BANER: Boundary-Aware LLMs for Few-Shot Named Entity Recognition

Quanjiang Guo[1], **Yihong Dong**[1], **Ling Tian**[1], **Zhao Kang**[1*], **Yu Zhang**[2], **Sijie Wang**[3]
[1] University of Electronic Science and Technology of China, Chengdu, China
[2] Harbin Institute of Technology, Shenzhen, China
[3] Nanyang Technological University, Singapore
guochance1999@163.com, dongyihong8@163.com, lingtian@uestc.edu.cn, zkang@uestc.edu.cn, yuzhang2717@gmail.com, wang1679@e.ntu.edu.sg

Abstract

Despite the recent success of two-stage prototypical networks in few-shot named entity recognition (NER), challenges such as over/under-detected false spans in the span detection stage and unaligned entity prototypes in the type classification stage persist. Additionally, LLMs have not proven to be effective few-shot information extractors in general. In this paper, we propose an approach called **B**oundary-**A**ware LLMs for Few-Shot **N**amed **E**ntity **R**ecognition (BANER) to address these issues. We introduce a *boundary-aware contrastive learning strategy* to enhance the LLM's ability to perceive entity boundaries for generalized entity spans. Additionally, we utilize LoRAHub to align information from the target domain to the source domain, thereby enhancing adaptive cross-domain classification capabilities. Extensive experiments across various benchmarks demonstrate that our BANER framework outperforms prior methods, validating its effectiveness. In particular, the proposed strategies demonstrate effectiveness across a range of LLM architectures. [1]

1 Introduction

Named Entity Recognition (NER) is a fundamental task in Natural Language Processing (NLP) that aims to detect the entity spans of text and classify them into pre-defined set of entity types. When there are sufficient labeled data, deep learning-based methods (Huang et al., 2015; Ma and Hovy, 2016; Lample et al., 2016; Chiu and Nichols, 2016) have achieved impressive performance. However, in practical applications, it is desirable to recognize new entity types that have not been seen in the source domain. It is time-consuming and labor-expensive to collect extra labeled data for these new types. Consequently, few-shot NER (Fritzler et al., 2019; Yang and Katiyar, 2020), which involves identifying unseen entity types based on only a few labeled samples for each class (also known as *support samples*) in the target domain, has attracted a lot of attention in recent years.

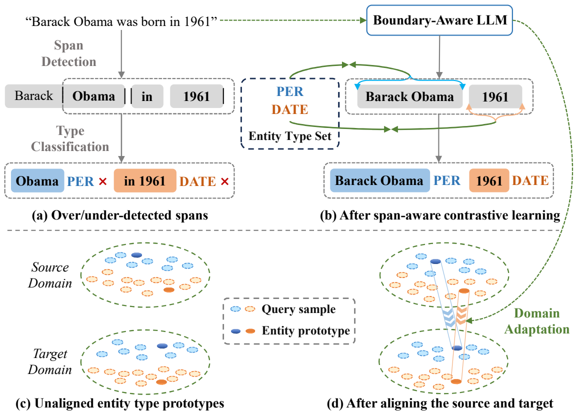

Figure 1: (a) shows under/over-detected false spans, (b) shows correct spans obtained by adopting our boundary-aware LLM, (c) shows unaligned entity type prototypes, (d) shows aligned prototypes obtained by our domain adaption strategy.

Previously end-to-end metric learning based methods (Yang and Katiyar, 2020; Das et al., 2022) dominate the field of few-shot NER. These approaches are designed to learn the intricate structure that includes both entity boundaries and entity types. However, their performance may degrade significantly when encountering a substantial domain gap. This degradation is primarily due to the challenge of understanding such complex structural information with only a few support examples for domain adaptation. Consequently, these methods often suffer from inadequate perception of boundary information, resulting in frequent misclassification of entity spans. Though LLMs have made remarkable success in various tasks, they have not proven to be effective few-shot information extractors in general (Ma et al., 2023; Zhang et al., 2024b).

*Corresponding Author
[1]The code and data are released on https://github.com/UESTC-GQJ/BANER.

Proceedings of the 31st International Conference on Computational Linguistics, pages 10375–10389
January 19–24, 2025.

Recent works demonstrate that adopting two-stage prototypical networks (Wang et al., 2022; Ma et al., 2022b; Li et al., 2023) can be effective to address aforementioned issue, which decompose NER task into two distinct stages: *entity span detection* and *entity type classification* tasks, executing each task sequentially. Since decomposed methods only need to locate the spans of named entities and are class-agnostic in the first stage, they can identify more accurate entity boundaries and achieve better performance than end-to-end approaches.

While these two-stage prototypical methods have shown promising progress, they also present two additional challenges. Firstly, at the entity span detection stage, these decomposed approaches merely detect possible spans, often overlooking the boundary-related semantic information of named entities. For instance, following entity span detection, the sentence in Figure 1(a) illustrates that the span for "Barack Obama" is inadequately detected, resulting in "Obama" being identified while "Barack" is overlooked. Conversely, the span for "1961" is excessively detected as "in 1961". These inaccuracies propagate errors into the subsequent entity type classification stage.

Secondly, in decomposed methods, prototypical networks aim to learn a type-related metric similarity function from test samples to classify entities based on their distance to prototypes. However, since the obtained prototypes are independently trained relative to the first stage, they may overlook entity type knowledge from the prior source domain. This can lead to difficulties in aligning the distribution of the same class across different domains. For example, in Figure 1(c), the entity types in the target domain exist independently of those in the source domain, leading to misaligned prototypes for the same entity type. This misalignment can severely impact the cross-domain performance of few-shot NER during the entity type classification stage.

To this end, we propose an approach called **B**oundary-**A**ware LLMs for Few-Shot **N**amed **E**ntity **R**ecognition (BANER). Our approach adopts the two-stage framework of the decomposed method but advances two steps further to effectively address the aforementioned challenges. For *entity span detection*, we design a boundary-aware contrastive learning strategy to reduce the gap between span embeddings of entities and their corresponding types using LLM. This strategy enhances the boundary perception capabilities of LLM, particularly for generalized entity spans. For *entity type classification*, we draw upon domain adaptation principles to construct refined prototypes that retain and align entity type knowledge from the source domain. This approach involves joint pretraining in the source domain and adaptive alignment across diverse target domains within the same LLM framework, facilitated by LoRAHub (Huang et al., 2023).

In summary, our contributions are as follows:

(1) We introduce a novel Few-Shot NER approach, BANER, which employs boundary-aware contrastive learning to enhance an LLM's ability to perceive entity boundaries. To our knowledge, this is the first integration of LLM with contrastive learning for few-shot NER tasks.

