A Novel Three-stage Framework for Few-shot Named Entity Recognition

Shengjie Ji[1,2], **Fang Kong**[1,2*]
[1]Laboratory for Natural Language Processing, Soochow University, Suzhou, China
[2]School of Computer Science and Technology, Soochow University, Suzhou, China
20215227043@stu.suda.edu.cn, kongfang@suda.edu.cn

Abstract

Different from most existing tasks relying on abundant labeled data, Few-shot Named Entity Recognition (NER) aims to develop NER systems that are capable of learning from a small set of labeled samples and then generalizing well to new, unseen data. In this paper, with the intention of obtaining a model that can better adapt to new domains, we design a novel three-stage framework for Few-shot NER, including teacher span recognizer, student span recognizer and entity classifier. We first train a teacher span recognizer which is based on a global boundary matrix to obtain soft boundary labels. Then we leverage the soft boundary labels learned by the teacher model to assist in training the student span recognizer,which can smooth the training process of span recognizer. Finally, we adopt the traditional prototypical network as entity classifier and incorporate the idea of prompt learning to construct a more generalizable semantic space. Extensive experiments on various benchmarks demonstrate that our approach surpasses prior methods.

Keywords: few-shot learning, named entity recognition, prompt learning

1. Introduction

Named Entity Recognition (NER) is a fundamental task in natural language processing (NLP), aiming to extract pre-defined named entities from text data.With the assistance of deep neural networks, previous methods (Lample et al., 2016; Chiu and Nichols, 2016; Li et al., 2022) have demonstrated remarkable performance on fully supervised NER tasks with sufficient labeled data.However, the issue of data scarcity is a major challenge for NER in practical scenarios.To this end, much recent research(Ding et al., 2021; Huang et al., 2020, 2021) in NER focuses on developing few-shot learning methods that can leverage limited labeled data and generalize to new domain.

Existing research on few-shot NER mainly follows two directions: one is to use the idea of meta-learning, focusing on studying the model's fast adaptation ability; the other's task setting is similar to full-shot scenario, emphasizing tapping into the universal capabilities of pre-trained large language models(LLMs). In this paper, we focus on the former and conduct further exploration. Several recent studies(Fritzler et al., 2019; Hou et al., 2020; Yang and Katiyar, 2020; Das et al., 2021; Ji et al., 2022) on meta-based Few-shot NER combines the strategies of metric-learning and sequence labeling, where classification is performed by evaluating the distance between each token in query set and the tokens in support set or the prototype of each entity class. Despite effectively avoiding the overfitting issue induced by scarce samples with their metric-learning-based approaches, the presence of non-entity noise can still cause interference to their method, as all non-entity tokens share the same prototype 'O'.To tackle this problem, some methods(Ma et al., 2022; Wang et al., 2022) decompose Few-shot NER task into two stages: entity span detection and entity typing, which productively solves the noise trouble arising from 'O' label. However, there also exist additional limitations in their decomposition methods. First, for the span recognizer, the conventional hard labeling method completely separates the commonalities between episodes, and the model just blindly adapts to new domains while discarding all previous domain knowledge. Such an approach makes the overall model training process too sharp and aggressive, while also making it difficult to ensure that the model can eventually converge to a good initialization point. Moreover, limited by the number of samples,employing a conventional prototypical network (ProtoNet) as entity classifier can have disadvantages like weak prototype representation accuracy and insufficient semantic space generalization ability.

In this paper, we propose a seminal *three-stage* framework for Few-shot NER to address the limitations mentioned above, which mainly includes three submodules: *Student span recognizer*, *Teacher span recognizer* and *Entity classifier*.We design a soft-label-enhanced span recognizer in the first two stages and a prompt-based prototypical network (Prompt-ProtoNet) in *Entity classifier*. Specifically, the span-based *Teacher span recognizer* is trained in a conventional manner, and eventually a teacher model with universal capabilities is obtained to pro-

*Corresponding author

LREC-COLING 2024, pages 1293–1305
20-25 May, 2024.

vide guidance to the student model. Then, we introduce the idea of soft label learning in *Student span recognizer* to smooth the domain transition process during training.Note that different from (Hinton et al., 2015) who propose soft label learning for model compression, we utilize soft boundary learning to smooth the training process, thereby retaining universal capabilities and eventually obtain more universal initial model weights. For *Entity classifier*, we propose Prompt-ProtoNet. Specifically, we employ prompt learning to build a general semantic space that assists in the construction of specific semantic space.With the help of prompt learning, we can take full advantage of LLMs to improve the accuracy of prototypical representation and the generalization of semantic space.

We summarize our main contributions as follows:

- We propose a novel three-stage framework for Few-shot NER, including: *Student span recognizer*, *Teacher span recognizer* and *Entity classifier*.
- As far as we know, we are the pioneers in applying the idea of soft label learning to the few-shot field, which is helpful for obtaining more universal and faster domain-adaptive initial model weights.
- Our work is the first to integrate prompt learning with prototypical network, which can take advantage of LLMs for a more generalizable semantic space and more accurate entity prototype representation in ProtoNet.
- Extensive experiments demonstrate that our method achieves new state-of-the-art performance on two widely used benchmarks.

2. Related work

2.1. Few-shot learning

Few-shot learning(Wang et al., 2020) aims at training models that can generalize to new classes only with very limited amounts of labeled data and now meta-learning(Hochreiter et al., 2001) has become a popular paradigm as it aligns well with the goal of few-shot tasks.Recent studies center around metric-based methods.Matching Networks(Vinyals et al., 2016) performs classification by measuring cosine similarity between images and the samples in the support set.Prototypical networks(Snell et al., 2017) utilizes the distance between each token and the prototypes constructed from the support set to determine its class.

2.2. Few-shot NER

Different from conventional NER with adequate labeled data, Few-shot NER is confronted with data scarcity and domain transfer. Existing research on few-shot NER mainly follows two ideas.One is using meta-learning methods.This type of research often follows the N-way K-shot meta-task paradigm and ultimately results in obtaining optimal model initialization parameters by training over a large number of domain episodes.The other is similar to full-shot scenarios, often combined with transfer learning and prompt learning, aiming to achieve model optimization in certain specific domains.

