OBP-LLM: Optimizing Boundary Perception of Large Language Model for Few-shot NER

Anonymous ACL submission

Abstract

Few-shot Named Entity Recognition (NER) enables models to learn effectively from limited annotated samples and perform robustly, even in resource-rich domains, addressing the challenge of scarce labeled data in many fields. Recently, Large Language Models (LLMs) have demonstrated strong adaptability and generalization capabilities in few-shot learning, offering new solutions for few-shot NER tasks. In this paper, we propose OBP-LLM, a novel approach that integrates attention-based contrastive learning and Direct Preference Optimization (DPO) to enhance the performance of large language models in few-shot tasks by optimizing the model's perception of entity boundaries. Experimental results demonstrate that our method significantly outperforms existing approaches on multiple Few-shot NER benchmarks, including Few-NERD and CrossNER, particularly in cross-domain and extremely low-resource scenarios. This study validates the potential of contrastive learning and DPO in optimizing LLMs and provides new directions and practical solutions for NER tasks in low-resource domains.

1 Introduction

Named Entity Recognition (NER) is a critical task in natural language processing closely related to numerous other tasks. It aims to extract entities from unstructured text and classify them into predefined categories, such as person names, location names, and organization names (Guo et al., 2009; Mollá et al., 2006; Nadeau and Sekine, 2007). In recent years, deep learning models have achieved significant progress in NER tasks, particularly supervised methods based on pre-trained models like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), which achieve high accuracy by training on large-scale annotated datasets. However, these traditional methods heavily rely on extensive manually annotated datasets, which are often costly and time-consuming to obtain. Additionally, they exhibit limited flexibility in cross-domain applications. To address these issues, Few-shot Learning (FSL) (Ding et al., 2021a; Huang et al., 2021)has emerged as a research focus on NER tasks. The strength of FSL is its capability to identify new categories with few annotated samples, reducing the need for large labeled datasets while greatly enhancing cross-domain adaptability.

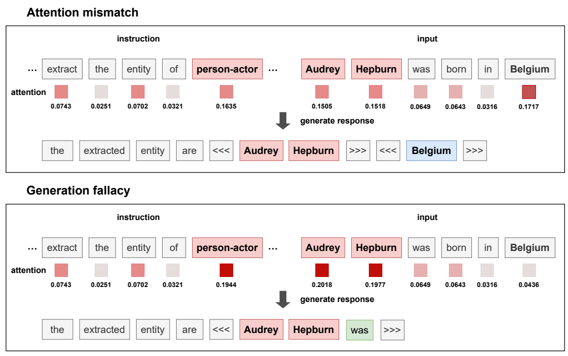


Figure 1: An illustration of two challenges applying the text generation framework of large language models to NER tasks. Here, we use Llama3.1-8b as the base model and compute the average of all multi-head attention scores at the 26th layer.

In the field of Few-shot NER, existing methods can be broadly categorized into two types:

(1) One-stage methods (Fritzler et al., 2019; Gao et al., 2019; Yang and Katiyar, 2020; Hou et al., 2020; Ma et al., 2022a): These methods transform NER tasks into sequence-labeling problems using prototype networks, classifying tokens by computing their distance to category prototypes. While computationally efficient, they are susceptible to interference from the non-entity label "O," degrading classification performance. Moreover, in transformer-based pre-trained models like BERT, self-attention mechanisms can cause cross-entity interference within the same sentence, leading to densely packed or overlapping entity distributions in the semantic space.

(2) Two-stage methods(Shen et al., 2021; Wang et al., 2022b; Ma et al., 2022b; Wang et al., 2022a; Dong et al., 2023): These approaches decompose NER tasks into two independent processes——span extraction and entity classification. The model first extracts all potential entity spans without assigning categories, followed by classification for each candidate span. While this decomposition improves entity boundary modeling, performance heavily depends on span extractor accuracy. Errors in span extraction inevitably impact entity classification.

With the rise of generative Large Language Models (LLMs), Few-shot NER tasks have seen breakthroughs. Compared to traditional pre-trained models, LLMs, such as Llama-3 (Dubey et al., 2024) and GPT-4 (Achiam et al., 2023), have larger parameter scales and stronger generalization capabilities. By designing various prompts, they can efficiently perform diverse NLP tasks without fine-tuning, demonstrating exceptional performance in few-shot learning scenarios (Zhang et al., 2024). While LLMs exhibit strong few-shot learning capabilities, we observe persistent challenges when using text-generation frameworks for NER tasks: 1) **Attention mismatch**: Input text suffers from attenuated attention allocation within the prompt, causing the model to focus on irrelevant tokens during response generation. 2) **Generation fallacy**: Although the model's attention is focused on the correct tokens, errors still occur during generation (e.g., incorrect entity boundaries).