(2) Leveraging an LLM pretrained on the source domain, we utilize LoRAHub to align information from target domains to enhance adaptive cross-domain classification capabilities.

(3) Experimental results on multiple few-shot NER datasets demonstrate that BANER achieves superior performance compared to previous state-of-the-art two-stage decomposed methods. Furthermore, we validate the generalizability of our strategies across various LLM architectures.

2 Related Work

Few-Shot NER Recently, few-shot named entity recognition (NER) has garnered considerable attention. Previous methods can be broadly categorized into two types: prompt-based and metric-based approaches. Prompt-based methods focus on leveraging the knowledge of pre-trained language models (LLMs) for NER through prompt learning techniques (Cui et al., 2021; Ma et al., 2022a; Huang et al., 2022; Lee et al., 2022). These methods utilize templates, prompts, or exemplary instances to effectively harness the pre-existing knowledge within LLMs.

With the rapid advancements in LLMs, there has been a surge in studies exploring direct prompting of LLMs for few-shot NER tasks (Wang et al., 2023; Xie et al., 2023). Additionally, there is emerging interest in straightforward instruction-tuning strategies (Zhou et al., 2024), or annotating raw data with LLMs to train task-specific foundational models for NER (Zhang et al., 2024b). However, their performance often diminishes when tasked with generating text that adheres to specific

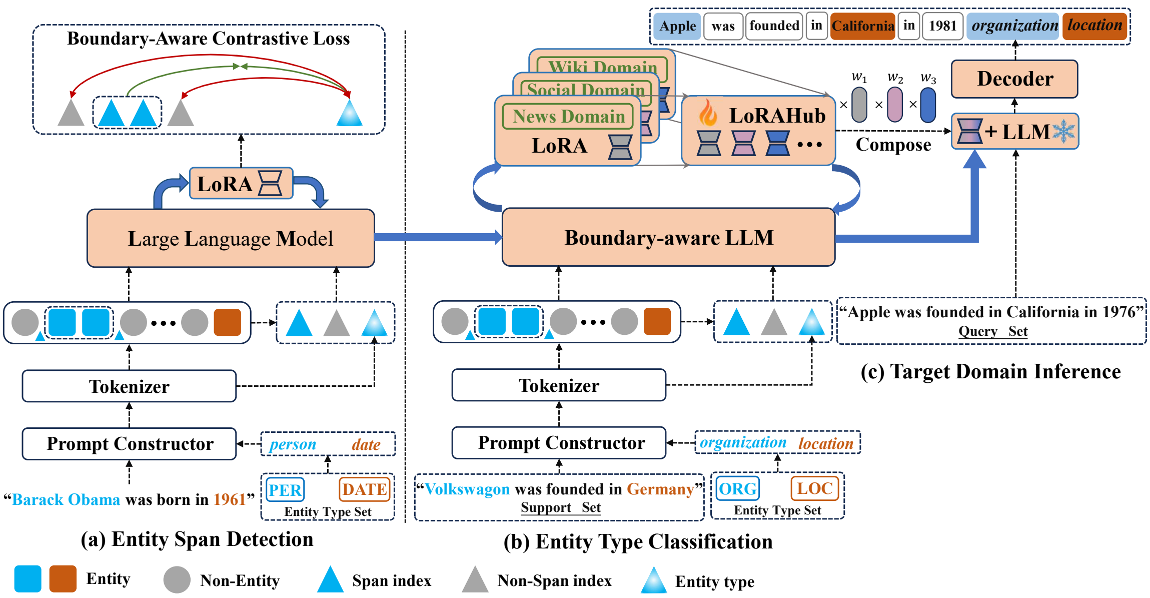

Figure 2: Overall structure of the proposed BANER. (a) Entity span detection with pre-training in the source domain. (b) Entity type classification with fine-tuning in the support samples of target domain. (c) Inference on the query set of target domain.

structured formats and domains, which is crucial in few-shot NER scenarios.

Metric-based methods, on the other hand, aim to learn a feature space with robust generalizability and classify test samples using nearest class prototypes (Snell et al., 2017; Fritzler et al., 2019; Ji et al., 2022; Ma et al., 2022b) or neighboring samples (Yang and Katiyar, 2020; Das et al., 2022). Nevertheless, the prototypical networks widely employed in these methods may fail to fully utilize entity type knowledge from the source domain during the type classification stage.

Moreover, recent research has focused on the two-stage architecture for few-shot named entity recognition (NER) (Shen et al., 2021; Wang et al., 2021; Zhang et al., 2022; Wang et al., 2022; Ma et al., 2022b; Li et al., 2023), where the task is decomposed into entity span detection and entity typing subtasks. These methods excel in learning entity boundary information under data-limited conditions and often achieve superior performance. However, they may encounter challenges such as over/under-detection of false entity spans during the span detection stage.

Contrastive Learning and Domain Adaptation Due to the robust generalization capabilities of contrastive learning, recent methods (Das et al., 2022; Huang et al., 2022) have adopted this approach for few-shot NER, employing contrastive losses between tokens or between tokens and prompts. However, these methods are end-to-end approaches and therefore inherently lack the ability to effectively learn entity boundary information. In contrast, our approach is decomposed, and our boundary-aware contrastive loss is designed between the span embeddings of entities and their corresponding types within the LLM framework. This method enables the learning of a span-aware feature space in LLMs, facilitating accurate boundary detection.

Domain adaptation tackles the challenge of dataset shift between source and target domains, particularly when only a few samples are available in the target domain. When labels in the target domain are scarce, the problem transitions into a semi-supervised scenario. Traditional approaches combine source and target data to enhance model training (Zhang et al., 2021; Zhang and Kang, 2024). In the context of type classification adaptation using LLMs, fine-tuning remains the predominant method (Grangier and Iter, 2022; Guo et al., 2021; Buonocore et al., 2023). Alternatively, strategies involve expanding the LLM's vocabulary with domain-specific tokens (Sachidananda et al., 2021; Zhu et al., 2024) or employing adversarial adaptation techniques such as knowledge distillation (Rietzler et al., 2020) or supervised fine-tuning (Ryu et al., 2022; Zhang et al., 2024a). In contrast, our approach leverages LoRAHub to dynamically align information from the target domain with that of the source domain.

3 Methodology

Figure 2 depicts the overall framework of our BANER. Like other two-stage methods, it comprises *entity span detection* and *entity type classification*. Notably, our approach incorporates boundary-aware contrastive learning and adaptive domain alignment strategies at these respective stages.