Numerous studies on meta-learning Few-shot NER are based on metric-learning.Among them, Fritzler et al. (2019) and Ji et al. (2022) employ prototypical networks.Zhang et al. (2023) propose a prompting method via k-nearest neighbor search.Das et al. (2021) first leverage Gaussian Embedding with a contrastive learning method.Cao et al. (2023) adopt Gaussian distribution as transition function.Additionally, Wang et al. (2021) decompose the task into a series of span-level procedures.Fang et al. (2023) uses a memory module to utilize the information from the source domain to augment prototypes. To address the noise disturbance caused by non-entity tokens, Ma et al. (2022), Wang et al. (2022) and Li et al. (2023) split Few-shot NER into two stages, while Ma et al. (2023) propose a representation learning method for "O" and unlabeled entities. Chen et al. (2023) also injects contextual NER capabilities into PLMs through pretraining.Dong et al. (2023) devise a joint pre-training and semantic decoupling method for Few-shot NER.Moreover, some studies explore the potential of LLMs in few-shot NER.Ashok and Lipton (2023) propose a prompting-based NER method.Ma et al. (2023) propose an adaptive filter-then-rerank paradigm.

Some studies abandon the meta-learning task setting and use prompt learning to discover the universal capabilities inherent in LLMs for Few-shot NER.Lee et al. (2021) find that concatenating suitable in-contexts after the input is effective. Ma et al. (2021) transform NER into an LM task consistent with the pre-training stage.Chen et al. (2021), Lu et al. (2022) and Chen et al. (2022) redesign the NER task as a generative manner. However, these prompt-based techniques which abandon the meta-learning setting tend to lack stability, and often require the designer to continuously optimize and adjust the prompt when facing new domain data.

In this paper, we mainly follow in the footsteps of the former(meta-learning based research) for further exploration.

Types	(1). Location (2). Person
Support	(1).The nearest tube station is [covent garden]$_{\text{Location}}$. (2).[Shami]$_{\text{Person}}$ was born into a shiite family in 1945 .
Query	Culbertson came back to fort union in 1840.
Output	Location: fort union Person: Culbertson

Figure 1: An example of the 2-way 1-shot setting on NER task.

3. Task Definition

For few-shot tasks, a model need to be trained in source domain $\mathcal{D}_{train}$ and then finetuned in a target domain $\mathcal{D}_{test}$ with sample-limited support set S_{test}.We follow the N-way K-shot setting as Ding et al. (2021) which means N entity types to be recognized and K samples available for each entity type in one episode.Figure 1 shows an example of a 2-way 1-shot NER task. More specifically, dataset $\mathcal{D}$ is composed of episodes and $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{eval} \cup \mathcal{D}_{test}$. In an episode $\mathcal{E} = (\mathcal{S}, \mathcal{Q}, \mathcal{T}) \in \mathcal{D}$, $\mathcal{S}$, $\mathcal{Q}$ and $\mathcal{T}$ mean support set, query set and entity type set, respectively.During the training phase, $\mathcal{S}_{train}$, $\mathcal{Q}_{train}$ and $\mathcal{T}_{train}$ are all available for model training while only $\mathcal{S}_{test}$ and $\mathcal{T}_{test}$ can be used in testing phase.

In this paper, we handle Few-shot NER as a span extraction task.For each sample (X, P, Y) in episode $\mathcal{E}$, $X = \{x_i\}_{i=1}^{L}$ represents the input sentence X with L tokens. $P = \{(i_t, j_t)\}_{t=1}^{M}$ means that the sentence X contains M entities, and the boundary index of the t-th entity is located by i_t and j_t. Moreover, $Y = \{y_{i_t,j_t}\}_{t=1}^{M}$ is the entity type set corresponding to the entity span set P.

4. Methods

In this work, we split Few-shot NER into three submodules:*Teacher span recognizer*, *Student span recognizer* and *Entity classifier*. In general, we use the first two stages to train a soft-label-enhanced span recognizer, and then train an entity classifier in the last stage. Figure 2 illustrates the overall framework of our method.

4.1. Span recognizer

In this section, we introduce the basic structure of our span recognizers.Note that the teacher span recognizer and student span recognizer share the same model architecture, and the only difference is in the loss function.For an input sentence $X = \{x_i\}_{i=1}^{L} \in \mathcal{S}_{train}$ with L tokens, we obtain the contextualized representations $H = \{h_i\}_{i=1}^{L}$ via BERT(Devlin et al., 2019).Furthermore, we use two feedforward layers to construct key and $query$ which depend on start and end indices, respectively.

$$q_i = W_q h_i + b_q, \qquad k_i = W_k h_i + b_k \tag{1}$$

where $q_i \in \mathbb{R}^d$ stands for entity head representation and $k_i \in \mathbb{R}^d$ denotes entity tail representation. $W_q \in \mathbb{R}^{d\times d}$ and $W_k \in \mathbb{R}^{d\times d}$ are trainable weights.$b_q \in \mathbb{R}^d$ and $b_k \in \mathbb{R}^d$ are biases. Next, we utilize self-attention mechanism(Vaswani et al., 2017) to calculate the probability score for each span.The specific probability scoring method refers to SpanProto(Wang et al., 2022):

$$s(i,j) = q_i^T k_j + W_v(h_i + h_j) \tag{2}$$

where, $W_v \in \mathbb{R}^{d\times d}$ is a trainable weight.

4.2. Teacher span recognizer

We first train a teacher span recognizer to learn soft boundary labels.In particular, we expect that the teacher model pays more attention to the capture of entity boundary information, regardless of entity type.Thus, we label the span belonging to any entity as the probability of 1, while the non-entity span as the probability of 0. During the training phase of teacher model, we adopt the loss function proposed by Su et al. (2022a)[1]:

$$\begin{aligned}\mathcal{L}_{hard} = &\log\left(1 + \sum_{(i,j)\in Pos} e^{-s(i,j)}\right) + \\ &\log\left(1 + \sum_{(i,j)\in Neg} e^{s(i,j)}\right)\end{aligned} \tag{3}$$

where, Pos and Neg denote the entity set and the non-entity set, respectively.Note that to maintain the consistency between training phase and testing phase, we only use support set $\mathcal{S}_{train} \in \mathcal{E}_{train}$ for model training. As a result, we can get the soft boundary label of span (i, j) as follows:

$$\rho(i,j) = sigmoid(s(i,j)). \tag{4}$$

4.3. Student span recognizer

For span recognizer, the conventional hard labeling method completely separates the commonalities between episodes. During training, the model just blindly adapts to new domain while constantly discarding previous domain knowledge. We believe that in the meta-learning setting, such a training

