Few-Shot Named Entity Recognition with Hybrid Multi-Prototype Learning

Zenghua Liao
National University of Defense Technology
Junbo Fei
National University of Defense Technology
Weixin Zeng
National University of Defense Technology
Xiang Zhao (xiangzhao@nudt.edu.cn)
National University of Defense Technology

Research Article

Keywords: Few shot, Named entity recognition, Hybrid multi-prototype, Rigorous sampling strategy

Posted Date: October 26th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2188949/v1

Few-Shot Named Entity Recognition with Hybrid Multi-Prototype Learning

Zenghua Liao · Junbo Fei · Weixin Zeng · Xiang Zhao

the date of receipt and acceptance should be inserted later

Abstract Information extraction provides the basic technical support for knowledge graph construction and Web applications. Named entity recognition(NER) is one of the fundamental tasks of information extraction. Recognizing unseen entities from numerous contents with the support of only a few labeled samples, also termed as *few-shot learning*, is a crucial issue to be studied. Few-shot NER aims at identifying emerging named entities from the context with the support of a few labeled samples. Existing methods mainly use the *same* strategy to construct a *single* prototype for each entity or non-entity class, which has limited expressiveness power and even biased representation. In this work, we propose a novel *hybrid multi-prototype* class representation approach. Specifically, for entity classes, we first insert labels after entities in support sentences to enrich the learned token and label embeddings with more contextual information. Then, for each entity span, the contextual token embeddings are averaged to form its *entity-level* prototype, while the contextual label embedding is considered as its *label-level* prototype. The set of prototypes for all entities in a class constitute the *multi-prototype* of this entity class. For non-entity class, we directly use the set of token embeddings to represent it, where *multi-prototype* refers to the multiple token embeddings. By treating the entity and non-entity classes differently, our *hybrid* strategy can extract more precise class representations from the support examples. Furthermore, we establish a harder and more reasonable experimental setting of few-shot NER by offering a rigorous sampling strategy. Extensive empirical results show that our proposal improves F1 scores by 3%∼10% absolute points over prior models on popular benchmark `Few-NERD` under both loose and our proposed rigorous sampling constraints, achieving state-of-the-art performance.

Zenghua Liao · Junbo Fei · Weixin Zeng · Xiang Zhao
National University of Defense Technology, 109 Deya Road, Changsha, Hunan
E-mail: liaozenghua18@nudt.edu.cn, it3rat0r@163.com, zengweixin13@nudt.edu.cn, xiangzhao@nudt.edu.cn

Keywords Few shot, Named entity recognition, Hybrid multi-prototype, Rigorous sampling strategy

1 Introduction

With the gradual maturity and vigorous development of Web technology, the era of Web 3.0 based on knowledge interconnection is coming. Knowledge Graph (KG) related technologies play an important role in promoting the development of future Web. And information extraction provides the key technical support for knowledge graph construction. Named entity recognition (NER) is one of the fundamental tasks of information extraction, which locates spans from unstructured text sequence and categorizes them with pre-defined **entity classes** (e.g., `Person` and `Film`) or **non-entity** class (i.e., `Outside`, also shortened as `O`) [28,17]. Under the supervised learning setting, a long list of approaches, especially those framed on deep neural networks, can cope with NER adequately [14,2,19,20]. However, the prerequisite for these supervised models is the heavy manual annotation of labeled data, which is time-consuming and labor-intensive. Hence, how to increase the ability to recognize unseen entities from numerous contents with the support of only a few labeled samples, also termed as *few-shot learning*, is a crucial issue to be studied.

In response, an increasing number of works contributing to **few-shot NER** have emerged in recent years. These studies consider NER as a sequence labeling problem that restricts each token (in the sentence) belonging to at most one class, and tackle it by metric-based meta-learning [13]. Among them, a representative work [9], denoted as ProtoNER, constructs a **prototype** by support examples to represent each class. Then, given queries, it predicts their labels (i.e., classes) with the nearest neighbor search according to their distances to the prototypes of classes. This is further illustrated in Example 1.

Example 1 *In Figure 1 is an example of few-shot NER. There is a support sentence containing three spans belonging to the* `Film` *class (i.e,* `Titanic`*,* `Inception` *and* `The Revenant`*), one span labeled as the* `Person` *class (i.e,* `Leonardo DiCaprio`*), and three spans belonging to the* `Outside` *class (i.e,* `,`*,* `and`*,* `starred`*).*

As shown in the left of Figure 1, ProtoNER constructs a prototype for each class by averaging the embeddings of all the tokens belonging to this class. For instance, the prototype for the `Film` *class is the average of the embeddings of* `Titanic`*,* `Inception`*,* `The`*, and* `Revenant`*. Then, given a token in query sentence (e.g.,* `Rob`*), ProtoNER calculates the distance between the token embedding and the prototypes of all classes (i.e.,* `Film`*,* `Person` *and* `Outside`*), and assigns the closest class to this token (e.g.,* `Person`*).*

Nevertheless, there are two notable issues with ProtoNER: **(1)** A single prototype is constructed for each class, denoted as the averaged embedding of all tokens in this class, which has limited expressiveness power and even biased representation. This is shown in Example 1, where using the averaged token

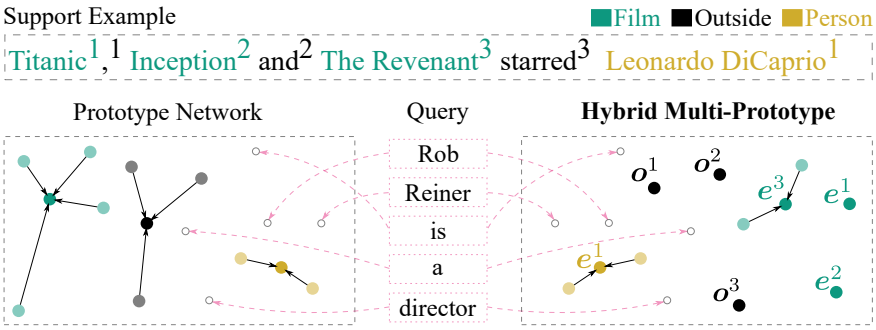


Fig. 1: An example of few-shot NER. For entity class (resp., non-entity class), ^{i} in support example denotes i-th entity (resp., token).

representation as the prototype of class `Film` would bias the class representation towards *drama* films. **(2)** The non-entity class `O` involves tokens with irrelevant (or even inconsistent) meanings, and hence the average of the token embeddings could result in a noisy prototype representation of the `O` class.

Several efforts have been proposed to mitigate these problems. For **(1)**, Hou et al. [12] use the class label (e.g., the string *film*) to augment the original prototype. However, label and token representations are learned separately and combined via weighted average, where label embeddings are obtained without context, which fails to learn precise label representations and hence cannot characterize the class sufficiently. For **(2)**, Yang et al. [33] abandon the concept of prototype and use the set of token embeddings to represent each class. Nevertheless, it only benefits the non-entity class `O` and falls short for entity classes (e.g., the mere token `The` in `The Revenant` can hardly be used to represent the class `Film`).

In this work, we aim to address the aforementioned issues by offering a *hybrid multi-prototype* construction approach, HMP. For **entity classes**, instead of using a *single* prototype to represent all entity spans in the class, we construct entity-level and label-level prototypes for each span, thus resulting in a *multi-prototype* representation of the class. Specifically, we make better use of the label information by inserting labels into support sentences, by which the token and label embeddings can be learned with more contextual information. Then for each entity, the contextual token embeddings are averaged to form its *entity-level* prototype, while the contextual label embedding is considered as its *label-level* prototype. The set of prototypes for all entity spans in a class constitutes the *multi-prototype* of this entity class. For **non-entity class**, we directly use the set of token embeddings to represent it, where the *multi-prototype* refers to the multiple token embeddings. By treating the entity and non-entity classes differently, our ***hybrid*** *multi-prototype* strategy can extract more precise class representations from the support examples, hence facilitating the inference in the query set.