To address these limitations, we propose a novel framework for LLMs based on contrastive learning and Reinforcement Learning, enhancing the model's perception of entity boundaries to ensure the generation of accurate entity responses. This framework achieves exceptional performance in extremely low-resource named entity recognition tasks by fine-tuning only a subset of LLM parameters via the LoRA method (Hu et al., 2021).

On the one hand, we impose constraints on the decoding phase during response generation, ensuring that generated tokens are derived solely from the input text. Additionally, we introduce attention-based contrastive learning during the Supervised Fine-Tuning (SFT) stage, bringing entities of the same category closer together while pushing different categories further apart in the semantic space, This optimization refines the distribution of entity representations, enabling a global semantic adjustment that enhances local attention mechanisms, thereby guiding the model to focus on the correct tokens.

On the other hand, to retain the rich boundary information utilized in two-stage methods without task decomposition (which risks cascading errors from subtasks), we construct preference data based on entity boundaries and error feedback from the initially aligned model. Through reinforcement learning, the model learns more precise boundary information and corrects previous errors to some extent. To simplify the reinforcement learning process, we adopt the computationally efficient Direct Preference Optimization (DPO) approach (Rafailov et al., 2023). Extensive experiments across multiple benchmarks demonstrate that our method consistently outperforms existing state-of-the-art approaches.

In summary, our main contributions are as follows:

(1) We propose a novel large language model-based approach to address few-shot NER tasks, which requires training only a subset of parameters yet demonstrates strong generalization capabilities on novel entity categories, especially in scenarios with extremely limited training samples.

(2)We introduce contrastive learning to optimize entity semantic representations, enhance attention during generation, and guide the model to focus on the correct tokens. Meanwhile, Direct Preference Optimization (DPO) enables the model to acquire richer entity boundary information while simultaneously refining itself through error feedback.

(3) Experiments conducted on two widely used few-shot NER benchmarks demonstrate that our method outperforms current state-of-the-art approaches, particularly in more challenging tasks.

2 Related Work

2.1 Few-shot Named Entity Recognition

Few-shot Named Entity Recognition (NER) aims to efficiently identify and classify entities with limited annotated data. The primary challenge is learning robust entity representations and achieving strong generalization under data scarcity.

One-stage methods directly model entity categories in the input text, typically employing metric learning strategies. Representative one-stage methods include Prototypical Networks (Snell et al., 2017) and other embedding space-based approaches, such as (Fritzler et al., 2019; Gao et al., 2019; Yang and Katiyar, 2020; Hou et al., 2020;

