. Washington: AAAI, 2021: 14310-14318.

[18] Sun S L, Sun Q F, Zhou K, *et al*. Hierarchical attention prototypical networks for few-shot text classification[C]//*Proceedings of the* 2019 *Conference on Empirical Methods in Natural Language Processing and the* 9*th International Joint Conference on Natural Language Processing*. Stroudsburg: EMNLP, 2019: 476-485.

[19] Geng R Y, Li B H, Li Y B, *et al*. Dynamic memory induction networks for few-shot text classification[C]//*Proceedings of the* 58*th Annual Meeting of the Association for Computational Linguistics.* Stroudsburg: ACL, 2020: 1087-1094.

[20] Han C C, Fan Z Q, Zhang D X, *et al*. Meta-learning adversarial domain adaptation network for few-shot text classification[C]//*Proceedings of the* 2021 *Conference in Annual Meeting of the Association for Computational Linguistic*. Stroudsburg: ACL, 2021: 1664-1673.

[21] Hou Y Y, Che W X, Lai Y K, *et al*. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network[C]//*Proceedings of the* 58*th Annual Meeting of the Association for Computational Linguistics.* Stroudsburg: ACL, 2020: 1381-1393.

[22] Ji B, Li S S, Gan S D, *et al*. Few-shot named entity recogni-

tion with entity-level prototypical network enhanced by dispersedly distributed prototypes[C]//*Proceedings of the* 29*th International Conference on Computational Linguistics*. Berlin: Springer-Verlag, 2022: 1842-1854.

[23] Chen Y F, Huang Z, Hu M H, *et al*. Decoupled two-phase framework for class-incremental few-shot named entity recognition[J]. *Tsinghua Science and Technology*, 2023, **28**(5): 976-987.

[24] Wang H M, Cheng L Y, Zhang W X, *et al*. Enhancing few-shot NER with prompt ordering based data augmentation [EB/OL]. [2023-05-19]. https://arxiv. org/pdf/2305.11791. pdf.

[25] Chen S G, Aguilar G, Neves L, *et al*. Data augmentation for cross-domain named entity recognition[C]//*Proceedings of the* 2021 *Conference on Empirical Methods in Natural Language Processing*. Stroudsburg: EMNLP, 2021: 5346-5356.

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

[27] Zhang M Z, Yan H, Zhou Y Q, *et al*. PromptNER: A prompting method for few-shot named entity recognition via *k* nearest neighbor search[EB/OL]. [2023-05-19]. https://arxiv.org/ pdf/ 2305.12217.pdf.

[28] Yu J T, Bohnet B, Poesio M. Named entity recognition as dependency parsing[C]//*Proceedings of the* 58*th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg: ACL, 2020: 6470-6476.

[29] Ding N, Chen Y L, Cui G Q, *et al*. Few-shot classification with hypersphere modeling of prototypes[C]//*Proceedings of the* 2023 *Conference in Annual Meeting of the Association for Computational Linguistic*. Stroudsburg: ACL, 2023: 895-917.

[30] Ding N, Xu G W, Chen Y L, *et al*. Few-NERD: A few-shot named entity recognition dataset[C]//*Proceedings of the* 59*th Annual Meeting of the Association for Computational Linguistics and the* 11*th International Joint Conference on Natural Language Processing*. Stroudsburg: ACL, 2021: 3198-3213.

[31] Sang E F T K, De Meulder F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition[C]//*Proceedings of the* 17*th Conference on Natural Language Learning at HLT-NAACL* 2003. Stroudsburg: ACL, 2003:142-147.

[32] Derczynski L, Nichols E, Van Erp M, *et al*. Results of the WNUT2017 shared task on novel and emerging entity recognition[C]//*Proceedings of the* 3*rd Workshop on Noisy User-generated Text*. Stroudsburg: EMNLP, 2017: 140-147.

[33] Pradhan S, Moschitti A, Xue N W, *et al*. Towards robust linguistic analysis using OntoNotes[C]//*Proceedings of the* 17*th Conference on Computational Natural Language Learning*. Stroudsburg: CoNLL, 2013: 143-152.

[34] Zeldes A. The GUM corpus: Creating multilayer resources in the classroom[J]. *Language Resources and Evaluation*, 2017, **51**(3): 581-612.

[35] Kenton J D M W C, Toutanova L K. BERT: Pre-training of deep bidirectional transformers for language understanding [C]//*Proceedings of the* 2019 *Conference on Natural Language Learning at HLT-NAACL*. Stroudsburg: ACL, 2019: 4171-4186.

[36] Loshchilov I, Hutter F. Decoupled weight decay regularization[EB/OL]. [2017-11-14]. https://arxiv.org/ pdf/1711.05101. pdf.

[37] Vinyals O, Blundell C, Lillicrap T, *et al*. Matching networks for one shot learning[C]//*Proceedings of the* 30*th International Conference on Neural Information Processing Systems*. San Cambridge: NIPS, 2016: 3630-3638.

[38] Van der Maaten L J P, Hinton G E. Visualizing high-dimensional data using t-SNE[J]. *Journal of Machine Learning Research*, 2008, **9**(11): 2579-2605.

□