Example 2 *Continue with Example 1. As shown in the bottom right of Figure 1, for entity class* `Film`*, its multi-prototype is the set* $\{\boldsymbol{e}^1, \boldsymbol{l}^1, \boldsymbol{e}^2, \boldsymbol{l}^2, \boldsymbol{e}^3, \boldsymbol{l}^3\}$*, where* $\boldsymbol{e}^1$ *(resp.,* $\boldsymbol{e}^2$*,* $\boldsymbol{e}^3$*) and* $\boldsymbol{l}^1$ *(resp.,* $\boldsymbol{l}^2$*,* $\boldsymbol{l}^3$*) are the entity-level and the label-level prototypes of the entity span* `Titanic` *(resp.,* `Inception, The Revenant`*), respectively. For* `O`*, its multi-prototype is the set of token embeddings* $\{\boldsymbol{o}^1, \boldsymbol{o}^2, \boldsymbol{o}^3\}$.

Then, for each token in the query sentence, HMP *calculates the distances between its embedding and all prototypes, and considers the class that the closest prototype belongs to as the label for this token.*

In addition, we notice that there is an issue with the N-way K-shot episodic sampling in previous few-shot NER experimental settings [12,33,6,29]. That is, to keep the context integrity for NER, they conduct *sentence-level* sampling, which is difficult to accurately satisfy the constraint of N-way K-shot. Therefore, they loose the restriction on K and only require that the final number of examples (K^*) for each entity class is above K, which, however, would lower the difficulty of this task since much more samples ($K^* > K$) would be provided for each class using the sentence-level sampling. This would in turn cause the inflation of evaluation results. To make the evaluation condition more similar to the original setting, in this work, we propose a rigorous sampling algorithm to keep K^* close to K. We compare HMP against state-of-the-art methods on popular benchmarks under both loose and our proposed rigorous sampling constraints, and the empirical results validate that HMP achieves the best performance, improving F1 scores by 3%∼10% absolute points.

Objectives. The research objectives of this work are:

- Exploring a class representation approach with less bias and more expressiveness. For this objective, we extend the single-prototype representation, which is commonly used in previous few-shot NER models, to the multi-prototype representation.
- Offering a flexible strategy to construct class representation. Taking into account the difference between entity classes and non-entity classes, we devise a hybrid construction method to obtain their multi-prototype representation.
- Making better use of class labels to offer contextual information for class representations. For this objective, we insert labels into support sentences to exploit the implicit information hidden in the labels.
- Examining whether the performance inflation exists in existing literature and establishing a fair experimental setting. To achieve this objective, we design a more rigorous sampling strategy and compare the performance of state-of-the-art models under both loose and our proposed strict sampling strategies.

Contributions. The main contributions of this work are:

- We propose a novel *multi-prototype* class representation strategy to alleviate the potential representation bias and improve the expressiveness of *single-prototype* methods.

- We devise a *hybrid* strategy to construct multi-prototypes for entity and non-entity classes according to their corresponding characteristics.
- We leverage the class labels to learn *contextual* token and label embeddings, which in turn can produce more accurate multi-prototype representation of classes.
- We put forward a rigorous experimental setting for few-shot NER, which is more reasonable and realistic than existing ones, so as to reduce the performance inflation of previous few-shot NER models and provide a fairer evaluation.

Organization. The next section discusses related works. Section 3 introduces the task formulations and our proposed rigorous evaluation setting for few-shot NER. Section 4 presents the HMP model, and Section 5 describes experiments and results. Section 6 concludes this article.

2 Related Work

Few-shot learning. Early studies on few-shot learning are relatively active in image processing [23], primarily focusing on classification problems, among which metric-based methods have been extensively explored [21,1,34]. These methods hold a hypothesis that the representation of each class can be obtained through a small amount of labeled data, and the representation of unlabeled item should have the highest similarity with that of the class to which it should belong. In the field of NLP, few-shot learning has also been investigated in tasks such as few-shot text classification [27,18], few-shot relationship extraction [11,10,31], few-shot entity typing [7,22], and few-shot NER [6].

Solutions to Few-shot NER. The majority of few-shot NER approaches [15, 9,12,33,16] consider few-shot NER as a sequence labeling problem that restricts each token (in the sentence) belonging to at most one class, and tackle it using meta-learning [13]. Fritzler et al. [9] directly transfer Prototypical Network [1] to few-shot NER, calculating a prototype for each entity class by averaging all token embeddings in the class and directly learning a b_0 to represent non-entity class. Hou et al. [12] further explore the label-enhanced prototype to alleviate potential representation bias of the entity class. However, label and token representations are learned separately and combined via weighted average, where label embeddings are obtained without context, which fails to learn precise label representations and hence cannot characterize the class sufficiently. Yang et al. [33] argue that tokens labeled O have no unified semantic meaning, and the learned prototype of the class O is mixed with noise. Nevertheless, it ignores the fact that the combination of all tokens gives the named entities specific meaning and individual tokens in named entities can hardly be used to represent the class. Thus, it only benefits the non-entity class O and falls short for entity classes.

There are also other types of approaches. While some cross-lingual enhanced [8,32] and cross-domain enhanced [35,24] methods aim to transfer the capability obtained in high resource to low resource, templates-based NER [3] follows templates-based NLP [25,26] and treats NER as a language model problem by ranking sentences filled by candidate named entity spans. There is also undefined class augmented method [30] that mines the trend of clustering in `O` class to better represent non-entity class.

In this paper, we treat few-shot NER as a sequence labeling problem and tackles it with metric-based meta-learning. Our proposed method differs from existing literature mainly in the following three aspects: 1) Instead of using a single prototype to represent the class, we construct a multi-prototype for each class; 2) We also insert labels into supporting sentences, thereby enhancing the accuracy of the class representation; 3) We use different representation construction methods according to the characteristics of entity classes and non-entity classes.

Evaluation setting of few-shot NER. The evaluation of few-shot NER follows the popular experimental setting in few-shot learning, i.e., iterative N-way K-shot episodic sampling. Nevertheless, such iterative episodic few-shot NER training and testing suffers from the issue in episode construction. Specifically, the sentence-level sampling can cause the inconsistency of shots in different classes. Li et al. [15] and Yang et al. [33] use greedy-based sampling strategy to build up a support set that satisfies the strict K-shot setting. Nevertheless, such a strategy cuts down the sampling space and increases the sampling time due to strict restrictions. Fritzler et al. [9] and Tong et al. [29] only ensure there are at least K entities for each class. Regrettably, these simple restrictions cause serious deviation of the average shot from the original setting K. Hou et al. [12] approximately construct K-shot support set by the minimum-including algorithm, which may lead to a particularly high frequency of certain classes. By converting K-shot into $K \sim 2K$-shot, Ding et al. [6] alleviate the problems of all the above strategies at the same time. But $2K$ is still a relatively loose upper limit, especially when K is large.

In this work, compared with current evaluation settings, we put forward a more rigorous sampling strategy for few-shot NER. Specifically, it adopts an upper limit $2K$ to avoid sampling too many entities for popular classes and optimizes the sampling results by deleting extra samples. As such, the average shot of each class becomes as close as possible to the original setting K.

3 Problem Formulation and Setup

In this section, we introduce the problem settings of NER and few-shot NER. Next, we point out issues in existing few-shot NER evaluation settings and propose a more rigorous and realistic sampling strategy.