[1]https://spaces.ac.cn/archives/8373

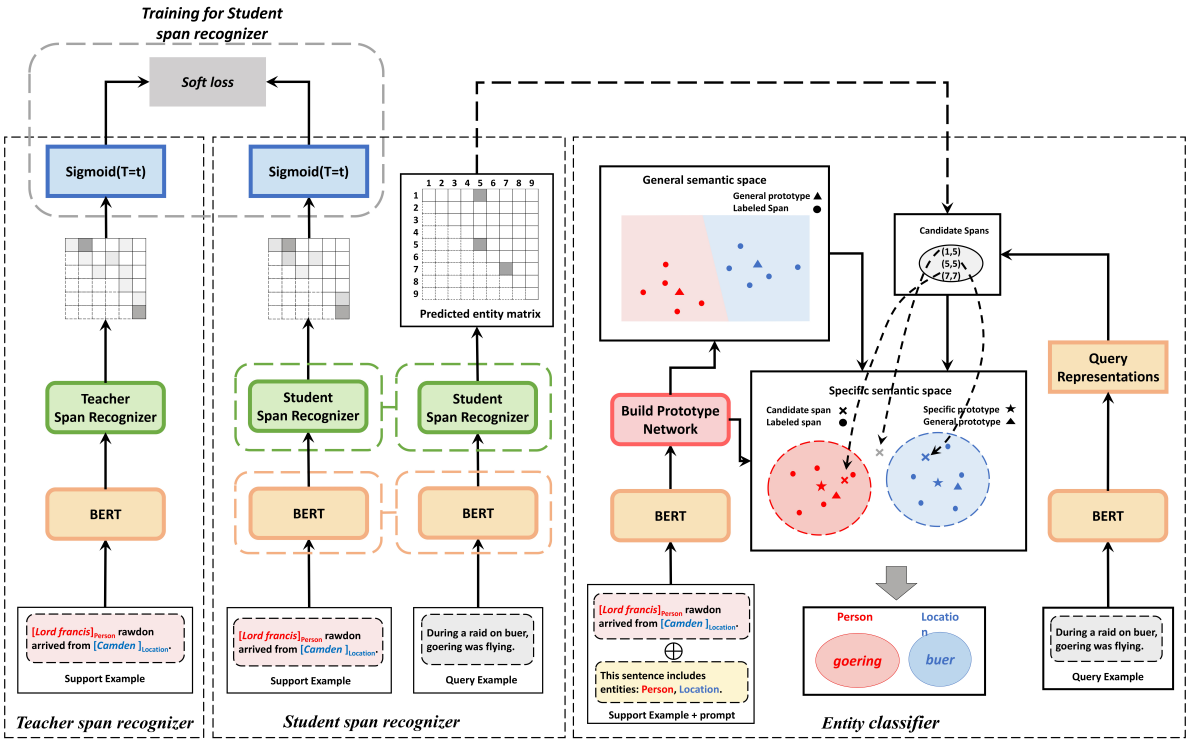

Figure 2: The framework of our three-stage model.We first train a reliable teacher span recognizer on the training set.Then, with the assistance of soft labels provided by the teacher model, we train a Student span recognizer with enhanced generalization ability.In the entity classification stage, we take advantages of prompt to build two semantic space for entity classification.

method will aggravate model forgetting, which hinders the model from learning task-relevant meta information and prevents it from converging to a general and quickly domain-adaptable initialization point. To address this issue, we consider letting a teacher model that has previewed knowledge and has certain universal capabilities provide assistance, so that the student model can softly adapt to new domains. This allows the student to not only adapt well to new domains, but also converge to a stable and universal initialization point.

During the training phase of student span recognizer, we employ the soft loss function proposed by Su et al. (2022b)[2]:

$$\mathcal{L}_{soft} = \log\left(1 + \sum_{(i,j)} e^{-s(i,j)+\log\rho(i,j)}\right) + \log\left(1 + \sum_{(i,j)} e^{s(i,j)+\log(1-\rho(i,j))}\right) \quad (5)$$

Note that in order to avoid potential prediction errors in soft label, we will also take into account the actual hard label.The hard loss function is the same as Eq.(3).As a result, the training loss of student span recognizer can be calculated as:

$$\mathcal{L}_{span} = \mathcal{L}_{hard} + \lambda\mathcal{L}_{soft} \quad (6)$$

where, λ is a hyper-parameter.Note that similar to the teacher model, student span recognizer also only uses S_{train} for training.

[2]https://spaces.ac.cn/archives/9064

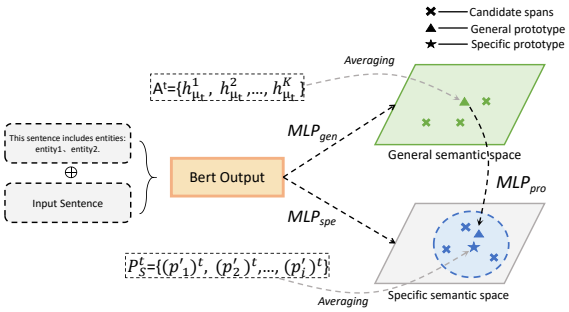

Figure 3: Mapping process of Prompt-ProtoNet.

4.4. Entity classifier

For entity classifier, we intend to assign a predefined entity type for the span extracted by student span recognizer.Inspired by prototypical networks(Snell et al., 2017), we further incorporate the idea of prompt learning and propose Prompt-

ProtoNet as our entity classifier. The Mapping process of Prompt-ProtoNet is visualized in Figure 3.

4.4.1. General semantic space

During model training, we randomly sample an episode $\mathcal{E}_{train} = (\mathcal{S}_{train}, \mathcal{Q}_{train}, \mathcal{T}_{train}) \in \mathcal{D}_{train}$. For an input sentence $X^k = \{x_i\}_{i=1}^{L} \in \mathcal{S}_{train}$, k means the k-th sample in support set $\mathcal{S}_{train}$.We first concatenate a prompt template "$This\ sentence\ includes\ entities : \mu_1,\ \mu_2,\ ...$" to the end of it.Note that μ_t is the description of entity class $t \in \mathcal{T}_{train}$ in episode $\mathcal{E}_{train}$.As a result, a new sentence $\mathcal{X}^k = \{x_1, ..., x_L, [SEP], ..., \mu_1, \mu_2, ...\}$ can be obtained. Then, we use a multi-layer perceptron MLP_{gen} to map embeddings into a general semantic space and obtain $H_{gen}^k = \{h_1, ..., h_L, ..., h_{\mu_1}, h_{\mu_2}, ...\}$. We take the first token of h_{μ_t} as a satellite node of prototype c_{gen}^t.Within a K-shot task, we can acquire A^t, a set of satellite nodes for entity class t in support set S_{train}, where $|A^t| = K$. Consequently, the general prototype c_{gen}^t of entity class t can be determined by averaging the satellite nodes in A^t:

$$c_{gen}^t = \frac{1}{K} \sum_{a^t \in A^t} a^t. \tag{7}$$

The span representation in general semantic space can be computed by summing up the boundary token representations: $p_{i,j} = h_i + h_j$. Here, $p_{i,j}$ denotes the representation of span starting with i and ending with j.