Task Formulation Given a sequence $X = \{x_i\}_{i=1}^{L}$ with L tokens, NER aims to assign each token x_i to its corresponding label $y_i \in Y \cup \mathrm{O}$, where Y is the pre-defined entity type set and O denotes non-entities. For few-shot NER, the NER model is first pretrained on data-sufficient source domain(s) $\mathcal{D}_s = \{(S_s, Q_s, Y_s)\}$ and then fine-tuned in target domain(s) $\mathcal{D}_t = \{(S_t, Q_t, Y_t)\}$ with only a few labeled samples. We adhere to the standard N-way K-shot setting as outlined in (Ding et al., 2021), where $S_{s/t} = \{(x_i, y_i)\}_{i=1}^{N\times K}$ denotes the support set, $Q_{s/t} = \{(x_j, y_j)\}_{j=1}^{N'\times K'}$ denotes the query set, $|Y_s| = N$ and $|Y_t| = N'$. Our task is to recognize entities in the query set Q_t from the target domain after adapting the model using its support set S_t. It is noteworthy that Y_s and Y_t exhibit little to no overlap.

3.1 Entity Span Detection

3.1.1 Prompt Representation

Formally, we denote the LLM as f_{LLM} and input instruction as I. The output (generated) token sequence is denoted as $Y = f_{\mathrm{LLM}}(X) = \{y_i\}_{i=1}^{L}$. For the classic auto-regressive generative model, the sampling probability of the model generating Y is formalized as follows:

$$\mathbb{P}(Y \mid I, X) = \prod_{t=1}^{L} \mathbb{P}(y_t \mid I, X, y_{<t}), \tag{1}$$

where y_t is the t-th token of the y, $y_{<t}$ represents the tokens before y_t. Utilizing generative language models for information extraction typically involves providing a prompt as input and generating results according to a specified format. In BANER, we adopt the default template for LoRA fine-tuning[2]. The prompt is fed into the LLM to perform entity span detection. An example of such a prompt is illustrated in Figure 5 in Appendix A.1.

According to the LLM's token generation rule, the objective loss for auto-regressively generating Y is as follows:

$$\mathcal{L}_g = -\sum_{(X,y)\in\mathcal{D}_s} \sum_{t=1}^{L} \log \mathbb{P}_{\theta+\theta_L}(y_t \mid I, X, y_{<t}), \tag{2}$$

where θ is the original parameters of LLM, θ_L is the LoRA parameters. Note that we only update LoRA parameters during the training process.

3.1.2 Boundary-Aware Contrastive Learning

We enumerate all m spans $S = \{s_1, s_2, ..., s_m\}$ for sequence X. For example, for sentence "Barack Obama was born in 1961", span indices (begin, end) of two entities are $\{(0, 2), (5, 6)\}$. We use b_i and e_i to denote the begin- and end- index representation of the span s_i in constructed prompt, respectively.

To enhance the LLM's ability to perceive entity boundaries, we employ the concept of contrastive learning (Khosla et al., 2020). We utilize two types of boundary-aware index representations, as illustrated in Figure 2(a), to construct positive and negative samples for each entity mention and its corresponding entity type. Specifically, the positive sample $\mathrm{pos_i}$ of entity span is calculated by concatenating h_{b_i} and h_{e_i-1} as $\mathrm{pos_i} = [h_{b_i}, h_{e_i-1}]$, where $h_{(\cdot)} = \mathrm{embedding}(\cdot)$ is the pre-trained tokenizer of LlaMA-2-7B. The negative sample neg_i of entity boundary is $\mathrm{neg}_i = [h_{b_i-1}, h_{b_i-2}, h_{e_i}, h_{e_i+1}]$. The original entity type representation o is the (begin, end) indices of entity type from constructed prompt in the same way.

To learn better boundary-aware feature space, we extract entity type embedding e_o, entity token embedding e_{pos_i} and e_{neg_i}, from outputs $H \in \mathbb{R}^{B\times L\times D}$ of 25th hidden states layer in LlaMA-2, where B is the batch size and D is the hidden dimension. The calculation formula are:

$$e_{o_i} = \mathrm{gather}(H, o_i) \in \mathbb{R}^{B\times 1\times D}, \tag{3}$$

$$e_{\mathrm{pos}_i} = \mathrm{gather}(H, \mathrm{pos}_i) \in \mathbb{R}^{B\times 2\times D}, \tag{4}$$

$$e_{\mathrm{neg}_i} = \mathrm{gather}(H, \mathrm{neg}_i) \in \mathbb{R}^{B\times 4\times D}, \tag{5}$$

where $\mathrm{gather}()$ is a tensor operation commonly used in deep learning frameworks (e.g., PyTorch), which allows for the selection and extraction of specific elements from a higher-dimensional tensor H based on specified indices. Then, we can calculate the boundary-aware contrastive loss by:

$$\mathcal{L}_{\mathrm{cl}} = -\frac{1}{B}\sum_{i=1}^{B} \log\left(\sigma(\mathrm{sim}(o, \mathrm{pos}_i) - \mathrm{sim}(o, \mathrm{neg}_i))\right), \tag{6}$$

[2]https://github.com/tatsu-lab/stanford_alpaca

$$\mathrm{sim}(o, \mathrm{pos}_i) = \sum_{i=1}^{m} (\frac{e_o}{\|e_o\|_2} \cdot \frac{e_{\mathrm{pos}_i}}{\|e_{\mathrm{pos}_i}\|_2}) \in \mathbb{R}^B, \quad (7)$$

$$\mathrm{sim}(o, \mathrm{neg}_i) = \sum_{i=1}^{m} (\frac{e_o}{\|e_o\|_2} \cdot \frac{e_{\mathrm{neg}_i}}{\|e_{\mathrm{neg}_i}\|_2}) \in \mathbb{R}^B, \quad (8)$$

where $\sigma()$ is the sigmoid function.

3.1.3 LLM Fine-Tuning

We introduce instruction tuning to effectively and efficiently align the LLM with the span detection task. Following the standard supervised fine-tuning method, we minimize the auto-regressive loss calculated between the ground truth and the LLM output. In our approach, we mask the loss positions corresponding to the prompt part. Specific prompt formats, task-specific instructions, and ground truth details are provided in the Appendix A.1. However, directly fine-tuning the entire model can be computationally intensive and time-consuming. To address this, we propose a lightweight fine-tuning strategy using LoRA. This method involves freezing the pre-trained model parameters and introducing trainable rank decomposition matrices into each layer of the Transformer architecture. This approach facilitates lightweight fine-tuning while reducing GPU memory consumption. The final learning objective is computed as follows:

$$\mathcal{L}_{\mathrm{span}} = \min_{\theta_{L_1}} (\mathcal{L}_g + \lambda \mathcal{L}_{\mathrm{cl}}), \quad (9)$$

where θ_{L_1} is the LoRA parameters at the span detection stage and λ is set to 0.001.