3.1 Named Entity Recognition

In this paper, we follow previous works [9,12,33,6,29] and formulate NER as a sequence labeling problem. Thus, the sentences in the original NER problem can be regarded as sequences of tokens [1]. Formally, given a sequence of tokens $\boldsymbol{x} = \{x_1, x_2, \cdots, x_n\}, x_i \in \mathcal{X}, i \in [1, n]$, where $\mathcal{X}$ is the set of all tokens, a sequence labeling classification model assigns a label $y_i \in \mathcal{C}$ to x_i, producing $\boldsymbol{y} = \{y_1, y_2, \cdots, y_n\}$, where $\mathcal{C}$ is the set of pre-defined classes. Correspondingly, the $(\boldsymbol{x}, \boldsymbol{y})$ pair represents a sentence and its label sequence. Notably, $\mathcal{C}$ can be split into a collection of entity classes $\mathcal{C}^+$ (e.g., `Film, Person`) and non-entity class `Outside` (shortened as `O`). An **entity** is a span of tokens belonging to the same entity class, and the label of an entity is its corresponding entity class.

3.2 Few-Shot Named Entity Recognition

We use the common iterative N-way K-shot episodic few-shot NER training paradigm [6,12]. Given class set $\mathcal{C}_{train} = \{c_i\}_{i=1}^{N}, c_i \in \mathcal{C}^+$ (***N*-way**), dataset $\mathcal{D}$ (all $(\boldsymbol{x}, \boldsymbol{y})$), for each step in training, one **episode** ($\mathcal{S}_{train}$ and $\mathcal{Q}_{train}$) is sampled to train model. Specifically, $\mathcal{S}_{train} = \{(\boldsymbol{x}, \boldsymbol{y})^{(i)}\}_{i=1}^{N_s}$ is the support set, $\mathcal{Q}_{train} = \{(\boldsymbol{x}, \boldsymbol{y})^{(j)}\}_{j=1}^{N_q}$ is the query set, and $\mathcal{S}_{train} \cap \mathcal{Q}_{train} = \emptyset$, Notably, in $\mathcal{S}_{train}$, for each class c_i, it is required that the number of entities labeled as c_i equals to K (***K*-shot**).

Models are trained on $\mathcal{Q}_{train}$ (i.e., predicting $\boldsymbol{y}$ given $\boldsymbol{x}$) with $\mathcal{S}_{train}$ as reference. All information in $\mathcal{S}_{train}$ and $\mathcal{Q}_{train}$ is available to models in training. In test phase, $\mathcal{S}_{test}$ and $\mathcal{Q}_{test}$ are constructed in the same way as in the training, except that the training and test class sets are disjoint, i.e., $\mathcal{C}_{test} \cap \mathcal{C}_{train} = \emptyset$. The **target** of few-shot NER is to predict $\boldsymbol{y}$ given $\boldsymbol{x}$ in $\mathcal{Q}_{test}$ using the trained model and $\mathcal{S}_{test}$.

3.3 Sampling Strategies

Issues with existing episodic sampling. Typically, to evaluate the few-shot learning models, N-way K-shot episodic sampling is adopted, where each episode (including support set and query set) is sampled from the original training and testing data, and the support set involves K examples for each of the N classes. To ensure the context integrity for few-shot NER, current methods conduct *sentence-level* sampling to construct each episode. Nevertheless, since each sentence contains varying numbers of entities, it is difficult for the sentence-level sampling to *accurately* satisfy the constraint of N-way K-shot. For instance, in Figure 1, the support sentence will never be selected under the *strict* 2-way 1-shot setting since there are 3 entities for the `Film` class.

[1] In the rest of the paper, we may use the word "sequence" to refer to "sentence".

To address this issue, existing methods [9,29,12,6] loose the restriction on the K value and only requires that the final number of examples (K^*) for each entity class is above K [2]. For example, in Ding et al. [6], a greedy sampling strategy is proposed to ensure that $K \leq K^* \leq 2K$. Nevertheless, this would lower the difficulty of this task since too many samples ($K^* > K$) are provided for each class.

Algorithm 1: Rigorous sampling algorithm.

Input : Dataset $\mathcal{D}$, N-class set C_N, N, K
Output : Support set S

$S \leftarrow \emptyset$;
for c in C_N **do**
  $Count_c = 0$;
while $\exists Count_c < K, c \in C_N$ **do**
  Randomly sample $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}$; $S^{temp} \leftarrow S \cup (\boldsymbol{x}, \boldsymbol{y})$;
  Compute $Count^{temp}$ with S^{temp};
  if $\exists Count_c^{temp} > 2K, c \in C_N$ **then**
    Continue
  else
    $S \leftarrow S^{temp}$; $Count \leftarrow Count^{temp}$
for $(\boldsymbol{x}, \boldsymbol{y})$ in S **do**
  $S^{temp} \leftarrow S - (\boldsymbol{x}, \boldsymbol{y})$; Compute $Count^{temp}$ with S^{temp};
  if $\exists Count_c^{temp} < K, c \in C_N$ **then**
    Continue
  else
    $S \leftarrow S^{temp}$; $Count \leftarrow Count^{temp}$
return S;

Our proposed rigorous sampling strategy. To make the evaluation conditions more similar to the original N-way K-shot setting, in this work, we propose a rigorous sampling algorithm to keep the average K^* value close to the setting K. Given an N-class set $\mathcal{C}_N \subset \mathcal{C}^+$, dataset $\mathcal{D}$, N, and K, we aim to sample support set $\mathcal{S}$. Specifically, we randomly sample $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}$ iteratively until the following condition is met: $\forall c \in \mathcal{C}_N$, the number of entities labeled as c is within the range $[K, 2K]$. Next, we delete the $(\boldsymbol{x}, \boldsymbol{y})$ pairs in S by following the criteria: the shot of any class will not be less than K because of removing $(\boldsymbol{x}, \boldsymbol{y})$ pairs from S. Finally, we end sampling when no $(\boldsymbol{x}, \boldsymbol{y})$ pair in S can be deleted. Algorithm 1 shows the detailed sampling process.

Our rigorous sampling algorithm can prevent sampling fluctuations caused by unbalanced class distribution and provide an evaluation condition that is much closer to the original N-way K-shot setting. This is because we adopt an upper limit $2K$ to avoid sampling too many entities for some popular classes.

[2] In this work, we denote these relaxed few-shot settings as N-way $\tilde{K}$-shot. The actual average K value is denoted as K^*.

More importantly, we further optimize the sampling results by deleting extra samples so that the average shot of each class becomes as close as possible to the setting K. We empirically validate that our rigorous sampling strategy is more reasonable and realistic than existing ones in Table 2 in Section 5. Further evaluations on few-shot NER also reveal that our strategy is able to reduce the performance inflation of previous models and provide a fairer evaluation condition.

4 Hybrid Multi-Prototype Learning for Few-shot NER

In this section, we first introduce the framework of HMP. Then we elaborate the design details. Finally, we describe the training and inference process.

4.1 Framework Overview

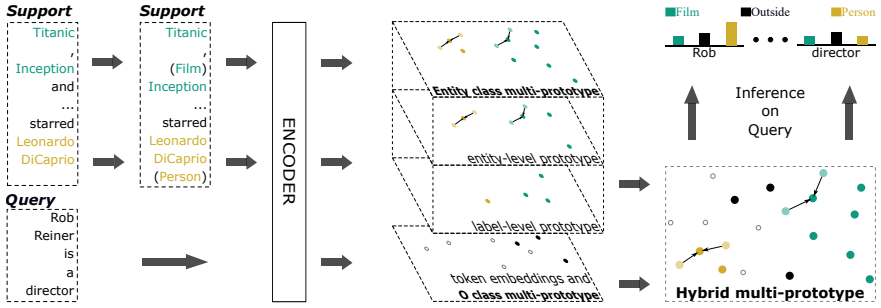


Fig. 2: Overview of Hybrid Multi-Prototype Few-Shot NER framework.

Figure 2 shows the framework of HMP. It takes episodes as input and outputs label sequences for tokens in the query set. The procedure can be divided into four stages: (1) For the support set, HMP first expands the sequences with entity labels to enrich the semantic meanings of sentences; Next, (2) HMP employs an encoder to embed the sequences in both support and query sets, producing contextual token and label embeddings; Afterwards, (3) based on the contextual embeddings, HMP generates *hybrid multi-prototype* representation for all classes; Finally, (4) HMP predicts the label for tokens in query set based on their distance with hybrid multi-prototype class representations.