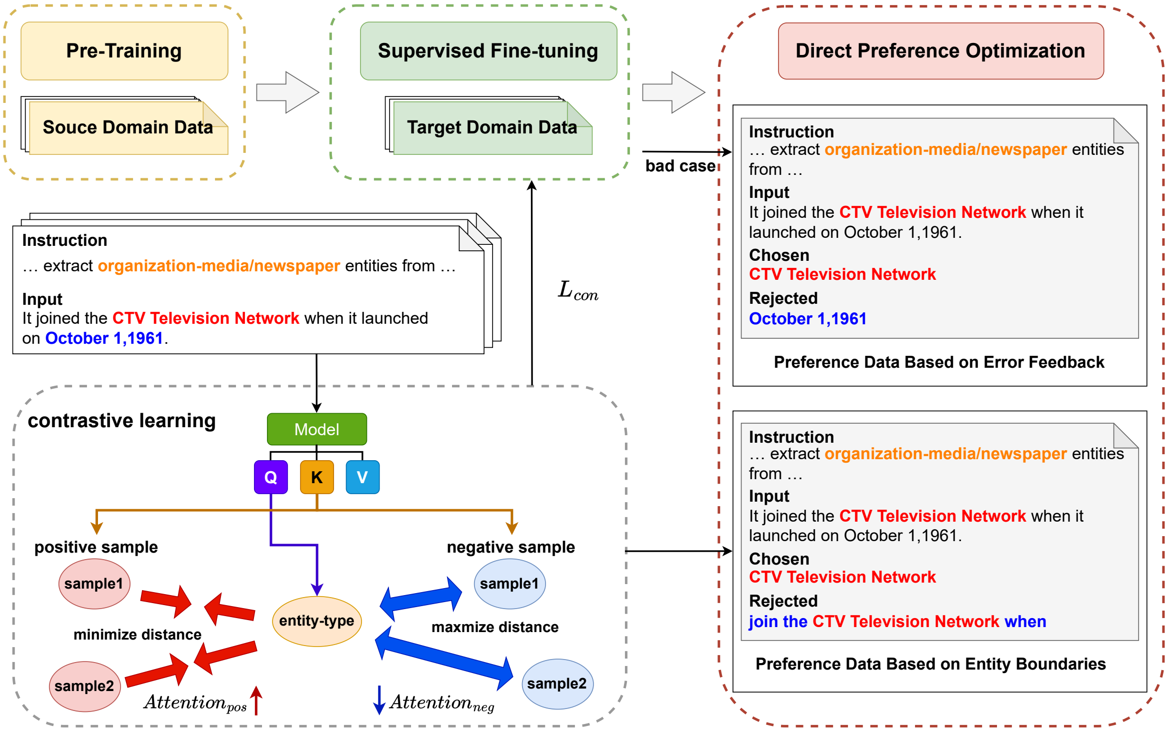


Figure 2: The overall architecture of OBP-LLM. It consists of three stages: pre-training on the source domain, supervised fine-tuning on the target domain, and Direct Preference Optimization.

Ma et al., 2022a).

Two-stage methods address few-shot NER by decomposing the task into two phases: span detection and entity classification (Shen et al., 2021; Wang et al., 2022b; Ma et al., 2022b; Wang et al., 2022a; Dong et al., 2023). Compared to one-stage methods, two-stage approaches place greater emphasis on boundary recognition capabilities. Although two-stage methods perform better in complex entity recognition scenarios, their staged design makes them susceptible to error propagation issues.

Furthermore, with the recent emergence of LLMs demonstrating remarkable capabilities in few-shot learning, several works have explored applying LLMs to few-shot NER tasks (Wang et al., 2023; Zhu et al., 2024).

2.2 Contrastive learning

The foundational concept of contrastive learning lies in the analysis of feature similarities and disparities. Hadsell et al. (2006) introduced contrastive loss, which refines feature representations by minimizing the distance between positive pairs while maximizing the separation between negative pairs. In recent years, contrastive learning has seen rapid advancements, particularly in computer vision and natural language processing (Chen et al., 2020; He et al., 2020). Today, contrastive learning has become a cornerstone technique in pre-training, finding widespread application (Reimers, 2019; Gao et al., 2021).

2.3 Direct Preference Optimization

Reinforcement Learning from Human Feedback (RLHF) optimizes model behavior by incorporating human preferences to align generated content with user expectations. Christiano et al. (2017) applied RLHF to simulated games and simple text generation tasks, while Ziegler et al. (2019) used human feedback to enhance the quality, coherence, and style of language model outputs, demonstrating its effectiveness in task optimization.

With the rise of pre-trained language models (e.g., the GPT series), RLHF has been widely adopted to improve text generation quality and control (Stiennon et al., 2020). The introduction of InstructGPT (Ouyang et al., 2022) and ChatGPT marks a key milestone in its application, driving its expansion in large-scale language models.

3 Method

3.1 Prompt Construction

Before starting the training process, we first construct a prompt for the LLM to adapt to the NER task (Zhang et al., 2024). Figure 3 below provides

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: I want you to extract **organization-media/newspaper** entities from the following input sentence, the entity of **organization-media/newspaper** refers to the entity that represents a specific media outlet, newspaper, or press organization in the input sentence.

input: It joined the **CTV Television Network** when it launched on October 1,1961.

response: The entities I extracted for you are <<< **CTV Television Network** >>> .

Figure 3: a example of prompt

an example of such a prompt, which consists of four parts: (1) The first line is a fixed description of the alpaca-lora method, introducing the following three sections: Instruction, Input, and Response. (2) instruction: In this part, we specify the entity categories to be extracted and briefly describe the definition of each entity type to help the LLM better understand the NER task. (3) input: The sentence from which entities are to be extracted. (4) response: This section contains the model's generated output, where each extracted entity is enclosed within <<< >>> identifiers.

3.2 Pre-Training in Source Domain

We first construct training data from the source Domain using the prompt designed in the previous subsection, enabling the model to perform NER tasks on the source Domain.

$$\mathcal{L} = -\sum_{t=1}^{T} \log p(y_t | x_{1:t-1}) \tag{1}$$

where T is the length of the generated sequence, y is the target output, $x_{1:t-1}$ is the input sequence before the current time step t, and $p(y_t|x_{1:t-1})$ is the probability of the model predicting y_t given $x_{1:t-1}$ as input.

However, we aim for the model to focus more on generating better and more accurate responses rather than overly emphasizing the instruction and input. Therefore, the loss during training is computed solely based on the tokens in the model's response.

$$\mathcal{L}_{source} = -\sum_{t=r}^{T} \log p(y_t | x_{r:t-1}) \tag{2}$$

where model's response $x_{res} = \{x_r, x_{r+1}, ...x_T\}$.