3.2 Entity Type Classification

Subsequently, we assign a specific entity class to each span identified during the entity span detection stage.

3.2.1 Prompt and Prototype Representation

As previously mentioned, a predefined (candidate) list of entity types must be input as the schema into the LLM to trigger type generation. Figure 6 in Appendix A.1 illustrates an example of the prompt used for this stage. Using this prompt, the model constructs a prototype for each given entity type, which is then used to assign the correct type to each detected entity span.

To achieve this, we construct prototypical networks (ProtoNet) as the backbone, utilizing LoRA tuning across different domains. To leverage the knowledge from support examples in the target domain and align it with the source domain, we propose enhancing ProtoNet on the LLM with domain adaptation. This approach aims to create a more representative embedding space where text spans from different entity classes are more distinguishable.

Let $S_k = \{z_1, z_2, \ldots, z_n\}$ denote the set of entity type spans in the constructed prompt, which is contained in a given support set S_t belonging to the entity class $y_k \in Y$. We compute the prototype p_k for each y_k by averaging the span representations of all $z_i \in S_k$:

$$p_k(S_t) = \frac{1}{|S_k|} \sum_{i=1}^{|S_k|} z_i. \quad (10)$$

3.2.2 LoRA Tuning across Different Domains

Given a training episode $\mathcal{D}_t$, we first utilize the support set S_t to compute prototypes for all entity classes in Y_t using Eq. 10. Subsequently, for each span s_i in the query set Q_t, we calculate the probability that s_i belongs to an entity class y_k based on the distance between its span representation and the prototype of y_k:

$$\mathbb{P}(y_k; z_i) = \frac{exp\{-d(p_k(\mathcal{S}_t), s_i)\}}{\sum_{y_i \in Y} exp\{-d(p_i(\mathcal{S}_t), s_i)\}}, \quad (11)$$

To compute the distance function $d(\cdot, \cdot)$, we define it as follows:

$$d(p_{k/i}(\mathcal{S}_t), s_i) = \frac{p_{k/i}(\mathcal{S}_t)}{\|p_{k/i}(\mathcal{S}_t)\|_2} \cdot \frac{s_i}{\|s_i\|_2}. \quad (12)$$

Our goal is to minimize the cross-entropy loss for each LoRA module in its corresponding target domain:

$$\mathcal{L}_{t_i} = \min_{\theta_{L_2}} \left(- \sum_{z_i \in Q_t} \log \mathbb{P}_{\theta+\theta_{L_2}}(y_k; z_i) \right), \quad (13)$$

where θ_{L_2} is the LoRA parameters at the type classification stage.

3.2.3 Composition of LoRA Modules

As depicted in Figure 2(b), we initially fine-tuned LoRA modules across various target domains. Specifically, for M distinct domains, we fine-tune M separate LoRA modules, each denoted as m_i for the domain $\mathcal{D}_{t_i} \in \mathcal{D}_t$. Each m_i can be defined as the product $A_i B_i$, where $A_i \in \mathbb{R}^{d \times r}$ and $B_i \in \mathbb{R}^{r \times k}$ are trainable low-rank matrices, with the rank r being significantly smaller than the dimensions d and k. The combined LoRA module $\hat{m}$ can be obtained by:

$$\hat{m} = (w_1 A_1 \cdots + w_N A_N)(w_1 B_1 + \cdots + w_N B_N). \quad (14)$$

To find the optimal w, the optimization process is guided by the cross-entropy loss to identify the best set $w_1, w_2, \cdots, w_N$ that minimizes the loss $\mathcal{L}_{t_i}$ on the target domain. Additionally, we incorporate L1 regularization to penalize the sum of the absolute values of w, helping to prevent extreme values. Consequently, the final objective of LoRAHub is to minimize $\mathcal{L}_{t_i} + \alpha \cdot \sum_{i=1}^{N} |w_i|$, where α serves as a hyperparameter.

3.3 Target Domain Inference

As illustrated in Figure 2(c), during target domain inference, we first extract candidate spans from query sentences and then classify these spans into specific entity types to obtain the final results. After training the LLM with boundary-aware contrastive learning, we generate candidate entity spans from a given sentence X as follows:

$$P(S|X;\theta+\theta_{L_1}) = \prod_{i=1}^{N} P(y_i|y_{<t}, X; \theta+\theta_{L_1}). \quad (15)$$

Next, we obtain the candidate span set S_{span}, which includes all potential spans to be assigned entity types during the entity type classification stage. For these candidate spans, the entity types are classified as follows:

$$P(C|X,S;\theta+\theta_{L_2}) = \prod_{i=1}^{N} P(y_i|y_{<t}, X, S; \theta+\theta_{L_2}). \quad (16)$$

Finally, we combine the results of span detection and type classification to determine the predicted labels for a sentence X as follows:

$$P(Y|S,C;\hat{\theta}) = P(S|X;\theta+\theta_{L_1}) \cdot P(C|X,S;\theta+\theta_{L_2}). \quad (17)$$

Dataset	Domain	# Sentences	# Entities	# Classes
Few-NERD	Wikipedia	188k	491k	66
OntoNotes	General	76k	104k	18
I2B2	Medical	140k	29k	23
CoNLL	News	20k	35k	4
WNUT	Social	5k	3k	6
GUM	Wiki	3k	6k	11

Table 1: Statistics of Datasets

4 Experiments

4.1 Experimental Setups

4.1.1 Datasets

Few-NERD [3] **(Ding et al., 2021)** It is the largest few-shot NER dataset containing 66 fine-grained entity types across 8 coarse-grained categories. Two tasks are considered for this dataset: (1) **intra**, where all entities in the train/dev/test splits belong to different coarse-grained types, and (2) **inter**, where the train/dev/test splits may share coarse-grained types but have mutually exclusive fine-grained entity types.

[3] `https://ningding97.github.io/fewnerd/`

Cross-Dataset To evaluate cross domain adaption, we follow Das et al. (2022) and take OntoNotes 5.0 (General) (Weischedel et al., 2013) as our source domain, and evaluate few-shot domain adaptation performances on I2B2'14 (Medical) (Stubbs and Uzuner, 2015), CoNLL'03 (News) (Tjong Kim Sang and De Meulder, 2003), WNUT'17 (Social) (Derczynski et al., 2017), and GUM (Wiki) (Zeldes, 2017) datasets.