4.2 Sequence Expansion and Embeddings

First, we propose to insert the entity labels into the original sequences in the support set to obtain *expanded* sequences. This is motivated by the fact

that the labels can enrich the sentence semantics and help learn more precise semantic embeddings for tokens. For instance, without sequence expansion, the token `Titanic` could refer to both the ship and the film, while its semantic meaning becomes much clearer by enriching the sentence with the label `Film`. Specifically, for each class, we only insert the labels after α entity spans. This is because inserting too many class labels could hurt the original meanings of sentences. In Section 5.6, we will discuss the influence of hyperparameter α on overall results.

Formally, given $(\boldsymbol{x}, \boldsymbol{y})$ in the support set, we denote the expanded sequence set as $\boldsymbol{x}' = \{x_1, x_2, y_1, \cdots, x_n, y_k\}$ that consists of n tokens and k inserted labels. Notably, our approach could make better use of the label information by leveraging it to guide the token embedding learning, while Hou et al. [12] fails to model such interactions between tokens and labels.

Next, we forward the expanded sequence set into an encoder, i.e., BERT [5], to obtain the *contextual* embeddings of tokens and labels. Specifically, the contextual embeddings of the expanded sequence set are:

$$\hat{\boldsymbol{x}}' = f_\theta(\boldsymbol{x}') = \{\hat{x}_1, \hat{x}_2, \hat{y}_1, \cdots, \hat{x}_n, \hat{y}_k\}, \tag{1}$$

where $f_\theta(\cdot)$ is the encoder and $\hat{\cdot}$ denotes the embedding.

4.3 Hybrid Multi-Prototype Representation

Given the contextual embeddings, we aim to generate the hybrid multi-prototype representation for classes using the support set. We use *hybrid* to highlight that we devise different approaches to handle entity classes and outside class, respectively, according to their specific characteristics.

Multi-prototype for entity classes. Given an entity class $c \in \mathcal{C}^+$, we use $\mathcal{E}_c$ to denote the entities in the support set that are labeled as c. For each entity $e \in \mathcal{E}_c$, we denote its *entity-level* prototype $\boldsymbol{e}_e$ as the averaged contextual embeddings of its tokens $\{x_1, x_2, \cdots, x_{|e|}\}$, and its *label-level* prototype $\boldsymbol{l}_e$ as the contextual embedding of its label y_e. Hence, the entity-level and label-level prototypes of all the entities in this class constitute the multi-prototype representation $\boldsymbol{c}$ of this entity class:

$$\boldsymbol{c} = \bigcup_{e \in \mathcal{E}_c} \{\boldsymbol{e}_e, \boldsymbol{l}_e\}. \tag{2}$$

Our multi-prototype can mitigate possible representation bias and improve the expression ability of single-prototype approaches. By calculating the entity-level prototype, we tackle the problem that individual tokens in the entities can hardly represent the corresponding class, and obtain a specific class representation. At the same time, we leverage the label-level prototype to improve the generalization ability and the expressiveness of the model because of the general information of class contained in label. Notably, it is difficult to fully represent the class with only a few examples, hence single-prototype can be

biased towards majority entities. Fortunately, our multi-prototype representation method can retain the representation of minority entities. We empirically validate the effectiveness of such designs in Table 8.

Multi-prototype for outside class. For the class `O`, following Yang et al. [33], we use token embeddings to represent the class, where the *multi-prototype* refers to the multiple token embeddings. Let $\mathcal{O} = \{x_1, x_2, \cdots, x_o\}$ be the tokens in the support set that are labeled as `O`. Then the multi-prototype representation of outside class is: $\boldsymbol{o} = \{\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_o\}$.

The multiple token embeddings alleviate the issue that the single-prototype cannot represent tokens with no uniform meaning in the class `O`. We empirically validate the effectiveness of such designs in Table 8 in Section 5.4.

4.4 Inference in the Query Set

Given a query sequence $\boldsymbol{x}^q = \{x_1^q, x_2^q, \cdots, x_n^q\}$ and their token embeddings $\hat{\boldsymbol{x}}^q = f_\theta(\boldsymbol{x}^q) = \{\hat{x}_1^q, \hat{x}_2^q, \cdots, \hat{x}_n^q\}$. To predict the label for token x_i^q, we first calculate the probability that it belongs to each class, and then consider the class with the highest probability as y_i^q. Specifically, the probability of token x_i^q belonging to class c is computed by:

$$\begin{aligned} p(c|x_i^q) &= \frac{\exp(-mindis(\hat{x}_i^q, \boldsymbol{c}))}{\sum_j \exp(-mindis(\hat{x}_i^q, \boldsymbol{c}_j))}, \\ mindis(\hat{x}_i^q, \boldsymbol{c}) &= \min_{c' \in \boldsymbol{c}} ||\hat{x}_i^q, c'||_2^2, \end{aligned} \tag{3}$$

where *mindis* denotes the *minimum* distance between the token embedding and the multi-prototype representations of this class (defined in Equation 2). In this work, we use the squared Euclidean distance as the distance measure. Note that the lower the *mindis*, the higher the probability.

Besides, label prediction is a sequential process, where label dependence could affect the results. For example, the label `Education` has a lower probability of appearing behind the label `Airport`. Therefore, we follow Yang et al. [33] by adopting an additional train-free Viterbi decoder to handle dependencies between labels and make more accurate predictions.

Training. In the training phase, we use the negative log likelihood loss to update the parameters in the encoder:

$$L = -\frac{1}{N} \sum_{i=1}^{N} \log(p(y_i^q = c_g | x_i^q)), \tag{4}$$

where c_g denotes the gold label, p represents the probability calculated using Equation 3 and N denotes the total number of tokens in the query set of the training data.

Table 1: Dataset statistics.

Dataset	Domain	# Class	# Sentence	# Entity
Few-NERD	General	66	188.2k	491.7k
WNUT	Social	6	5690	3890
re3d	Security	11	4697	3135

5 Experiments

In this section, we first introduce the datasets and compare the sampling strategies for generating the training data for few-shot NER. Next, we detail the specific experimental settings. Afterwards, we introduce the main evaluation results and then the ablation results. Finally, we present the case study and hyperparameter study.

5.1 Dataset and Sampling

Dataset. We use three datasets `Few-NERD` (containing `Few-NERD (INTER)` and `Few-NERD (INTRA)`) [6], `WNUT` [4], and `re3d` [3] to evaluate our proposed model. The statistics are shown in Table 1.

- `Few-NERD` is the first and only dataset specially constructed for few-shot NER with 8 coarse-grained and 66 fine-grained entity classes. Two few-shot NER subtasks, `INTER` and `INTRA`, are developed adopting different splitting strategies. For the former, the data is divided into different sets (train/dev/test) according to the fine-grained types of entities. For the latter, coarse-grained types are used to split the data, which means that there is very little shared knowledge between different sets. We compare HMP against state-of-the-art methods on this dataset.
- `WNUT` mainly focuses on named entities in the social domain. Unlike `Few-NERD`, which is collected on Wikipedia, `WNUT` comes from social platforms. While the sentences from Wikipedia have high qualities, e.g., correct grammatical structures and consistent spellings, the sentences on the social platforms have many issues such as incorrect syntactic structure, inconsistent spelling, more slang usage, and a large number of abbreviations. As thus, even supervised NER approaches cannot tackle `WNUT` very well. In this work, we use this dataset to assess the generalization capability of few-shot NER approaches.
- `re3d` is constructed using documents relevant to the defense and security analysis domain. Entities tend to be more specialized than the aforementioned two datasets. We also use this dataset to assess the generalization capability of few-shot NER approaches.