3.3 Supervised Fine-tuning with contrastive learning

After pre-training on the Source Domain, we perform supervised fine-tuning (SFT) on the model using a small number of Target Domain samples. Similar to the Source Domain, we fine-tune the model with next-token prediction. However, unlike the Source Domain, we introduce attention-based contrastive learning during SFT. By constructing positive and negative sample pairs, we optimize entity representations, improve internal attention, and enhance the model's perception of entity boundaries. The process of constructing positive and negative sample pairs is as follows:

For a given input x_i and the entity category p_i to be extracted, $q_{i,j}$ represents the entity in x_i. $q_{i,j} \in C_i^{pos}, |C_i^{pos}| = J$. Then, $(p_i, q_{i,j})$ forms a positive sample pair, with a total of J pairs. For each positive sample pair $(p_i, q_{i,j})$, we select K tokens $n_{i,j,k}$ near the boundary of entity $q_{i,j}$, where $n_{i,j,k} \in C_{i,j}^{neg}, \left|C_{i,j}^{neg}\right| = K$, and $(p_i, n_{i,j,k})$ forms a negative sample pair, with a total of K pairs. In this way, the model can implicitly learn some information related to entity boundaries. We apply contrastive learning to the model's attention to improve internal attention, increasing focus on positive samples and reducing focus on negative samples. The contrastive loss function is defined as follows:

$$\begin{aligned}\mathcal{L}_{con} = -\frac{1}{N}\sum_{i=1}^{N} \log(\sigma(\sum_{j}^{J}(\mathbf{e}_i^{Q,type} \cdot \mathbf{e}_{i,j}^{K,pos}) \\ - \sum_{k}^{K}(\mathbf{e}_i^{Q,type} \cdot \mathbf{e}_{i,j,k}^{K,neg})))\end{aligned} \tag{3}$$

We used cosine similarity to represent the distance between positive and negative sample pairs. Specifically, for a given input x_i, $\mathbf{e}_i^{Q,type}$ represents the embedding of the entity category p_i output by the Q projector in the model. $\mathbf{e}_{i,j}^{K,pos}$ and $\mathbf{e}_{i,j,k}^{K,neg}$ represent the embeddings of the positive sample $q_{i,j}$ and the negative sample $n_{i,j,k}$ output by the K projector, respectively. It is important to note that the output embeddings from both the Q projector head and the K projector head are averaged across all heads and normalized at the 26th layer. Here, we choose the output of the Q and K projector heads instead of the hidden layer states to further improve internal attention and achieve faster convergence and better performance during training.

By combining the SFT loss and the contrastive learning loss, we obtain the overall loss function for fine-tuning the target domain. This allows the model to adapt to the target domain while optimizing the representations' distribution in the semantic space. Here, λ is used to control the weight of the contrastive learning loss, and in our experiments, $\lambda = 0.01$.

$$\mathcal{L}_{target} = \mathcal{L}_{sft} + \lambda \mathcal{L}_{con} \quad (4)$$

$$\mathcal{L}_{sft} = \mathcal{L}_{source} \quad (5)$$

3.4 Direct Preference Optimization on Entity Boundary

After the initial alignment of the model with the target domain via SFT, we use preference data based on entity boundaries and error feedback to adjust the model's generation preferences using RLHF. This enables the model to learn more accurate entity boundaries and correct existing errors. The process of constructing preference data is as follows:

(1)Preference Data Based on Entity Boundaries For the data shown in Figure 3, we generate incorrect entity responses by shifting one token left or right from the correct entity boundaries. These incorrect entity responses are labeled as low-preference '**rejected**' samples, while the original correct responses are labeled as high-preference '**chosen**' samples.

chosen: The entities I extracted for you are <<< CTV Television Network >>>.

rejected: The entities I extracted for you are <<< join the CTV Television Network when >>>.

(2)Preference Data Based on Error Feedback We use the training data from the previous phase to test the SFT model that has undergone the first alignment. The misclassified entity extraction results are then used to construct the preference data. Specifically, the original correct answers are labeled as the '**chosen**' data in the preference dataset, while the incorrect responses generated by the SFT model are labeled as the '**rejected**' data.