During training time, for $(X_{\mathcal{Q}}, P_{\mathcal{Q}}, Y_{\mathcal{Q}}) \in \mathcal{Q}_{train}$, we aim to minimize the distance between each entity span and its corresponding prototype. The loss in general semantic space can be calculated as follows:

$$\begin{aligned} \mathcal{L}_{\text{gen}} &= \frac{1}{|P_{\mathcal{Q}}|} \sum_{\substack{(i,j) \in P_{\mathcal{Q}} \\ y_{i,j} \in Y_{\mathcal{Q}}}} -\log \theta\left(y_{i,j} | (i,j)\right) \\ \theta\left(y_{i,j} | (i,j)\right) &= softmax(-d(c_{gen}^{y_{i,j}}, p_{i,j})) \end{aligned} \tag{8}$$

and $d(\cdot,\cdot)$ denotes Euclidean distance.$y_{i,j}$ represents the entity type of the span starting with i and ending with j.

4.4.2. Specific semantic space

Similar with section 4.4.1, given the concatenated sentence $\mathcal{X}^k$, embeddings are mapped into the specific semantic space by another MLP_{spe} and then we can obtain $H_{spe}^k = \left\{h_1^{'}, ..., h_L^{'}, ..., h_{\mu_1}^{'}, h_{\mu_2}^{'}, ...\right\}$. Likewise, we calculate span representation by: $p_{i,j}^{'} = h_i^{'} + h_j^{'}$.

Different from general prototypes, specific prototypes are constructed by entity spans.Specifically, we acquire the specific prototype c_{spe}^t for entity type $t \in \mathcal{T}_{train}$ by averaging span representations which share the same entity type t:

$$\begin{aligned} c_{spe}^t &= \frac{1}{M_t} \sum_{P_{\mathcal{S}} \in \mathcal{S}} \sum_{\substack{(i,j) \in P_{\mathcal{S}} \\ y_{i,j} \in Y_{\mathcal{S}}}} \mathbb{I}(y_{i,j} = t) p_{i,j}^{'} \\ M_t &= \sum_{P_{\mathcal{S}} \in \mathcal{S}} \sum_{\substack{(i,j) \in P_{\mathcal{S}} \\ y_{i,j} \in Y_{\mathcal{S}}}} \mathbb{I}(y_{i,j} = t). \end{aligned} \tag{9}$$

where, $\mathbb{I}(\cdot)$ is a filtering function.M_t is the count of entities belonging to class t in support set $\mathcal{S}$. Then with the help of specific prototype c_{spe}^t,we can compute specific loss like Eq.(8):

$$\begin{aligned} \mathcal{L}_{\text{spe}} &= \frac{1}{|P_{\mathcal{Q}}|} \sum_{\substack{(i,j) \in P_{\mathcal{Q}} \\ y_{i,j} \in Y_{\mathcal{Q}}}} -\log \theta^{'}\left(y_{i,j} | (i,j)\right). \\ \theta^{'}\left(y_{i,j} | (i,j)\right) &= softmax(-d(c_{spe}^{y_{i,j}}, p_{i,j}^{'})) \end{aligned} \tag{10}$$

Afterwards, we establish a connection between two semantic spaces.The connection step involves another MLP to map the embeddings of prototypes from general semantic space into specific semantic space:

$$\left(c_{spe}^t\right)^{'} = MLP_{pro}(c_{gen}^t). \tag{11}$$

For episode $\mathcal{E}_{train} = (\mathcal{S}_{train}, \mathcal{Q}_{train}, \mathcal{T}_{train})$, we attempt to reduce the distance between original specific prototype and the new specific prototype mapped from general semantic space.The loss function between prototypes that share the same entity class t can be formulated as follows:

$$\mathcal{L}_{pro} = \frac{1}{N} \sum_{t \in \mathcal{T}_{train}} -\log \theta(c_{spe}^t | (c_{spe}^t)^{'}) \tag{12}$$

The final loss is a weighted sum of the three losses:

$$\mathcal{L}_{classier} = \mathcal{L}_{gen} + \mathcal{L}_{spe} + \alpha \mathcal{L}_{pro}. \tag{13}$$

where, α is hyper-parameter.

4.5. Inference

Given an episode $\mathcal{E}_{test} = (\mathcal{S}_{test}, \mathcal{Q}_{test}, \mathcal{T}_{test})$, we first finetune student span recognizer in support set $\mathcal{S}_{test}$ and predict candidate entities in $\mathcal{Q}_{test}$.Note that for fast adaptation to new domains, student span recognizer is only finetuned according to hard labels as Eq.(3).Then we select the spans that satisfy $\rho(i,j) \geq 0.5$ as candidate entities.Next, we utilize Prompt-ProtoNet to classify the candidate entities.We first employ $\mathcal{S}_{test}$ to build prototypes, and then classify each candidate entity according to its distance to each specific prototype. Additionally, drawing inspiration from SpanProto(Wang et al., 2022), we mark candidate spans as negative if their distances to all prototypes are larger than r and r is a hyper-parameter.