Sampling strategies. We report the final averaged shot value K^* generated by different sampling algorithms in Table 2, where 5000 support sets are

[3] `https://github.com/kotwanikunal/entity-recognition-datasets`

sampled from the testing set of `INTER`. Besides, we also define the *deviation degree*, to characterize the class-wise deviation from K:

$$D = \frac{1}{N_e} \sum_{1}^{N_e} \sqrt{\sum_{i=1}^{N} (K^i_{actual} - K)^2}, \tag{5}$$

where N_{iter} represents the number of episodes, N refers to the number of classes, and K^i_{actual} denotes actual shots of the i-th class in one episode. The deviation degree D can reflect the balance of the samples generated for each class, where an imbalanced distribution of samples would cause the value to become larger.

Table 2: Final average shot K^* and deviation degree D in support set under four settings. $N\text{w}\tilde{K}\text{s}$ refers to N-way $\tilde{K}$-shot.

Strategy	K^*				D			
	5w$\tilde{1}$s	**5w$\tilde{5}$s**	**10w$\tilde{1}$s**	**10w$\tilde{5}$s**	**5w$\tilde{1}$s**	**5w$\tilde{5}$s**	**10w$\tilde{1}$s**	**10w$\tilde{5}$s**
Fritzler et al. [9]	16.8	65.6	28.0	111.8	36.5	118.0	85.1	272.2
Ding et al. [6]	1.7	8.6	1.8	9.2	1.8	9.1	2.8	14.0
Hou et al. [12]	1.7	5.4	1.7	5.5	2.3	1.5	4.2	4.9
Ours	**1.3**	**5.2**	**1.3**	**5.2**	**0.9**	**0.7**	**1.4**	**1.3**

It reads from Table 2 that our rigorous sampling strategy keeps the average K^* value close to K, and attains a very low deviation degree, while other methods are apt to generate more samples for each class and have higher deviation degrees. Notably, although the average K^* value generated by Hou et al. [12] is close to ours, we discover that it is unstable and tends to generate large number of samples for certain classes, as can be observed from the deviation degrees. To further empirically validate this, we calculate the percentage of K^* values that are *larger* than $1.5K$ under 5-way $\tilde{1}$-shot and 10-way $\tilde{1}$-shot settings. The results reveal that, the percentages of these large K^* values are very small in our sampling strategy (10% and 0%, respectively), while these figures for Hou et al. [12] are 50% and 60%, respectively.

In all, the analysis above demonstrates that our proposed sampling strategy is the closest to the original N-way K-shot setting.

5.2 Experimental Setting

We test the model with episodic evaluation, a widely adopted evaluation method in few-shot learning, where we employ our rigorous sampling strategy to generate the support set and query set in the episode. All support sets are sampled under the $\tilde{K}$-shot constraint, but all query sets are sampled satisfying the same $\tilde{1}$-shot setting. 15,000 episodes are sampled for training, while 5,000 for testing. Note that the class sets of training and testing data are disjoint.

Besides, since the classes of Few-NERD have two hierarchies, we utilize both coarse-grained and fine-grained labels in INTER, while only fine-grained labels in INTRA, to calculate label-level prototypes. That is, we calculate two label-level prototypes for each entity on INTER. For WNUT and re3d, we only use the test set of these datasets to evaluate our model which is trained on Few-NERD.

Implementation details. We adopt uncased BERT-Base[4] as our backbone to obtain contextual representations of sequences and adopt the best hyperparameter values reported by Ding et al. [6][5]. We use PyTorch[6] to implement our models and all of them can be fit into V100 GPU with 32G memory. The training under each setting lasts for hours, and exhibits similar efficiency empirically to state-of-the-art methods. We set the hyperparameter α to 1 and provide further discussion in Section 5.6.

Metrics. All experiments are repeated five times with different random seeds, and the mean and standard deviation of the precision (P), recall (R), and micro F1 are calculated. We report the F1 of all experiments, and selectively report P and R following the settings of the previous studies.

Baselines. Four competitive models are used in our experiments. ProtoNER [9] employs Prototype Network to calculate prototypes for each class and classifies tokens by the similarity with prototypes. LTC [12] uses label representation to improve the prototype quality and considers label dependencies. We replace the Viterbi decoder used in Struct [33] to focus on comparing prototype representations. NNShot [33] directly uses the similarity between tokens to classify queries. Struct [33] improves NNShot with Viterbi decoder to obtain the most likely label sequence.

5.3 Overall Performance

The overall performance of the models on Few-NERD are summarized in Table 3, 4, 5, and 6 respectively. It can be observed that HMP consistently outperforms state-of-the-art models across all evaluation settings. Next, we analyze the results in detail.

On prototype construction. Our hybrid strategy of building multi-prototype by class characteristics benefits the performance. From the results, it is obvious that, under each setting, the R of ProtoNER is higher than the P, while LTC is just the opposite and even the P of LTC is higher than that of other models. This is because, ProtoNER tends to predict tokens as the entity class due to its noisy prototype of the non-entity class O. As thus, much more false positives emerge, resulting in a larger R (number of correct entities / number of gold entities) and a lower P (number of correct entities / number of predicted entities). On the contrary, the number of entities predicted by

[4] https://huggingface.co/

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

[6] https://pytorch.org/

Table 3: Performance on `Few-NERD` under 5-way settings. † indicates results from Ding et al. [6]. K^* denotes average shot. *Loose Samp.* denotes sampling strategy adopted by Ding et al. [6], and *Rigorous Samp.* is our sampling method. The best results are in **bold**.

Model	Few-NERD (INTER)					
	5-way $\tilde{1}$-shot			5-way $\tilde{5}$-shot		
	P	R	F1	P	R	F1
Loose Samp.		$K^* = 1.7$			$K^* = 8.6$	
ProtoNER †	39.0±0.0	51.7±0.3	44.4±0.1	53.7±1.8	**65.0±2.2**	58.8±1.4
LTC	**68.1±0.4**	38.7±0.8	49.3±0.6	**68.0±0.1**	42.5±0.9	52.3±0.7
NNShot †	50.4±0.6	58.8±0.1	54.3±0.4	45.8±3.5	56.5±2.9	50.6±3.3
Struct †	58.1±1.0	56.6±1.5	57.3±0.6	60.4±0.3	54.4±3.5	57.2±2.1
HMP	58.8±0.8	**61.5±1.3**	**60.1±0.7**	59.3±0.5	61.6±0.8	**60.4±0.6**
Rigorous Samp.		$K^* = 1.3$			$K^* = 5.2$	
ProtoNER	32.1±0.6	49.6±0.3	39.0±0.5	41.9±0.8	**65.9±8.0**	50.0±0.8
LTC	**68.1±0.3**	40.9±0.2	51.1±0.1	**67.0±0.1**	46.4±0.3	54.8±0.2
NNShot	43.4±1.3	53.2±0.9	47.8±1.1	45.9±1.8	58.8±1.6	51.6±1.7
Struct	53.2±0.9	52.5±1.8	52.8±0.9	54.8±1.3	57.3±2.3	56.0±1.3
HMP	56.5±2.0	**57.7±3.0**	**56.1±2.5**	59.0±0.8	63.4±0.6	**61.1±0.7**

Table 4: Performance on `Few-NERD` under 10-way settings. † indicates results from Ding et al. [6]. K^* denotes average shot. *Loose Samp.* denotes sampling strategy adopted by Ding et al. [6], and *Rigorous Samp.* is our sampling method. The best results are in **bold**.