The statistics of datasets are shown in Table 1.

4.1.2 Baselines

We compare our proposed BANER with the *one-stage* and *two-stage* types. The *one-stage* baselines include **ProtoBERT** (Snell et al., 2017), **NNShot** (Wiseman and Stratos, 2019), **StructShot** (Yang and Katiyar, 2020), **CONTaiNER** (Das et al., 2022) and **MANNER** (Fang et al., 2023). The *two-stage* baselines include: **ESD** (Wang et al., 2022), **DecomposedMetaNER** (Ma et al., 2022b), **TadNER** (Li et al., 2023), **TSFNER** (Ji and Kong, 2024), and **BDCP** (Xue et al., 2024).

4.1.3 Evaluation Details

Evaluation on Few-NERD Following the methodology of Ma et al. (2022b), we adopt the episode-level evaluation approach. Each episode consists of a support set and a query set, structured in the N-way K-shot format. During evaluation, our model trained on the source domain predicts on the query set using information from the support set. To ensure fairness in comparisons, we compute the Micro F1 score based on episode data processed according to Ding et al. (2021). Results are reported as the mean F1 score ± standard deviation across 5 random seeds.

Evaluation on Cross-Dataset Yang and Katiyar (2020) points out the limitation that sampling test episodes may not accurately reflect real-world performance due to varying data distributions. They advocate for sampling support sets and subsequently evaluating models on the original test set. Each support set consists of K examples for each label. The final Micro F1 scores and standard deviations are calculated based on different sampled support sets. Following Yang and Katiyar (2020) and Das et al. (2022), we adopt this evaluation schema specifically for **cross-domain** settings. To ensure fair comparisons, we employ the support sets sampled according to the methodology proposed by Das et al. (2022)[4].

Parameters	Value	# Comment
temperature	0	control the randomness of generation
top_p	1	determine the cumulative probability for nucleus sampling
top_k	65536	limit the number of highest probability tokens considered
num_beams	4	set the number of beams for beam search
max_new_tokens	128	define the maximum number of tokens to generate

Table 2: Main parameters in inference.

4.1.4 Implementation Details

To construct BANER, we utilize LLaMA-2-7B as the pre-trained LLM backbone with FP16 precision and employ LoRA for prompt-tuning and model inference. During source domain training, we optimize using AdamW (Loshchilov and Hutter, 2019) with a learning rate of 3×10^{-4}, a batch size of 1, and training over five epochs with a micro batch size of one. The cutoff length is set to 256, and no validation set is used (i.e., val_set_size $= 0$). For LoRA, we set $r = 32$, $\alpha = 16$, and a dropout rate of 0.05. Distributed Data Parallel (DDP) is not employed for parameter search during training.

For target domain inference, Table 2 outlines the key parameters used in result generation. To ensure the robustness of generative language model outputs, our method incorporates task-specific instructions as inputs for entity span detection and type classification. Implementation is carried out using PyTorch 1.9.0[5] and executed on two Tesla A800-80G GPUs.

4.2 Main Results

Tables 3 and 4 present the comparative results between our method and baselines on the **Few-NERD** and **Cross-Dataset** benchmarks, respectively. Several key observations emerge:

1) Overall, two-stage methods consistently outperform one-stage methods, underscoring the efficacy of task decomposition in few-shot NER tasks.

2) BANER consistently outperforms all baselines in all settings, often exceeding the performance of the second-best models by a notable margin. In particular, in the challenging **intra** task, BANER achieves an average increase in the F1 score of 5. 2%.

[4]https://github.com/psunlpgroup/CONTaiNER
[5]https://pytorch.org/

3) Furthermore, in the 1-shot and 5-shot **Cross-Dataset** settings, BANER outperforms baselines by 2.3% and 5.1%, respectively. These results underscore the robustness of BANER in addressing cross-domain few-shot NER challenges.

4) TadNER, a competitive model, exhibits significantly degraded performance under certain settings, such as GUM. This issue primarily arises from dense entity sentences where boundary perception between different entities becomes challenging. In contrast, BANER effectively mitigates this challenge through the boundary-aware contrastive learning strategy, enabling accurate detection of entity spans and achieving superior performance.

4.3 Ablation Study

To validate the effectiveness of the main components in BANER, we introduce the following variant baselines for the ablation study:

BANER *w/o* Boundary-Aware Span Detection (BASD): This variant removes the boundary-aware contrastive learning at the span detection stage and directly extracts entity spans using LLMs.

BANER *w/o* Domain-Adaptation LoRAHub (DAL): This variant removes the composition of different LoRA modules at the type classification stage, using a single LoRA module to classify entities instead.

BANER *w/o* Span Detection Fine-Tuning (SDF): This variant skips the fine-tuning on the support set of the target domain at the span detection stage.

BANER *w/o* Type Classification Fine-Tuning (TCF): This variant skips the fine-tuning on the support set of the target domain at the type classification stage.

BANER *w/o ALL*: This variant performs the few-shot NER task using the original LLMs (e.g., LlaMA-2-7B) without any of the enhancements provided by BANER.

From Table 5, we observe the following:

1) The removal of the boundary-aware contrastive learning strategy results in a performance decline across most cases, particularly in entity-sparse datasets like I2B2, where many spans are falsely detected.

2) Omitting the domain-aware LoRAHub leads to a significant performance decrease. This indicates that our model effectively aligns a better prototype space for entity types, which is crucial in cross-domain scenarios.

Paradigms	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	
One-stage	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
	FSLS	30.38±2.85	28.31±4.03	46.85±3.49	40.76±3.18	36.58	44.52±4.59	44.01±3.35	59.74±2.51	56.67±1.75	51.24
	CONTaiNER	41.51±0.07	36.62±0.04	57.83±0.01	51.04±0.24	46.75	50.92±0.29	47.02±0.24	63.35±0.07	60.14±0.16	55.36
Two-stage	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
	TadNER	60.78±0.32	55.44±0.08	67.94±0.17	60.87±0.22	61.26	64.83±0.14	64.06±0.19	72.12±0.12	69.94±0.15	67.74
	TSFNER	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
	BDCP	53.96±0.92	52.17±0.56	59.25±0.28	56.91±1.12	55.57	**69.68±1.50**	67.15±0.28	71.12±0.97	68.13±0.55	69.02
	BANER	**64.95±0.85**	**61.24±0.82**	**72.14±0.33**	**67.53±0.12**	**66.47**	69.26±0.94	**67.43±0.35**	**76.53±0.51**	**72.24±0.22**	**71.37**

Table 3: F1 scores with standard deviations on Few-NERD. The best results are in **bold** and the second best ones are underlined.