[4]https://github.com/microsoft/vert-papers/tree/master/papers

Models	Intra					Inter				
	1∼2-shot		5∼10-shot		Avg.	1∼2-shot		5∼10-shot		Avg.
	5 way	10 way	5 way	10 way		5 way	10 way	5 way	10 way	
ProtoBERT†	20.76±0.84	15.05±0.44	42.54±0.94	35.40±0.13	28.44	38.83±1.49	32.45±0.79	58.79±0.44	52.92±0.37	45.75
NNShot†	25.78±0.91	18.27±0.41	36.18±0.79	27.38±0.53	26.90	47.24±1.00	38.87±0.21	55.64±0.63	49.57±2.73	47.83
StructShot†	30.21±0.90	21.03±1.13	38.00±1.29	26.42±0.60	28.92	51.88±0.69	43.34±0.10	57.32±0.63	49.57±3.08	50.53
CONTaiNER(Das et al., 2021)	40.43	33.84	53.70	47.49	43.87	55.95	48.35	61.83	57.12	55.81
SpanProto*	39.76±1.72	31.62±0.73	51.05±0.96	46.05±0.31	42.12	55.72±1.21	50.22±1.03	62.65±0.11	57.64±0.45	56.56
ESD†	36.08±1.60	30.00±0.70	52.14±1.50	42.15±2.60	40.09	59.29±1.25	52.16±0.79	69.06±0.80	64.00±0.43	61.13
DecomposedMetaNER†	49.48±0.85	42.84±0.46	62.92±0.57	57.31±0.25	53.14	64.75±0.35	58.65±0.43	71.49±0.47	68.11±0.05	65.75
Ours	**56.35±0.64**	**50.51±0.36**	**65.22±0.52**	**58.35±0.19**	**57.61**	**68.20±0.79**	**64.72±0.23**	**72.86±0.46**	**68.62±0.27**	**68.60**

Table 1: F1 scores on FewNERD.The best results are in **bold**.† denotes the result reported in Ma et al. (2022)[4].∗ represents the results we reproduce with the same dataset version.

	1-shot					5-shot				
	News	Wiki	Social	Mixed	Avg.	News	Wiki	Social	Mixed	Avg.
TransferBERT†	4.75±1.42	0.57±0.32	2.71±0.72	3.46±0.54	2.87	15.36±2.81	3.62±0.57	11.08±0.57	35.49±7.60	16.39
SimBERT†	19.22	6.91	5.18	13.99	11.33	32.01	10.63	8.20	21.14	18.00
Matching Network†	19.50±0.35	4.73±0.16	17.23±2.75	15.06±1.61	14.13	19.85±0.74	5.58±0.23	6.61±1.75	8.08±0.47	10.03
ProtoBERT†	32.49±2.01	3.89±0.24	10.68±1.40	6.67±0.46	13.43	50.06±1.57	9.54±0.44	17.26±2.65	13.59±1.61	22.61
L-TapNet+CDT(Hou et al., 2020)	44.30±3.15	12.04±0.65	20.80±1.06	15.17±1.25	23.08	45.35±2.67	11.65±2.34	23.30±2.80	20.95±2.81	25.31
DecomposedMetaNER†	46.09±0.44	17.54±0.98	25.14±0.24	34.13±0.92	30.73	58.18±0.87	31.36±0.91	31.02±1.28	45.55±0.90	41.53
Ours	**57.42±0.28**	**30.89±0.75**	**27.91±0.44**	**37.72±0.83**	**38.49**	**62.44±0.56**	**38.57±0.64**	**31.23±1.02**	**46.64±0.49**	**44.62**

Table 2: F1 scores on Cross-DataSet.The best results are in **bold**.† denotes the result reported in Ma et al. (2022).

5. Experiments

5.1. Datasets

We evaluate our approach on two popular meta-based datasets: Few-NERD(Ding et al., 2021)[5] and Cross-DataSet(Hou et al., 2020)[6]. **Few-NERD** is annotated with a schema of 8 coarse-grained and 66 fine-grained entity types.Moreover, it involves two different tasks: **Intra** and **Inter**.For **Intra**, entities in the training set, validation set, and test set belong to different coarse-grained types.For **Inter**, coarse-grained types can be shared across different sets, while there is no overlap between fine-grained entity types.Few-NERD also adopts four sampling settings for each task: 5-way 1∼2-shot, 5-way 5∼10-shot, 10-way 1∼2-shot and 10-way 5∼10-shot. **Cross-DataSet** is based on four datasets from different domains: CoNLL-03(Tjong Kim Sang, 2002)(News), GUM(Zeldes, 2017)(Wiki), WNUT-17(Derczynski et al., 2017)(Social) and OntoNotes(Pradhan et al., 2013)(Mixed).It has two sampling settings: 1-shot and 5-shot.To ensure fairness in comparison, we directly use the data provided by Hou et al. (2020).

5.2. Baselines

We compare the performance of our approach with various strong Few-shot NER models, including ProtoBERT(Ding et al., 2021; Hou et al., 2020), NNshot(Ding et al., 2021), StructShot(Ding et al., 2021), CONTaiNER(Das et al., 2021), ESD(Wang et al., 2021), L-TapNet+CDT(Hou et al., 2020), TransferBERT(Hou et al., 2020), SimBERT(Hou et al., 2020), Matching Network(Hou et al., 2020), DecomposedMetaNER(Ma et al., 2022) and SpanProto(Wang et al., 2022). Details about them are present in A.1.

5.3. Implementation details

Following previous methods, we use BERT-base-uncased(Devlin et al., 2019) as our encoder.Within our three-stage framework, the BERT encoders used by the three models are independent, and the parameters of Student span recognizer are trained from scratch.In detail, we use AdamW(Loshchilov and Hutter, 2019) as our optimizer.We set the dropout ratio(Srivastava et al., 2014) to 0.3, batch size to 4, max sequence length to 128 and train all models for 1 epoch.In addition, we perform grid search to find the best parameters setting for each benchmark.As a result, weight λ for Student span recognizer, weight α for Classifier, threshold r are 0.6, 0.2 and 5.0, respectively.More details about parameters are provided in Appendix A.2.

5.4. Main results

Table 1 and Table 2 respectively illustrate the comparison results of our approach and other baselines on Few-NERD and Cross-DataSet.Based on them, we can conclude that our approach outperforms previous methods with a large margin.Under exactly the same task setting, the comparison between our approach and current SOTA model DecomposedMetaNER(the one with the best performance and the most comprehensive experiments at present) reveals that our approach achieves an average improvement of 4.34% and 2.85% on Few-NERD

[5]https://github.com/thunlp/Few-NERD
[6]https://github.com/AtmaHou/FewShotTagging

	Intra	Inter
Ours	**57.48**	**68.60**
1)w/o Soft Boundary Learning	56.12	66.63
2)w/o Prompt	53.28	65.07
3)w/o Soft Boundary Learning w/o Prompt	51.85	61.90

Table 3: The average F1 scores of ablation study on Few-NERD.