Model	Few-NERD (INTER)					
	10-way $\tilde{1}$-shot			10-way $\tilde{5}$-shot		
	P	R	F1	P	R	F1
Loose Samp.		$K^* = 1.8$			$K^* = 9.2$	
ProtoNER †	32.6±0.2	48.9±2.9	39.1±0.9	47.9±0.5	**61.8±1.7**	54.0±0.4
LTC	**66.2±0.2**	36.6±0.2	47.1±0.2	**67.1±0.1**	41.1±0.4	51.0±0.3
NNShot †	42.7±2.1	52.2±1.8	47.0±2.0	45.2±0.8	56.1±0.4	50.0±0.4
Struct †	52.8±0.3	46.6±0.9	49.5±0.5	58.0±0.9	43.0±2.2	49.4±1.8
HMP	53.7±0.8	**55.5±1.4**	**54.6±0.9**	57.1±0.8	58.9±1.5	**58.0±0.5**
Rigorous Samp.		$K^* = 1.3$			$K^* = 5.2$	
ProtoNER	25.8±1.0	42.7±1.0	32.2±0.9	38.0±0.5	56.3±1.3	45.4±0.7
LTC	**66.4±0.4**	35.1±0.6	46.0±0.6	**67.2±0.5**	36.5±0.4	47.3±0.2
NNShot	35.1±2.1	45.0±2.7	39.4±2.3	37.0±0.4	49.3±1.5	42.3±0.7
Struct	46.7±0.6	41.4±2.1	43.9±1.4	52.1±1.0	43.4±1.6	47.3±1.4
HMP	50.5±1.1	**48.9±2.3**	**49.7±1.7**	56.1±0.8	**56.6±1.7**	**56.3±0.9**

LTC decreases due to the weighted average label and token embeddings and Viterbi decoder. Specifically, the separately obtained label embeddings, which cannot accurately represent the corresponding class, and the label dependency introduced by Viterbi decoder both make the tokens in the query are difficult to be identified as entity class. Besides, NNShot alleviates the issue of ProtoNER by using token embeddings to represent corresponding class. But due to ambiguous entity class representations, NNShot has a limited improvement in performance. Also, due to Viterbi decoder, Struct get an increase in F1 score. Different from these approaches, our model benefits from the hybrid

Table 5: Performance on `Few-NERD` under 5-way settings. † indicates results from Ding et al. [6]. K^* denotes average shot. *Loose Samp.* denotes sampling strategy adopted by Ding et al. [6], and *Rigorous Samp.* is our sampling method. The best results are in **bold**.

Model	Few-NERD (INTRA)					
	5-way 1̃-shot			5-way 5̃-shot		
	P	R	F1	P	R	F1
Loose Samp.	$K^* = 1.7$			$K^* = 8.5$		
ProtoNER †	18.6±1.0	31.8±1.0	23.5±0.9	35.9±0.7	**50.5±1.9**	41.9±0.6
LTC	**54.6±1.0**	14.4±0.7	22.6±0.8	**61.2±0.6**	22.7±0.4	33.2±0.3
NNShot †	29.0±1.0	33.4±1.4	31.0±1.2	32.9±2.5	39.2±2.2	35.7±2.4
Struct †	37.8±1.1	34.3±0.3	35.9±0.7	48.0±1.4	32.7±2.6	38.8±1.7
HMP	46.1±0.9	**34.7±1.2**	**39.5±0.7**	48.6±0.9	42.6±1.0	**45.4±0.7**
Rigorous Samp.	$K^* = 1.2$			$K^* = 5.2$		
ProtoNER	14.4±0.6	31.1±0.9	19.7±0.7	28.2±0.6	**49.4±1.4**	35.9±0.6
LTC	**55.4±0.3**	21.3±0.2	30.8±0.3	**60.1±0.5**	22.0±1.0	32.2±1.2
NNShot	24.2±0.6	29.1±0.9	26.4±0.7	30.1±1.6	39.1±1.6	34.0±1.5
Struct	34.2±2.2	29.0±2.0	31.0±1.5	46.2±3.3	39.3±1.5	42.4±1.9
HMP	46.5±0.9	**35.0±2.2**	**39.9±1.4**	47.7±0.8	44.3±1.3	**45.9±1.0**

Table 6: Performance on `Few-NERD` under 10-way settings. † indicates results from Ding et al. [6]. K^* denotes average shot. *Loose Samp.* denotes sampling strategy adopted by Ding et al. [6], and *Rigorous Samp.* is our sampling method. The best results are in **bold**.

Model	Few-NERD (INTRA)					
	10-way 1̃-shot			10-way 5̃-shot		
	P	R	F1	P	R	F1
Loose Samp.	$K^* = 1.8$			$K^* = 9.1$		
ProtoNER †	16.5±0.5	24.6±0.7	19.8±0.6	28.9±0.8	**43.1±0.8**	34.6±0.6
LTC	**52.0±0.2**	15.6±0.8	24.0±0.9	**56.6±0.6**	17.5±0.2	26.7±0.2
NNShot †	20.4±0.2	23.6±0.5	21.9±0.2	25.5±0.6	30.3±1.7	27.7±1.1
Struct †	29.9±1.1	22.0±0.7	25.4±0.8	40.6±2.2	19.6±2.7	26.4±2.6
HMP	38.7±0.9	**30.2±1.7**	**33.9±0.7**	47.4±1.3	38.6±1.7	**42.5±0.8**
Rigorous Samp.	$K^* = 1.2$			$K^* = 5.2$		
ProtoNER	11.8±0.3	22.9±0.3	15.6±0.3	22.3±0.6	**41.0±1.2**	28.9±0.6
LTC	**53.9±1.0**	16.7±0.5	25.5±0.5	**57.2±0.9**	15.5±0.7	24.4±0.8
NNShot	16.9±0.8	22.0±0.5	19.1±0.6	22.7±0.7	30.0±1.1	25.8±0.8
Struct	26.8±1.0	21.7±2.0	24.0±1.5	40.4±0.9	25.1±1.9	30.9±1.5
HMP	40.8±0.7	**29.5±0.8**	**34.2±0.5**	43.8±3.8	37.0±5.7	**39.6±2.1**

strategy of building multi-prototype by class characteristics, obtains precise class representations, and thus achieving state-of-the-art performance.

On sampling strategies. Our rigorous sampling strategy provides a fairer experimental setting. From the tables, it is obvious that the performance of the models trained with loose sampling strategy is generally better than that trained with rigorous one. However, this overall performance inflation benefits from much more support examples in the setting. In other words, loose sampling strategies lower the difficulty of few-shot NER task. In contrast, our sampling strategy effectively limits the average shot near the original setting, thus the distribution of entities with different characteristics in the same class

is more difficult to learn and finally reduces the performance inflation of previous few-shot NER models. In this connection, it could be considered as a fairer experimental setting for few-shot NER task.

Table 7: F1 on `WNUT` and `re3d` using episodic evaluation. For `WNUT`, † indicates results from Yang et al. [33]. For `re3d`, † indicates results from Tong et al. [29]. The best results are in **bold**.

Model	WNUT		re3d	
	$\tilde{1}$-shot	$\tilde{5}$-shot	$\tilde{1}$-shot	$\tilde{5}$-shot
ProtoNER †	15.8±4.1	22.1±3.1	26.8	27.8
LTC †	17.5±2.9	26.0±2.1	23.3	35.8
NNShot †	20.2±6.0	26.7±4.0	-	-
Struct †	20.5±5.2	27.9±3.2	-	-
ProtoNER	16.3±0.6	22.5±0.8	25.5±0.8	36.2±2.2
LTC	20.2±0.8	28.3±1.5	21.8±1.8	33.1±2.7
NNShot	18.6±0.6	23.5±1.1	35.8±5.5	47.3±3.1
Struct	19.7±0.1	22.6±0.8	36.3±1.9	48.9±1.2
HMP (ours)	**31.5±0.7**	**33.3±0.9**	**45.6±3.6**	**49.8±4.5**

On generalization ability. Our model shows good generalization ability in other domains. We conduct $\tilde{1}$-shot and $\tilde{5}$-shot experiments on the test sets of `WNUT` and `re3d`, using the trained model on `Few-NERD`. Table 7 shows the results, where our model still achieves state-of-the-art performance.