After constructing the preference data, in traditional RLHF methods, we first need to train a reward model to evaluate and score the generated responses on the preference data $D = \left\{x^{(i)}, y_w^{(i)}, y_l^{(i)}\right\}_{i=1}^{N}$, where $y_w^{(i)}$ and $y_l^{(i)}$ represent the preferred and non-preferred generations given input $x^{(i)}$, respectively. According to the Bradley-Terry (BT) model, the negative log-likelihood loss for the reward model is defined as:

$$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} [\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))] \quad (6)$$

Where σ is the logistic function, during initialization, $r_\phi(x, y)$ is typically implemented by adding a linear layer on top of the SFT model $\pi^{sft}(y|x)$ from the previous stage to score the model's generations. After obtaining the trained reward model, the large language model is further optimized based on feedback from the reward model. This process is formulated as:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)} [r_\phi(x, y)] - \beta\mathbb{D}_{\mathbf{KL}} [\pi_\theta(y \mid x) ||| \pi_{\text{ref}}(y \mid x)] \quad (7)$$

where β is a parameter controlling the deviation from the baseline reference policy model π_{ref}, $\pi_\theta(y|x)$ is the current language model, and both pi_{ref} and $\pi_\theta(y|x)$ are initialized with the SFT model $\pi^{sft}(y|x)$. This ensures that the model is optimized toward higher rewards, as scored by the reward model while preventing the generation distribution from deviating too far from the SFT model, which could otherwise lead to unpredictable and undesirable outputs.

To simplify the training process and avoid the need for training a reward model, we use the Direct Preference Optimization (DPO) method to perform the model policy optimization. Based on the derivation from the Equation 7, we obtain the following:

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi} [r(x, y)] - \beta\mathbb{D}_{\text{KL}} [\pi(y|x) \,\|\, \pi_{\text{ref}}(y|x)]$$

$$= \min_{\pi} \mathbb{E}_{x\sim\mathcal{D}} \mathbb{E}_{y\sim\pi(y|x)} \left[\log \frac{\pi(y|x)}{\frac{1}{Z(x)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right)} - \log Z(x)\right] \quad (8)$$

where $Z(x)$ is the partition function. We will not elaborate on the derivation method here. For a detailed derivation, please refer to the paper on DPO (Rafailov et al., 2023).

$$Z(x) = \sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right) \quad (9)$$

The explicit optimal solution $\pi^*(y|x)$ for model $\pi(y|x)$ is:

$$\pi^*(y|x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right) \quad (10)$$

The form of the reward model $r(x, y)$ can be derived as follows:

$$r^*(x, y) = \beta \log \frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x) \quad (11)$$

By substituting the reward model $r(x,y)$ into the loss function under the Bradley-Terry (BT) model Equation 6 for optimization, the optimal solution is directly obtained through the process of training the reward model.

$$\mathcal{L}_R(r_\phi,\mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\log\sigma\Big(\beta\log\frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} - \beta\log\frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)}\Big)\Big] \quad (12)$$

The preference generation $y_w^{(i)}$ corresponds to the '**chosen**' part of the preference data we construct, while the non-preferred generation $y_l^{(i)}$ corresponds to the '**rejected**' part.

Finally, we incorporate a portion of the model's SFT loss into the training process to prevent the model from deviating too much from the initial alignment results. α represents the weight of the SFT loss.

$$\mathcal{L}_{dpo}' = \alpha\mathcal{L}_{sft}(\pi^*(y|x)) + \mathcal{L}_{dpo} \quad (13)$$

4 Experiments

4.1 Datasets

We selected two widely used few-shot named entity recognition benchmarks for evaluation: Few-NERD and CrossNER.

Few-NERD: Few-NERD (Ding et al., 2021b) is a large-scale, fine-grained manually annotated NER dataset with 8 coarse-grained and 66 fine-grained entity categories. It provides two few-shot settings: Inter and Intra. In the Inter setting, the training, validation, and test sets share all coarse-grained categories but have disjoint fine-grained entity categories. In the Intra setting, entity categories are disjoint at both coarse-grained and fine-grained levels. Here, we use the episode data released by Ding et al. for experiments, defining the few-shot tasks as N-way K~2K-shot scenarios, where N-way indicates the number of entity categories in the task, and K~2K-shot denotes the sampling of K~2K training instances per entity category.

CrossNER: CrossNER (Hou et al., 2020) consists of four datasets: CoNLL-2003 (Sang and De Meulder, 2003), GUM (Zeldes, 2017), WNUT-17 (Derczynski et al., 2017), and OntoNotes (Pradhan et al., 2013), coming from four distinct domains: News, Wiki, Social, and Mixed. We used the episode data constructed by Hou et al. (2020), selecting two domains for training, one for validation, and one for testing.

Datasets	Domain	#Sent	#Labels
Few-NERD	Mixed	188.2k	66
CoNLL-03	News	20.7k	5
GUM	WiKi	3.5k	12
WNUT-17	Social	5.6k	7
OntoNotes	Mixed	159.6k	19

Table 1: The statistics of each dataset.

4.2 Baselines

For the baselines, we refer to previous works and select several strong methods from both one-stage and two-stage paradigms.