Paradigms	Models	1-shot					5-shot				
		I2B2	**CoNLL**	**WNUT**	**GUM**	**Avg.**	**I2B2**	**CoNLL**	**WNUT**	**GUM**	**Avg.**
One-stage	ProtoBERT	13.4±3.0	49.9±8.6	17.4±4.9	17.8±3.5	24.6	17.9±1.8	61.3±9.1	22.8±4.5	19.5±3.4	30.4
	NNShot	15.3±1.6	61.2±10.4	22.7±7.4	10.5±2.9	27.4	22.0±1.5	74.1±2.3	27.3±5.4	15.9±1.8	34.8
	StructShot	21.4±3.8	62.4±10.5	24.2±8.0	7.8±2.1	29.0	30.3±2.1	74.8±2.4	30.4±6.5	13.3±1.3	37.2
	FSLS	18.3±3.5	50.9±6.5	14.3±5.5	12.6±2.8	24.0	25.4±2.7	63.9±3.3	24.0±3.2	18.8 ±2.2	33.1
	CONTaiNER	21.5±1.7	61.2±10.7	27.5±1.9	18.5±4.9	32.2	36.7±2.1	75.8±2.7	32.5±3.8	25.2±2.7	42.6
	MANNER	24.3±2.1	48.8±3.5	27.9±1.8	23.1±2.3	31.0	33.9±2.0	68.7±3.2	34.9± 2.5	40.7±1.2	44.6
Two-stage	DecomposedMetaNER	15.5±3.0	61.2±9.2	27.7±5.3	20.3±4.2	31.2	19.8±2.6	75.2±5.8	29.8±3.9	33.5±2.4	39.6
	TadNER	39.3±3.8	70.4±10.6	32.8±4.8	24.2±4.1	41.7	45.2±2.3	80.5±3.6	34.5±4.6	35.1±2.2	48.8
	TSFNER	35.0±0.9	62.5±4.1	28.3±2.5	**32.3±3.0**	39.5	40.6±2.5	72.4±5.6	34.7±2.4	38.9±0.9	46.7
	BDCP	33.2±3.1	63.9±8.3	30.3±2.0	31.1±1.5	39.6	37.7±2.2	69.8±8.9	34.0±1.6	34.6±1.5	44.0
	BANER	**40.2±1.0**	**72.6±3.1**	**34.1±2.1**	29.3±2.8	**44.0**	**47.1±2.2**	**81.2±2.9**	**43.2±1.2**	**44.0±0.9**	**53.9**

Table 4: F1 scores with standard deviations for Cross-Dataset.

Models	1-shot				5-shot				Avg.
	I2B2	CoNLL	WNUT	GUM	I2B2	CoNLL	WNUT	GUM	
BANER	**40.2**	**72.6**	**34.1**	**29.3**	**47.1**	**81.2**	**43.2**	**44.0**	**49.0**
w/o BASD	22.7	65.7	30.7	26.1	30.1	73.9	39.0	39.3	40.9
w/o DAL	30.3	64.0	32.5	27.0	34.6	73.8	39.5	40.2	42.2
w/o SDF	37.3	68.8	31.2	28.1	45.0	76.5	40.3	42.2	46.2
w/o TCF	39.2	69.0	32.1	28.0	45.7	78.2	40.9	42.4	46.9
w/o ALL	20.9	41.3	17.0	15.6	24.5	56.1	20.3	18.2	26.7

Table 5: Ablation study results for Cross-Dataset.

3) Eliminating fine-tuning in both the span detection and type classification stages causes a minor performance drop. This demonstrates that the prototype in the source domain aligns well with the target domain, and that LLMs already possess good boundary perception abilities despite encountering different entity types in the target domain after training in the source domain.

4) Although LLMs exhibit superior performance in few-shot tasks compared to most pretrained models, they still lag behind our approach. The significant disparity compared to the original LlaMA-2-7B underscores our model's effective utilization of provided support samples from the target domain, thereby enhancing the performance of LLMs in few-shot scenarios.

4.4 Examination of other LLMs

To evaluate the generalizability of our enhanced entity boundary perception, we extend BANER to other mainstream open-source LLMs under the GUM 5-shot setting, including Mistral-7B (Jiang et al., 2023) and LlaMA-3-8B. As shown in Figure

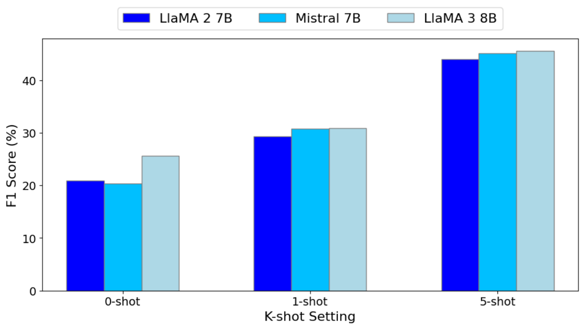

Figure 3: F1 Score for different LLMs under the GUM 5-shot setting.

3, substituting the LLM in BANER with these models leads to significant improvements in F1 scores for both 1-shot and 5-shot scenarios compared to the 0-shot baseline. This demonstrates the broad applicability and effectiveness of our method across different LLM architectures.

4.5 Impact of Different Hidden Layers

To determine which hidden layer's output in LlaMA-2 captures higher-level abstract information for constructing a better boundary-aware feature space, we compare overall performance by calculating the contrastive learning loss across different hidden layers under the GUM 5-shot setting. The performance of different hidden layers is shown in Figure 4. We observe that the highest F1 score is achieved when calculating the contrastive learning loss on the 25th layer. Notably, unlike other layers where there is a significant disparity between recall and precision, the 25th layer exhibits a relatively small difference between these metrics.

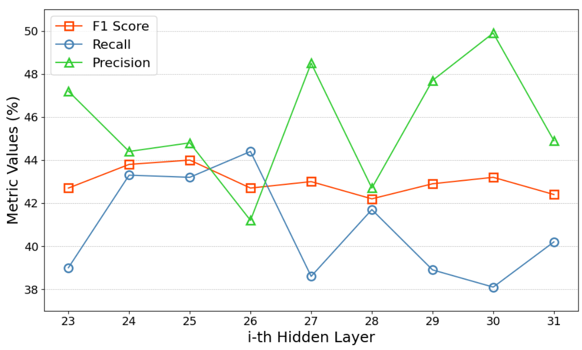

Figure 4: F1 Score, Recall, and Precision for different hidden layers under the GUM 5-shot setting.