5.4 Ablation Study

Table 8: F1 over different components on `INTER`. *-viterbi* indicates without Viterbi decoder. *-expansion* indicates without sequence expansion. *-label-level prototypes* indicates using sequence expansion but without label-level prototypes. *prototype→token* indicates using tokens to represent entity class. *multiple→single* indicates using single-prototype to represent entity class. *token→prototype* indicates using single-prototype to represent non-entity class.

Model	5-way $\tilde{5}$-shot	10-way $\tilde{5}$-shot
HMP	**61.1**	**56.3**
-viterbi	-5.0	-5.6
-expansion	-4.3	-3.4
-label-level prototypes	-2.5	-2.7
prototype→token	-4.2	-7.1
multiple→single	-2.8	-2.0
HMP-E	**56.8**	**52.9**
prototype→token	-0.8	-5.6
multiple→single	-3.1	-2.1
token→prototype	-3.7	-5.4

To provide a deeper insight of each component in HMP, we conduct ablation analysis under 5-way $\tilde{5}$-shot and 10-way $\tilde{5}$-shot settings in Table 8. HMP-E denotes HMP trained without sequence expansion.

Usefulness of Viterbi decoder. The results of HMP without Viterbi decoder (*-viterbi*) show that capturing the dependencies between labels can improve performance. While the F1 score descends, the performance of our model is still superior to that of the state-of-the-art models (e.g., Struct with Viterbi decoder).

Usefulness of sequence expansion. The drop in F1 caused by removing the sequence expansion (*-expansion*) indicates that inserting the class labels into the support examples is beneficial to contextual embedding learning and hence the overall results.

Usefulness of label-level prototypes. In more detail, when using expanded sentences but without the corresponding label-level prototypes (*-label-level prototypes*), the performance falls between that of HMP-E (without the sentence expansion) and HMP, demonstrating the usefulness of the label-level prototypes, as well as sequence expansion.

Usefulness of multi-prototype. We further verify the effectiveness of multi-prototype from three aspects: *prototype→token*, *multiple→single* and *token→prototype*. **First**, instead of multi-prototype, we use tokens to represent entity class (*prototype→token*). The results confirm that the multi-prototype can characterize the entity class better than the tokens. **Second**, we delve deeper into different representations of prototypes, where the single-prototype is calculated for each entity class by averaging all tokens in the same class (*multiple→single*). The decreased performance of HMP and HMP-E shows that the multi-prototype can effectively improve the expressiveness of the single-prototype methods. **Third**, we use single-prototype to represent class O on the basis of the second step (*token→prototype*). The drop in results confirms that the multiple tokens is more suitable to represent class O than single-prototype.

The above ablation results demonstrate that our proposed hybrid multi-prototype construction is of great help to the improvement of performance.

5.5 Case Study

We conduct case study to show that our proposed multi-prototype construction method can generate less biased class representation, which in turn can bring more accurate few-shot NER results. Specifically, as shown in Figure 3, we select two typical cases from the dataset, with each case containing both partial support and query sets. Entities are identified with colors, and different colors represent different entity classes. The correct inference results on the query sentences are marked with **True** (following the sentence). The models that produce the inference results are appended to the corresponding sentences.

The results in Figure 3 can reflect the bias of the prototype construction of current methods. Specifically, ProtoNER and LTC incline to omit certain entities, e.g., the underlined craddock of query set 1 in Figure 3-(a), or part of the tokens in entity, e.g., the underlined matthias of query set 2 in Figure 3-(a). This is because they use the average of all token embeddings to represent

Part of Support Set	1.mackendrick got along poorly with the producers of the film . 2.carlos won the 200-meter dash in 19.92 seconds , beating world-record holder tommie smith .
Query Set 1	craddock made 83 appearances for the club . (**True**, HMP) craddock made 83 appearances for the club . (ProtoNER, LTC, NNShot, Struct)
Query Set 2	matthias hues was also cast as the new russian character . (**True**, HMP) matthias hues was also cast as the new russian character . (ProtoNER, LTC, Struct) matthias hues was also cast as the new russian character . (NNShot)

(a)

Part of Support Set	1.brett also stars in series 8 and 10 , outside of christine 's tenure as headteacher . 2.the stg 44 is generally considered the first selective fire military rifle to popularize the assault rifle concept .
Query Set 1	the armament of cm-12 is identical to cm-11 's . (**True**, HMP, ProtoNER, LTC) the armament of cm-12 is identical to cm-11 's . (NNShot, Struct)
Query Set 2	matthias hues was also cast as the new russian character . (**True**, HMP) matthias hues was also cast as the new russian character . (ProtoNER, LTC) matthias hues was also cast as the new russian character . (NNShot, Struct)

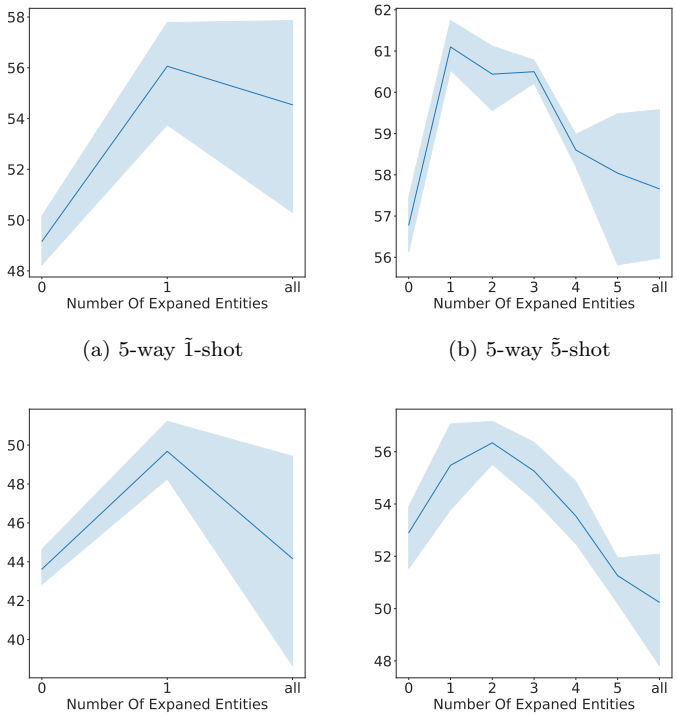
(b)

Fig. 3: Case study of bias in Few-shot NER.

each class, which lacks expressiveness and tends to be biased towards popular entities. Meanwhile, Struct and NNShot tend to predict more tokens as entities, e.g., the underlined 's (resp., russian) of query set 1 (resp., query set 2) in Figure 3-(b), since they use the set of token embeddings to represent each class, which includes many irrelevant tokens into the class representation and misleads the inference process. In contrast to previous methods, our proposal HMP can generate expressive and less biased prototypes by overcoming the aforementioned issues, leading to consistently better results.

5.6 Hyperparameter Study

We discuss hyperparameter α in the process of sequence expansion. We discover that inserting a certain number of (but not all) labels can be the most effective are consistent on all datasets. The F1 curve in Figure 4 confirms that the proper number of inserted labels has a great effect on performance. In most cases, inserting one label for each class can greatly improve the overall performance of the model.



(a) 5-way $\tilde{1}$-shot (b) 5-way $\tilde{5}$-shot

(c) 10-way $\tilde{1}$-shot (d) 10-way $\tilde{5}$-shot

Fig. 4: F1 over different number of expanded entities (controlled by α) on `INTER`.

6 Conclusion and future work

We propose a hybrid multi-prototype class representation approach for few-shot NER, which calculates multi-prototype to represent entity class and uses token embeddings to represent non-entity class. Extensive empirical results show that the hybrid prototype construction strategy and the multi-prototype strategy are of great help to the generation of less biased representations, which also leads to state-of-the-art few-shot NER performance. Moreover, we introduce a rigorous experimental setting for few-shot NER, which can provide a reasonable and fairer evaluation condition. We further demonstrate the adaptability of our model to corpus on social media on the `WNUT` dataset.