One-stage paradigms include ProtoBERT (Fritzler et al., 2019), Matching Network (Vinyals et al., 2016), StructShot (Yang and Katiyar, 2020), NNShot (Yang and Katiyar, 2020), CONTAINER (Snigdha et al., 2022), and LTapNet+CDT (Hou et al., 2020).

Two-stage paradigms include ESD (Wang et al., 2022b), DecomMeta (Ma et al., 2022b), SpanProto (Wang et al., 2022a), and MSDP (Dong et al., 2023).

4.3 Implementation Details

We chose Meta's Llama3.1-8b (Dubey et al., 2024), available on HuggingFace, as the initial language model. For subsequent training, we employed the LoRA method, fine-tuning only a subset of the large language model's parameters to reduce hardware requirements. The LoRA rank was set to 8, and the LoRA alpha was set to 16. During the SFT phase, the parameter λ, controlling the contrastive learning loss, was set to 0.01, while in the DPO training phase, the parameter β was set to 0.1, and the weight α for the SFT loss was set to 0.2.

We used Adam as the optimizer and applied different learning rates across training stages: a learning rate of 3e-4 for the source domain pre-training and target domain SFT phases, and 5e-6 for the DPO phase. The warm-up ratio was set to 0.1.

All experiments were conducted using a single 4090 GPU for both training and testing.

4.4 Main Result

Tables 2 and 3 present the main results comparing our method with other baselines. We have the following observations: 1) Our proposed OBP-LLM significantly outperforms previous methods by a large margin on both the Few-NERD and CrossNER benchmarks. Compared to MSDP, it achieves overall average improvements of 2.26% and 17.16%

Paradigms	Models	Intra 1∼2-shot 5 way	Intra 1∼2-shot 10 way	Intra 5∼10-shot 5way	Intra 5∼10-shot 10 way	Intra Avg.	Inter 1∼2-shot 5 way	Inter 1∼2-shot 10 way	Inter 5∼10-shot 5way	Inter 5∼10-shot 10 way	Inter Avg.
One-stage	ProtoBERT	23.45±0.92	19.76±0.59	41.93±0.55	34.61±0.59	29.94	44.44±0.11	39.09±0.87	58.80±1.42	53.97±0.38	49.08
	NNShot	31.01±1.21	21.88±0.23	35.74±2.36	27.67±1.06	29.08	54.29±0.40	46.98±1.96	50.56±3.33	50.00±0.36	50.46
	StructShot	35.92±0.69	25.38±0.84	38.83±1.72	26.39±2.59	31.63	57.33±0.53	49.46±0.53	57.16±2.09	49.49±1.77	53.34
	CONTaiNER	40.43	33.84	53.70	47.49	43.87	55.95	48.35	61.83	57.12	55.81
	OBP-LLM	**75.54±2.14**	**74.31±2.38**	**78.01±3.11**	**75.74±2.22**	**75.90**	**81.14±4.41**	**79.23±1.63**	**81.30±3.79**	**80.29±2.92**	**80.49**
Two-stage	ESD	41.44±1.16	32.29±1.10	50.68±0.94	42.92±0.75	41.83	66.46±0.49	59.95±0.69	74.14±0.80	67.91±1.41	67.12
	DecomMeta	52.04±0.44	43.50±0.59	63.23±0.45	56.84±0.14	53.9	68.77±0.24	63.26±0.40	71.62±0.16	68.32±0.10	67.99
	SpanProto	54.49±0.39	45.39±0.72	65.89±0.82	59.37±0.47	56.29	73.36±0.18	66.26±0.33	75.19±0.77	70.39±0.63	71.3
	MSDP	56.35±0.28	47.13±0.69	66.80±0.78	64.69±0.51	58.74	76.86±0.22	69.78±0.31	84.78±0.69	81.50±0.71	78.23

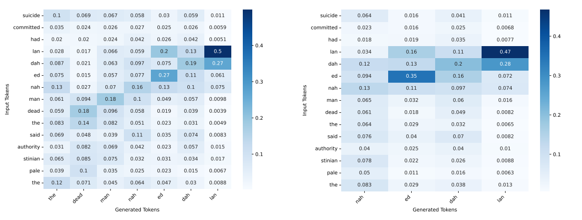
Table 2: F1 scores on Few-NERD for both inter and intra settings.