5 Conclusion

In this paper, we propose the BANER framework for few-shot named entity recognition (NER), addressing entity span detection and entity type classification in two stages. For entity span detection, we introduce a boundary-aware contrastive learning strategy to minimize the distance between span embeddings of entities and their corresponding types using LLMs. Building on this, we employ domain adaptation with LoRAHub to construct more accurate prototypes that preserve and align entity type knowledge from the source domain during the entity classification stage. Extensive experiments demonstrate that BANER outperforms previous state-of-the-art methods and is applicable to various LLMs.

Limitations

Our work has two main limitations: 1) BANER employs a single, specific prompt template for each stage, utilizing descriptive task instructions and limited answer options. However, there exist numerous alternative templates for generative language models. This limitation suggests the potential for future research to explore various prompt templates to enhance entity boundary detection and entity type understanding. 2) Limited access to high-performance computing facilities has prevented us from evaluating our approach on large LLMs, such as LlaMA-3-70B. This limitation highlights the potential for future work to investigate different model architectures for improved few-shot NER performance.

References

Tommaso Mario Buonocore, Claudio Crema, Alberto Redolfi, Riccardo Bellazzi, and Enea Parimbelli. 2023. Localizing in-domain adaptation of transformer-based biomedical language models. *Journal of Biomedical Informatics*, 144:104431.

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.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. Association for Computational Linguistics.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. 2022. CONTaiNER: Few-shot named entity recognition via contrastive learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353, Dublin, Ireland. Association for Computational Linguistics.

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.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. 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)*, pages 3198–3213, Online. Association for Computational Linguistics.

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.

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*, SAC '19, page 993–1000, New York, NY, USA. Association for Computing Machinery.

David Grangier and Dan Iter. 2022. The trade-offs of domain adaptation for neural language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3802–3813.

Yanzhu Guo, Virgile Rennard, Christos Xypolopoulos, and Michalis Vazirgiannis. 2021. Bertweetfr: Domain adaptation of pre-trained language models for french tweets. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 445–450. Association for Computational Linguistics.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*.

Yucheng Huang, Kai He, Yige Wang, Xianli Zhang, Tieliang Gong, Rui Mao, and Chen Li. 2022. COPNER: Contrastive learning with prompt guiding for few-shot named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2515–2527, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Bin Ji, Shasha Li, Shaoduo Gan, Jie Yu, Jun Ma, Huijun Liu, and Jing Yang. 2022. Few-shot named entity recognition with entity-level prototypical network enhanced by dispersedly distributed prototypes. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1842–1854, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Shengjie Ji and Fang Kong. 2024. A novel three-stage framework for few-shot named entity recognition. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1293–1305, Torino, Italia. ELRA and ICCL.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

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, Akshen Kadakia, Kangmin Tan, Mahak Agarwal, Xinyu Feng, Takashi Shibuya, Ryosuke Mitani, Toshiyuki Sekiya, Jay Pujara, and Xiang Ren. 2022. Good examples make a faster learner: Simple demonstration-based learning for low-resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2687–2700, Dublin, Ireland. Association for Computational Linguistics.

Yongqi Li, Yu Yu, and Tieyun Qian. 2023. Type-aware decomposed framework for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8911–8927, Singapore. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuanjing Huang. 2022a. Template-free prompt tuning for few-shot NER. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5721–5732, Seattle, United States. Association for Computational Linguistics.

Tingting Ma, Huiqiang Jiang, Qianhui Wu, Tiejun Zhao, and Chin-Yew Lin. 2022b. Decomposed meta-learning for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1584–1596, Dublin, Ireland. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601.

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2020. Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4933–4941.

Minho Ryu, Geonseok Lee, and Kichun Lee. 2022. Knowledge distillation for bert unsupervised domain adaptation. *Knowledge and Information Systems*, 64(11):3113–3128.

Vin Sachidananda, Jason Kessler, and Yi-An Lai. 2021. Efficient domain adaptation of language models via adaptive tokenization. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 155–165.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021. Locate and label: A two-stage identifier for nested 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)*, pages 2782–2794, Online. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Amber Stubbs and Özlem Uzuner. 2015. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus. *Journal of biomedical informatics*, 58:S20–S29.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Peiyi Wang, Runxin Xu, Tianyu Liu, Qingyu Zhou, Yunbo Cao, Baobao Chang, and Zhifang Sui. 2022. An enhanced span-based decomposition method for few-shot sequence labeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5012–5024, Seattle, United States. Association for Computational Linguistics.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

Yaqing Wang, Haoda Chu, Chao Zhang, and Jing Gao. 2021. Learning from language description: Low-shot named entity recognition via decomposed framework. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1618–1630, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*.

Sam Wiseman and Karl Stratos. 2019. Label-agnostic sequence labeling by copying nearest neighbors. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5363–5369.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023. Empirical study of zero-shot ner with chatgpt. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7935–7956.

Xiaojun Xue, Chunxia Zhang, Tianxiang Xu, and Zhendong Niu. 2024. Robust few-shot named entity recognition with boundary discrimination and correlation purification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19341–19349.

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. *Lang. Resour. Eval.*, 51(3):581–612.

Xinghua Zhang, Bowen Yu, Yubin Wang, Tingwen Liu, Taoyu Su, and Hongbo Xu. 2022. Exploring modular task decomposition in cross-domain named entity recognition. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 301–311.

Yabin Zhang, Haojian Zhang, Bin Deng, Shuai Li, Kui Jia, and Lei Zhang. 2021. Semi-supervised models are strong unsupervised domain adaptation learners. *arXiv preprint arXiv:2106.00417*.

Yu Zhang, Kehai Chen, Xuefeng Bai, Zhao Kang, Quanjiang Guo, and Min Zhang. 2024a. Question-guided knowledge graph re-scoring and injection for knowledge graph question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8972–8985.

Yu Zhang and Zhao Kang. 2024. Tpn: Transferable proto-learning network towards few-shot document-level relation extraction. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE.

Zhen Zhang, Yuhua Zhao, Hang Gao, and Mengting Hu. 2024b. Linkner: Linking local named entity recognition models to large language models using uncertainty. In *Proceedings of the ACM on Web Conference 2024*, pages 4047–4058.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2024. UniversalNER: Targeted distillation from large language models for open named entity recognition. In *The Twelfth International Conference on Learning Representations*.