The theoretical implication of this work lies in two aspects: 1) The proposed hybrid multi-prototypes can effectively alleviate the representation bias caused by existing class representation approaches, and offer more expressive class representations for few-shot NER; 2) This article also verifies that different sampling methods do have a great impact on the final results, and a more

reasonable and rigorous experimental setting should be used to ensure fair comparison among different models.

The practical implication of this work is to offer a more effective approach for identifying named entities in a given text and classifying them into predefined entity classes with a few support examples, i.e., few-shot NER.

Furthermore, such advance in few-shot NER can reduce the dependence on manually annotated data, thereby accelerating the extraction of knowledge in emerging domains.

We hope our work can provide insights into the few-shot NER task. In the future, we plan to investigate how to generate more accurate class representations by an adaptive construction.

7 Declarations

Ethical Approval. This declaration is not applicable.

Competing interests. I declare that all authors have no competing interests as defned by Springer, or other interests that might be perceived to infuence the results and discussion reported in this paper.

Authors' contributions. Zenghua Liao and Junbo Fei wrote the main manuscript text and designed the methodology framework. Weixin Zeng and Xiang Zhao prepared experiment. All authors reviewed the manuscript.

Funding. This declaration is not applicable.

Availability of data and materials. We will release the source code at the final version.

References

1. Bai, L., Zhang, M., Zhang, H., Zhang, H.: Ftmf: Few-shot temporal knowledge graph completion based on meta-optimization and fault-tolerant mechanism. World Wide Web pp. 1–28 (2022)
2. Chiu, J.P.C., Nichols, E.: Named entity recognition with bidirectional lstm-cnns. Transactions of the Association for Computational Linguistics **4**, 357–370 (2016)
3. Cui, L., Wu, Y., Liu, J., Yang, S., Zhang, Y.: Template-based named entity recognition using BART. In: Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021, *Findings of ACL*, vol. ACL/IJCNLP 2021, pp. 1835–1845 (2021)
4. Derczynski, L., Nichols, E., van Erp, M., Limsopatham, N.: Results of the WNUT2017 shared task on novel and emerging entity recognition. In: Proceedings of the 3rd Workshop on Noisy User-generated Text, pp. 140–147. Copenhagen, Denmark (2017)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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), pp. 4171–4186. Minneapolis, Minnesota (2019)
6. Ding, N., Xu, G., Chen, Y., Wang, X., Han, X., Xie, P., Zheng, H., Liu, Z.: Few-NERD: A few-shot named entity recognition dataset. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International

Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 3198–3213. Online (2021)
7. Eberts, M., Pech, K., Ulges, A.: Manyent: A dataset for few-shot entity typing. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 5553–5557 (2020)
8. Feng, X., Feng, X., Qin, B., Feng, Z., Liu, T.: Improving low resource named entity recognition using cross-lingual knowledge transfer. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, pp. 4071–4077 (2018)
9. Fritzler, A., Logacheva, V., Kretov, M.: Few-shot classification in named entity recognition task. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC '19, p. 993–1000. New York, NY, USA (2019)
10. Gao, T., Han, X., Zhu, H., Liu, Z., Li, P., Sun, M., Zhou, J.: FewRel 2.0: Towards more challenging few-shot relation classification. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6250–6255. Hong Kong, China (2019)
11. Han, X., Zhu, H., Yu, P., Wang, Z., Yao, Y., Liu, Z., Sun, M.: FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4803–4809. Brussels, Belgium (2018)
12. Hou, Y., Che, W., Lai, Y., Zhou, Z., Liu, Y., Liu, H., Liu, T.: 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, pp. 1381–1393. Online (2020)
13. Huang, J., Li, C., Subudhi, K., Jose, D., Balakrishnan, S., Chen, W., Peng, B., Gao, J., Han, J.: Few-shot named entity recognition: An empirical baseline study. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 10408–10423 (2021)
14. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: 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, pp. 260–270. San Diego, California (2016)
15. Li, J., Chiu, B., Feng, S., Wang, H.: Few-shot named entity recognition via meta-learning. IEEE Transactions on Knowledge and Data Engineering pp. 1–1 (2020)
16. Li, J., Shang, S., Shao, L.: Metaner: Named entity recognition with meta-learning. In: Proceedings of The Web Conference 2020, pp. 429–440 (2020)
17. Li, M., Li, Z., Yang, Q., Chen, Z., Zhao, P., Zhao, L.: A crowd-efficient learning approach for ner based on online encyclopedia. World Wide Web **23**(1), 453–470 (2020)
18. Li, X., Yin, H., Zhou, K., Zhou, X.: Semi-supervised clustering with deep metric learning and graph embedding. World Wide Web **23**(2), 781–798 (2020)
19. Lin, S., Gao, J., Zhang, S., He, X., Sheng, Y., Chen, J.: A continuous learning method for recognizing named entities by integrating domain contextual relevance measurement and web farming mode of web intelligence. World Wide Web **23**(3), 1769–1790 (2020)
20. Liu, F., Mao, Q., Wang, L., Ruwa, N., Gou, J., Zhan, Y.: An emotion-based responding model for natural language conversation. World Wide Web **22**(2), 843–861 (2019)
21. Liu, K., Liu, W., Ma, H., Huang, W., Dong, X.: Generalized zero-shot learning for action recognition with web-scale video data. World Wide Web **22**(2), 807–824 (2019)
22. Ma, Y., Cambria, E., Gao, S.: Label embedding for zero-shot fine-grained named entity typing. In: COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan, pp. 171–180 (2016)
23. Miller, E., Matsakis, N., Viola, P.: Learning from one example through shared densities on transforms. In: Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662), vol. 1, pp. 464–471 vol.1 (2000)
24. Nguyen, H.V., Gelli, F., Poria, S.: DOZEN: cross-domain zero shot named entity recognition with knowledge graph. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1642–1646 (2021)

25. Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P.S.H., Bakhtin, A., Wu, Y., Miller, A.H.: Language models as knowledge bases? In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2463–2473 (2019)
26. Sun, C., Huang, L., Qiu, X.: Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. 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), pp. 380–385 (2019)
27. Sun, S., Sun, Q., Zhou, K., Lv, T.: Hierarchical attention prototypical networks for few-shot text classification. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pp. 476–485 (2019)
28. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, pp. 142–147 (2003)
29. Tong, M., Wang, S., Xu, B., Cao, Y., Liu, M., Hou, L., Li, J.: Learning from miscellaneous other-class words for few-shot named entity recognition. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 6236–6247. Online (2021)
30. Tong, M., Wang, S., Xu, B., Cao, Y., Liu, M., Hou, L., Li, J.: Learning from miscellaneous other-class words for few-shot named entity recognition. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, pp. 6236–6247 (2021)
31. Wen, W., Liu, Y., Ouyang, C., Lin, Q., Chung, T.: Enhanced prototypical network for few-shot relation extraction. Information Processing & Management **58**(4), 102596 (2021)
32. Xu, L., Zhang, X., Zhao, X., Chen, H., Chen, F., Choi, J.D.: Boosting cross-lingual transfer via self-learning with uncertainty estimation. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 6716–6723 (2021)
33. Yang, Y., Katiyar, A.: 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), pp. 6365–6375. Online (2020)
34. Yoon, S.W., Seo, J., Moon, J.: TapNet: Neural network augmented with task-adaptive projection for few-shot learning. In: Proceedings of the 36th International Conference on Machine Learning, *Proceedings of Machine Learning Research*, vol. 97, pp. 7115–7123 (2019)
35. Zhou, J.T., Zhang, H., Jin, D., Zhu, H., Fang, M., Goh, R.S.M., Kwok, K.: Dual adversarial neural transfer for low-resource named entity recognition. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pp. 3461–3471 (2019)