Paradigms	Models	1-shot CONLL-03	1-shot GUM	1-shot WNUT-17	1-shot OntoNotes	1-shot Avg.	5-shot CONLL-03	5-shot GUM	5-shot WNUT-17	5-shot OntoNotes	5-shot Avg.
One-stage	Matching Network	19.50±0.35	4.73±0.16	17.23±2.75	15.06±1.61	14.13	19.85±0.74	5.58±0.23	6.61±1.75	8.08±0.47	10.03
	ProtoBERT	32.49±2.01	3.89±0.24	10.68±1.40	6.67±0.46	13.43	50.06±1.57	9.54±0.44	17.26±2.65	13.59±1.61	22.61
	L-TapNet+CDT	44.30±3.15	12.04±0.65	20.80±1.06	15.17±1.25	23.08	45.35±2.67	11.65±2.34	23.30±2.80	20.95±2.81	25.31
	OBP-LLM	**59.55±3.32**	**44.63±4.78**	**65.43±3.86**	**55.31±3.35**	**56.23**	**65.67±3.08**	**51.82±4.41**	**64.66±2.31**	**59.81±2.25**	**60.49**
Two-stage	DecomMeta	46.09±0.44	17.54±0.98	25.14±0.24	34.13±0.92	30.73	58.18±0.87	31.36±0.91	31.02±1.28	45.55±0.90	41.53
	SpanProto	47.70±0.49	19.92±0.53	28.31±0.61	36.41±0.73	33.09	61.88±0.83	35.12±0.88	33.94±0.50	48.21±0.89	44.79
	MSDP	49.14±0.52	21.88±0.29	30.10±0.56	38.05±0.88	34.79	63.98±0.80	36.53±0.81	35.61±0.72	49.99±0.95	46.53

Table 3: F1 scores under 1-shot and 5-shot setting on CrossNER.

Methods	Few-NERD Intra	Few-NERD Inter	CrossNER 1-shot	CrossNER 5-shot
OBP-LLM	75.90	80.49	56.23	60.49
w/o contrastive learning	74.56	79.07	53.83	59.23
w/o dpo	75.04	79.56	54.82	59.42

Table 4: The ablation study results (average F1 score) for Few-NERD and CrossNER.



(a) base model (b) with CL

Figure 4: The comparison of attention heatmaps, where (a) represents the Llama3.1-8b model with only SFT training, and (b) represents the model with contrastive learning added during the SFT phase.

on Few-NERD Inter and Few-NERD Intra, respectively, and a 21.75% improvement on CrossNER, demonstrating the effectiveness of our approach. 2) Among previous methods, two-stage paradigms consistently outperformed one-stage paradigms. However, our method, which preserves the integrity and coherence of the NER task within a one-stage paradigm, is the first to surpass two-stage methods in all aspects. 3) The Intra scenario in Few-NERD is more challenging as entity categories in the training, validation, and test sets are disjoint not only at the fine-grained level but also at the coarse level. Similarly, CrossNER is difficult due to both different entity categories and datasets from diverse domains. Previous methods have significant room for improvement in these tasks. Our OBP-LLM shows remarkable improvements in both Few-NERD and CrossNER, demonstrating its strong generalization ability in few-shot learning, especially in cross-domain scenarios.

4.5 Ablation Study

We conducted ablation studies on the main components of OBP-LLM, focusing on 1) contrastive learning during the SFT phase and 2) Direct Preference Optimization (DPO) based on entity boundary information. The results are shown in Table 4

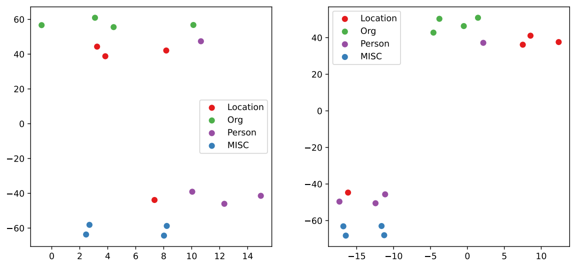
1) When either of these components is removed, the overall average performance of the model declines, indicating that both components are necessary and highly effective.

2) When contrastive learning is removed, the average F1 score drops by 1.26% to 2.4%, with a more pronounced decline in the cross-dataset task CrossNER. This demonstrates that contrastive learning effectively optimizes the model for cross-domain tasks.

3) When DPO is removed, the average F1 score decreases by approximately 1% overall. Compared to contrastive learning, the drop in F1 score is smaller, as DPO primarily refines the model's judgment of entity boundaries while maintaining the initial alignment results of the large language model.

4.6 Effectiveness of Contrastive learning

In the SFT phase, we introduce contrastive learning to optimize the distribution of entity representations in the model's semantic space and enhance



(a) base model (b) with CL

Figure 5: t-SNE visualization of entity representations on CrossNER for the base model and the base model with contrastive learning, with each color representing a different entity category.

its attention mechanism, improving performance in entity recognition tasks.