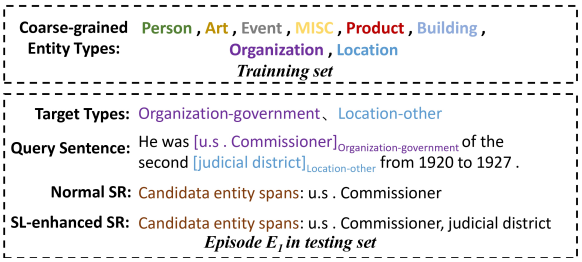

Figure 4: Case study of two different span recognizers in Few-NERD inter. **Normal SR**: directly use teacher span recognizer for entity span extracting. **SL-enhanced SR**: use soft-label-enhanced student span recognizer as an alternative.

intra and Few-NERD inter, respectively.Likewise, on the 1-shot setting and 5-shot setting of Cross-DataSet, our approach exhibits an average improvement of 7.76% and 3.04%, respectively.In addition, Table 1 also indicates that our method shows a more significant improvement on the more challenging Few-NERD intra than on Few-NERD inter, which indicates that our method has better domain adaptation capability and richer universal knowledge.Furthermore, we observe that our approach shows a higher improvement on the 1-shot setting compared to the 5-shot setting.We attribute this to the fact that the prompt-based general semantic space provides effective guidance in the case of sparse data, but its advantage becomes less apparent in the the 5-shot setting which contains relatively abundant data.

6. Analysis

6.1. Ablation study

In order to verify the effectiveness of each component in our approach, we conduct ablation experiments with the following variants. 1)w/o Soft Boundary Learning: we directly replace Student span recognizer with Teacher span recognizer and then finetune it on the support set to extract entity spans. 2)w/o Prompt: we remove the general semantic space constructed by prompts and use a conventional ProtoNet as the entity classifier. 3)w/o Soft Boundary Learning w/o Prompt: we use fine-tuned Teacher span recognizer as entity span extractor and conventional ProtoNet as the classifier.

Table 3 provides evidence for the necessity of each component in our proposed approach.In summary, we can conclude that: 1)the incorporation of soft label learning can indeed improve the fast domain adaptation capability of our model; 2)the joint use of prompt and ProtoNet can expand the universal knowledge of semantic space and improve the accuracy of prototypes; 3)the integration of soft-label-enhanced span recognizer and Prompt-ProtoNet can lead to a significant improvement in the overall performance of the model.

Model	Intra		Inter	
	1-shot	5-shot	1-shot	5-shot
MAML span recognizer	66.78	72.68	72.46	74.33
Teacher span recognizer	70.65	76.88	70.32	74.55
Student span recognizer	**73.08**	**78.39**	**73.28**	**76.66**

Table 4: The average F1 scores of span recognizers on Few-NERD.Details are presented in Appendix A.3.

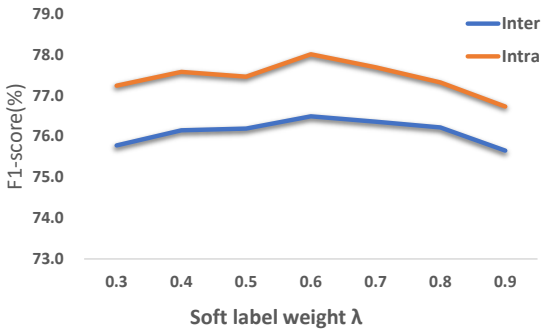

Figure 5: Impact of λ on the F1 score of student span recognizer on 5-way 5∼10-shot Few-NERD.

6.2. Impact of soft boundary learning

Through a case study, we investigate the benefits of soft boundary learning for span recognizer.Figure 4 shows that the student model trained with soft labels can identify the entity span 'judicial district', while the teacher model overlooks it.Additionally, Table 4 demonstrates that Student span recognizer enhanced by soft labels outperforms Teacher span recognizer comprehensively, and can also easily surpass MAML-enhanced span recognizer, which is a classic meta-learning algorithm commonly used in various few-shot tasks.

We attribute this to the guidance from the teacher model which helps smooth the training process of the student model and finally converges to a universal, fast domain-adaptive initialization point.

Then, we analyze how the weight of soft loss λ impacts the model during training.As presented in Figure 5, Student Span Recognizer exhibits the highest performance when the weight of soft label loss is controlled at 0.6.

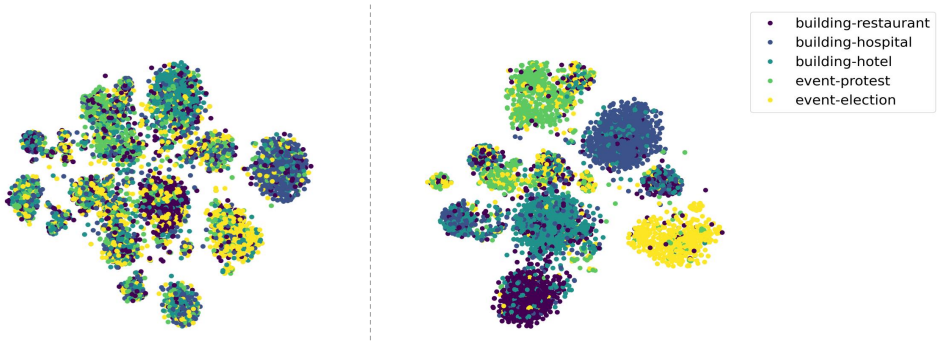

Figure 6: The t-SNE visualization of entity representations on Few-NERD Intra, 5-way 5∼10-shot validation set.We randomly choose 5 classes which include a total of 9722 samples.The left section shows the conventional ProtoNet's 2D visualization, while the right depicts Prompt-ProtoNet's 2D visualization.

Classifier	Intra		Inter	
	1-shot	5-shot	1-shot	5-shot
ProtoNet	61.57	75.70	83.54	91.35
Prompt-ProtoNet	**72.45**	**78.83**	**90.51**	**93.27**

Table 5: The average F1 scores on Few-NERD.Results are all based on ground truth entities.For detail results, please refer to Appendix A.4.

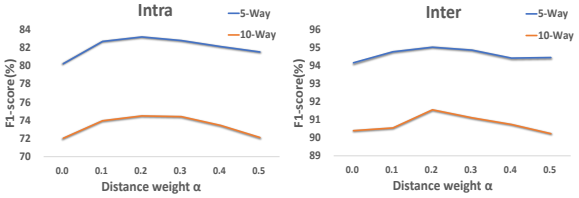

Figure 7: The influence of hyper-parameter α on the F1 score of Prompt-ProtoNet and the results are based ib the 5∼10-shot setting of Few-NERD.