Xudong Zhu, Zhao Kang, and Bei Hui. 2024. Fcds: Fusing constituency and dependency syntax into document-level relation extraction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7141–7152.

A Appendix

A.1 Examples of Prompt

Figures 5 and 6 provide examples of the prompts used in the two stages of our method. To tailor these prompts to our task, we design a specific output format for the LLM. Each output starts with <im_start> and ends with <im_end>. For instances involving multiple entity spans and types, we encapsulate them together using <<< >>>.

A.2 Baselines

1) *one-stage methods*:

- **ProtoBERT** (Snell et al., 2017)is a popular few-shot method built on prototypical networks, utilizing BERT as its backbone;
- **NNShot** (Wiseman and Stratos, 2019) is a straightforward approach that utilizes token-level nearest neighbor classification;
- **StructShot** (Yang and Katiyar, 2020) adopts an additional Viterbi decoder on top of NNShot;
- **CONTaiNER** (Das et al., 2022) leverages contrastive learning to infer the distributional distance between Gaussian embeddings of entities;
- **MANNER** (Fang et al., 2023) uses a memory module and optimal transport to adapt source domain information for few-shot tasks in the target domain.

2) *two-stage methods*:

- **ESD** (Wang et al., 2022) enhances prototypical networks with inter- and cross-span attention , and introduces multiple prototypes for the O label;
- **DecomposedMetaNER** (Ma et al., 2022b) integrates model-agnostic meta-learning into prototypical networks to more effectively leverage the support set;
- **TadNER** (Li et al., 2023) employs type-aware contrastive learning and span filtering to construct precise prototypes and eliminate false spans;
- **TSFNER** (Ji and Kong, 2024) incorporates a teacher span recognizer for generating soft labels, a student span recognizer, and a prompt-based entity classifier;
- **BDCP** (Xue et al., 2024) introduces an entity boundary discriminative module for span detection and refines entity-context correlations to mitigate textual adversarial attacks.

A.3 Detailed Type Names

Following (Li et al., 2023), we substitute the original dataset labels with their corresponding natural-language forms of type names employed in our prompt. Tables 6 and 7 present the detailed conversions for various datasets.

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.

Instruction:
Please extract the entity of person in the input sentence given below , the entity of person refers to the entity that represents the identity or role of a specific person in the input sentence.

Input:
Pacific Standard owner , Jonathan M. Stan , displays the Santorum cocktail drink as a finished product at the bar . (2012) .

Response:
<im_start> I can extract entities for you, the extracted entities are <<< Pacific Standard owner >>> <<< Jonathan M. Stan >>> <im_end>

Figure 5: Example of the prompt in entity span detection.

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.

Instruction:
Please choose the correct type for the input entities from the type list. The type list is `["abstract","animal","event","object","organization","person","place","plant","quantity","substance","time"]`.

Input:
<<< Pacific Standard owner >>> <<< Jonathan M. Stan >>>

Response:
<im_start> I can choose the correct entity types for you, the entity types are <<< person >>> <<< person >>> <im_end>

Figure 6: Example of the prompt in entity entity classification.

Dataset	Labels	Type names
Few-NERD	art-broadcastprogram	broadcast program
	art-film	film
	art-music	music
	art-other	other art
	art-painting	painting
	art-writtenart	written art
	person-actor	actor
	person-artist/author	artist author
	person-athlete	athlete
	person-director	director
	person-other	other person
	person-politician	politician
	person-scholar	scholar
	person-soldier	soldier
	product-airplane	airplane
	product-car	car
	product-food	food
	product-game	game
	product-other	other product
	product-ship	ship
	product-software	software
	product-train	train
	product-weapon	weapon
	other-astronomything	astronomy thing
	other-award	award
	other-biologything	biology thing
	other-chemicalthing	chemical thing
	other-currency	currency
	other-disease	disease
	other-educationaldegree	educational degree
	other-god	god
	other-language	language
	other-law	law
	other-livingthing	living thing
	other-medical	medical
	building-airport	airport
	building-hospital	hospital
	building-hotel	hotel
	building-library	library
	building-other	other building
	building-restaurant	restaurant
	building-sportsfacility	sports facility
	building-theater	theater
	event-attack/battle /war/militaryconflict	attack battle war military conflict
	event-disaster	disaster
	event-election	election
	event-other	other event
	event-protest	protest
	event-sportsevent	sports event
	location-bodiesofwater	bodies of water
	location-GPE	geographical social political entity
	location-island	island
	location-mountain	mountain
	location-other	other location
	location-park	park
	location-road/railway /highway/transit	road railway highway transit
	organization-company	company
	organization-education	education
	organization-government /governmentagency	government agency
	organization-media/newspaper	media newspaper
	organization-other	other organization
	organization-politicalparty	political party
	organization-religion	religion
	organization-showorganization	show organization
	organization-sportsleague	sports league
	organization-sportsteam	sports team

Table 6: Original labels and their corresponding natural-language-form type names of Few-NERD.

Dataset	Labels	Type names
I2B2'14	AGE	age
	BIOID	biometric ID
	CITY	city
	COUNTRY	country
	DATE	date
	DEVICE	device
	DOCTOR	doctor
	EMAIL	email
	FAX	fax
	HEALTHPLAN	health plan number
	HOSPITAL	hospital
	IDNUM	ID number
	LOCATION_OTHER	location
	MEDICALRECORD	medical record
	ORGANIZATION	organization
	PATIENT	patient
	PHONE	phone number
	PROFESSION	profession
	STATE	state
	STREET	street
	URL	url
	USERNAME	username
	ZIP	zip code
CoNLL'03	PER	person
	LOC	location
	ORG	organization
	MISC	miscellaneous
GUM	abstract	abstract
	animal	animal
	event	event
	object	object
	organization	organization
	person	person
	place	place
	plant	plant
	quantity	quantity
	substance	substance
	time	time
WNUT'17	corporation	corporation
	creative-work	creative work
	group	group
	location	location
	person	person
	product	product
Ontonotes	CARDINAL	cardinal
	DATE	date
	EVENT	event
	FAC	fac
	GPE	geographical social political entity
	LANGUAGE	language
	LAW	law
	LOC	location
	MONEY	money
	NORP	nationality religion
	ORDINAL	ordinal
	ORG	organization
	PERCENT	percent
	PERSON	person
	PRODUCT	product
	QUANTITY	quantity
	TIME	time
	WORK_OF_ART	work of art

Table 7: Original labels and their corresponding natural-language-form type names of datasets under Cross-Dataset settings.