Model	Intra		Inter	
	1-shot	5-shot	1-shot	5-shot
ESD	33.04	47.15	55.73	66.53
CONTaiNER	37.16	50.60	52.15	59.48
Teacher w/o finetune	36.83	46.96	59.22	64.57
Ours w/o finetune	**45.75**	**50.63**	**65.64**	**68.29**

Table 6: The average F1 without being finetuned according to support set during testing.

6.3. Effectiveness of prompt learning

To investigate the effectiveness of Prompt-ProtoNet, we compare the classification performance of our Prompt-ProtoNet with conventional ProtoNet on ground truth entity spans.Results in Table 5 confirms the effectiveness of Prompt-ProtoNet, particularly in the extremely low-sample 1∼2-shot settings, where prompts can endow ProtoNet with general knowledge and further improve its generalization ability. To better illustrate the generalization improvement of Prompt-ProtoNet over traditional ProtoNet, we randomly select five entity types in Few-NERD and then utilize t-SNE(van der Maaten and Hinton, 2008) to map entity representations into a 2-dimensional space. Figure 6 displays the visualization results, where Prompt-ProtoNet is capable of clustering representations of the same entity class even in various episodes, while also dispersing representations of different entity classes.Therefore, it is evident that the semantic space constructed by Prompt-ProtoNet has stronger generalization ability and entity-level knowledge extraction capability.

We further explore the effect of information interaction between two semantic spaces.We conduct experiments about α in Eq.(13), which controls the distance weight between the mapped general prototypes and specific prototypes.From Figure 7, we can find that although two semantic spaces are built from different perspectives, connecting them via spatial mapping is indeed effective.

6.4. Real-world applicability

Models may be limited and unable to perform finetuning based on the support set in practical scenarios.In this situation, we find that our method can achieve competitive results even without being finetuned according to support set.Table 6 displays the overall performance of our method on Few-NERD.We believe this is also attributed to the generalization boost provided by soft-label learning and capacity to learn general knowledge provided by Prompt-ProtoNet.

7. Conclusion

In this paper, we propose a novel *three-stage* framework for Few-shot NER, which decomposes the task into three submodules: *Student span recognizer*, *Teacher span recognizer* and *Entity classifier*.To smooth the training process in meta-task setting, we incorporate the idea of soft label learning into span-based span recognizer, which can

improve the model's domain adaptation capability.In addition, we propose Prompt-ProtoNet, which builds a general semantic space with the help of prompt to enhance the generalization potential of ProtoNet.Extensive experiments on two widely used benchmarks validate the effectiveness of our approach.In our future research, we are considering integrating in-context learning and pre-training to further uncover the capabilities of our model.

8. Acknowledgements

This work was supported by the Project 62276178 under the National Natural Science Foundation of China, the Key Project 23KJA520012 under the Natural Science Foundation of Jiangsu Higher Education Institutions and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

9. Bibliographical References

Dhananjay Ashok and Zachary C. Lipton. 2023. PromptNER: Prompting For Named Entity Recognition. *arXiv e-prints*, page arXiv:2305.15444.

Jiarun Cao, Niels Peek, Andrew Renehan, and Sophia Ananiadou. 2023. Gaussian distributed prototypical network for few-shot genomic variant detection. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 26–36, Toronto, Canada. Association for Computational Linguistics.

Jiawei Chen, Qing Liu, Hongyu Lin, Xianpei Han, and Le Sun. 2022. Few-shot named entity recognition with self-describing networks. *arXiv preprint arXiv:2203.12252*.

Jiawei Chen, Yaojie Lu, Hongyu Lin, Jie Lou, Wei Jia, Dai Dai, Hua Wu, Boxi Cao, Xianpei Han, and Le Sun. 2023. Learning in-context learning for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13661–13675, Toronto, Canada. Association for Computational Linguistics.

Xiang Chen, Ningyu Zhang, Lei Li, Xin Xie, Shumin Deng, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Lightner: A lightweight generative framework with prompt-guided attention for low-resource ner. *arXiv preprint arXiv:2109.00720*.

Jason P.C. Chiu and Eric Nichols. 2016. Named Entity Recognition with Bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca J Passonneau, and Rui Zhang. 2021. Container: Few-shot named entity recognition via contrastive learning. *arXiv preprint arXiv:2109.07589*.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Hai-Tao Zheng, and Zhiyuan Liu. 2021. Few-nerd: A few-shot named entity recognition dataset. *arXiv preprint arXiv:2105.07464*.

Guanting Dong, Zechen Wang, Jinxu Zhao, Gang Zhao, Daichi Guo, Dayuan Fu, Tingfeng Hui, Chen Zeng, Keqing He, Xuefeng Li, Liwen Wang, Xinyue Cui, and Weiran Xu. 2023. A multi-task semantic decomposition framework with task-specific pre-training for few-shot ner. *ArXiv*, abs/2308.14533.

Jinyuan Fang, Xiaobin Wang, Zaiqiao Meng, Pengjun Xie, Fei Huang, and Yong Jiang. 2023. MANNER: A variational memory-augmented model for cross domain few-shot named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4261–4276, Toronto, Canada. Association for Computational Linguistics.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in named entity recognition task. In *Proceedings of the*

34th ACM/SIGAPP Symposium on Applied Computing, pages 993–1000.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Sepp Hochreiter, A Steven Younger, and Peter R Conwell. 2001. Learning to learn using gradient descent. In *Artificial Neural Networks—ICANN 2001: International Conference Vienna, Austria, August 21–25, 2001 Proceedings 11*, pages 87–94. Springer.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1381–1393, Online. Association for Computational Linguistics.

Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2020. Few-shot named entity recognition: A comprehensive study. *arXiv preprint arXiv:2012.14978*.

Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2021. Few-shot named entity recognition: an empirical baseline study. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 10408–10423.

Bin Ji, Shasha Li, Shaoduo Gan, Jie Yu, Jun Ma, and Huijun Liu. 2022. Few-shot named entity recognition with entity-level prototypical network enhanced by dispersedly distributed prototypes. *arXiv preprint arXiv:2208.08023*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Dong-Ho Lee, Mahak Agarwal, Akshen Kadakia, Jay Pujara, and Xiang Ren. 2021. Good examples make a faster learner: Simple demonstration-based learning for low-resource ner. *arXiv preprint arXiv:2110.08454*.

Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. 2022. Unified named entity recognition as word-word relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10965–10973.

Yongqi Li, Yu Yu, and Tieyun Qian. 2023. Type-aware decomposed framework for few-shot named entity recognition. *arXiv preprint arXiv:2302.06397*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. *arXiv preprint arXiv:2203.12277*.

Ruotian Ma, Xuanting Chen, Zhang Lin, Xin Zhou, Junzhe Wang, Tao Gui, Qi Zhang, Xiang Gao, and Yun Wen Chen. 2023. Learning "O" helps for learning more: Handling the unlabeled entity problem for class-incremental NER. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5959–5979, Toronto, Canada. Association for Computational Linguistics.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Qi Zhang, and Xuanjing Huang. 2021. Template-free prompt tuning for few-shot ner. *arXiv preprint arXiv:2109.13532*.

Tingting Ma, Huiqiang Jiang, Qianhui Wu, Tiejun Zhao, and Chin-Yew Lin. 2022. Decomposed meta-learning for few-shot named entity recognition. *arXiv preprint arXiv:2204.05751*.

Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. 2023. Large Language Model Is Not a Good Few-shot Information Extractor, but a Good Reranker for Hard Samples! *arXiv e-prints*, page arXiv:2303.08559.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014.

Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Jianlin Su, Ahmed Murtadha, Shengfeng Pan, Jing Hou, Jun Sun, Wanwei Huang, Bo Wen, and Yunfeng Liu. 2022a. Global pointer: Novel efficient span-based approach for named entity recognition. *arXiv preprint arXiv:2208.03054*.

Jianlin Su, Mingren Zhu, Ahmed Murtadha, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2022b. Zlpr: A novel loss for multi-label classification. *arXiv preprint arXiv:2208.02955*.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

Jianing Wang, Chengyu Wang, Chuanqi Tan, Minghui Qiu, Songfang Huang, Jun Huang, and Ming Gao. 2022. Spanproto: A two-stage span-based prototypical network for few-shot named entity recognition. *arXiv preprint arXiv:2210.09049*.

Peiyi Wang, Runxin Xu, Tianyu Liu, Qingyu Zhou, Yunbo Cao, Baobao Chang, and Zhifang Sui. 2021. An enhanced span-based decomposition method for few-shot sequence labeling. *arXiv preprint arXiv:2109.13023*.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online. Association for Computational Linguistics.

Amir Zeldes. 2017. The gum corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Mozhi Zhang, Hang Yan, Yaqian Zhou, and Xipeng Qiu. 2023. PromptNER: A Prompting Method for Few-shot Named Entity Recognition via k Nearest Neighbor Search. *arXiv e-prints*, page arXiv:2305.12217.

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
MAML SR	66.15	67.40	73.01	72.34	72.17	72.74	74.87	73.78
Teacher SR	70.29	71.00	75.94	77.81	70.47	70.16	74.21	74.88
Student SR	**72.27**	**73.89**	**78.01**	**78.76**	**73.14**	**73.41**	**76.49**	**76.83**

Table 7: Detailed F1 scores of two entity span extractors on Few-NERD.SR: Span recognizer.

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
ProtoNet	66.57	56.56	80.16	71.24	87.77	79.31	93.21	89.49
Prompt-ProtoNet	**78.07**	**66.83**	**83.17**	**74.48**	**93.04**	**87.98**	**95.00**	**91.54**

Table 8: Detailed F1 scores obtained by two entity classifiers on Few-NERD.Note that the results are all based on ground truth entity spans.

A. Appendix

A.1. Details about baselines

Following Ma et al. (2022), we compare our *three-stage* method with following competitive models.

SimBERT(Hou et al., 2020) directly uses an unfinetuned BERT(Devlin et al., 2019) as the encoder and then makes classification according to the most similar token in support set.

ProtoBERT(Fritzler et al., 2019) employs support set to construct prototypes in the form of averaging embeddings, and then classifies tokens according to the most nearest prototypes.

TransferBERT(Hou et al., 2020) first pretrains a domain transfer model based on BERT on source domains, and then finetune it on support set for classification.

NNshot(Yang and Katiyar, 2020) determines the label of token in query set based on the distance between tokens in support set.

StructShot(Yang and Katiyar, 2020) employs an additional Viterbi decoder with the help of abstract transition probability matrixes during its inference phase.

Matching Network(Hou et al., 2020) utilizes matching network(Vinyals et al., 2016) with BERT embedding for classification.

TapNet+CDT(Hou et al., 2020) proposes a CRF-based few-shot sequence labeling framework, and introduces Collapsed Dependency Transfer to transfer label dependencies across domains.

CONTaiNER(Das et al., 2021) incorporates the idea of contrastive learning and optimizes the distribution distance between tokens through Gaussian distribution representation.

ESD(Wang et al., 2021) focuses on the information interaction between spans and strengthens the prototypical network through a series of span-related procedures.

Hyper-parameter	Value
Teacher learning rate	{2e-5, 1e-4, 2e-4}
Student learning rate	{2e-5, 1e-4, 2e-4}
Finetune learning rate	{2e-5, 8e-5, 1e-4}
Dropout rate	{0.1, 0.3, 0.5}
Weight α	{0.4, 0.5, 0.6}
Weight λ	{0.2, 0.3, 0.4}
Margin r	{4, 5, 6}

Table 9: The searching scope for each hyper-parameter in our experiments.

DecomposedMetaNER(Ma et al., 2022) decomposes NER into two submodules (span detection, entity typing) to alleviate information noise caused by non-entity tokens and further employs MAML(Finn et al., 2017) to enhance performance.

SpanProto(Wang et al., 2022) also splits NER into two stages: span extraction and entity classification.Additionally, it highlights the importance of boundary limitation in prototypical network.

A.2. Implementation Details

We validate our model on the validation set at intervals of 100 steps and the checkpoint with the highest F1 score performance within one epoch will be selected.For hyper-parameters tuning, a grid search is performed, with the search space outlined in Table 9.

A.3. Details of Span recognizer

The detailed results of MAML span recognizer, Teacher span recognizer and Student span recognizer are listed in Table 7.

A.4. Details of Prompt-ProtoNet

A detailed comparison of the experimental information between conventional ProtoNet and our Prompt-ProtoNet is presented in Table 8.