To evaluate this, we perform SFT on the base Llama3.1-8b model, using it as the baseline. We randomly select some data to test the impact of contrastive learning on attention. Figure 4 shows a representative comparison of attention heatmaps, with “nahed dahlan” as the standard output. The base model focuses excessively on irrelevant tokens, leading to redundant output, while contrastive learning helps the model focus on the correct entity tokens.

Additionally, we visualized the distribution of entity representations in the semantic space using t-SNE, as shown in Figure 5. Compared to the base model, the model trained with contrastive learning shows a significantly more compact distribution of entities within the same category. However, the improvement in the boundary distinction between different categories of entities was relatively less pronounced. This is because, when constructing negative samples for contrastive learning, to avoid extreme imbalance in the number of positive and negative samples, we selected tokens near the entities rather than all tokens outside the correct entities, many of which are non-entity tokens.

Overall, the results demonstrate that the contrastive learning we introduced effectively improves entity semantic representations and enhances model performance in entity recognition tasks.

4.7 Effectiveness of Direct Preference Optimization

To validate the Direct Preference Optimization (DPO) based on entity boundaries and error feedback, which strengthens the model’s learning of entity boundary information and performs effective error correction after the model’s first alignment, we randomly selected 500 samples from the CrossNER task for testing. We also selected representative instances, as shown in Figure 6. In this case, the baseline model has undergone only the first alignment in the SFT phase. It can be observed that the output of the baseline model may contain word redundancies in entity boundaries, even though these redundant words sometimes do not affect the overall meaning of the entity. After the DPO phase, the model can identify more accurate entity boundaries and also correct some previously erroneous responses.

Input	the hood opening reminds me of a classic ***saab 900***$_{product}$.
Output	saab 900
Baseline	**a classic saab 900 ×**
Con-DpoNER	**saab 900** √
Input	are the legality of votes cast by ***non citizens***$_{group}$ checked after they have been cast.
Output	non citizens
Baseline	**the legality of votes ×**
Con-DpoNER	**non citizens** √

Figure 6: A case of CrossNER.The correct and incorrect entities are highlighted in red and green, respectively.

5 Conclusion

We propose OBP-LLM, a method for optimizing entity boundary perception in large language models. By introducing attention-based contrastive learning during the SFT phase, we enhance the distribution of entity representations and improve attention, enabling the model to focus on the correct entity tokens. Additionally, we apply RLHF for secondary alignment optimization based on entity boundary information and error feedback, simplifying the training process with DPO. Extensive experiments demonstrate that our approach, requiring only partial parameter training, outperforms previous SOTA baselines, especially in more challenging tasks.

Limitations

As mentioned in Section 4.6, due to the construction method, our model shows limited improvement in distinguishing boundaries between different entity categories in contrastive learning. We believe there is significant room for optimization in the negative sample construction method, which will be a focus of our future research.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

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.

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

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021a. 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.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021b. 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.

Guanting Dong, Zechen Wang, Jinxu Zhao, Gang Zhao, Daichi Guo, Dayuan Fu, Tingfeng Hui, Chen Zeng, Keqing He, Xuefeng Li, et al. 2023. A multi-task semantic decomposition framework with task-specific pre-training for few-shot ner. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 430–440.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in named entity recognition task. In *Proceedings of the 34th ACM/SIGAPP symposium on applied computing*, pages 993–1000.

T Gao, X Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP 2021-2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*.

Tianyu Gao, Xu Han, Zhiyuan Liu, and Maosong Sun. 2019. Hybrid attention-based prototypical networks for noisy few-shot relation classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6407–6414.

Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named entity recognition in query. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 267–274, New York, NY, USA. Association for Computing Machinery.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006*, pages 1735–1742.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2021. Few-shot named entity recognition: An empirical baseline study. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 10408–10423. Association for Computational Linguistics (ACL).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jie Ma, Miguel Ballesteros, Srikanth Doss, Rishita Anubhai, Sunil Mallya, Yaser Al-Onaizan, and Dan Roth. 2022a. Label semantics for few shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1956–1971.

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.

Diego Mollá, Menno Van Zaanen, and Daniel Smith. 2006. Named entity recognition for question answering. In *Proceedings of the Australasian Language Technology Workshop 2006*, pages 51–58.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: your language model is secretly a reward model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 53728–53741.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Erik 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.

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.

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

Sarkar Snigdha, Sarathi Das, Arzoo Katiyar, Rebecca J Passonneau, and Rui Zhang. 2022. Container: Few-shot named entity recognition via contrastive learning. In *60th Annual Meeting of the Association for Computational Linguistics, ACL 2022*, pages 6338–6353. Association for Computational Linguistics (ACL).

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

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

Jianing Wang, Chengyu Wang, Chuanqi Tan, Minghui Qiu, Songfang Huang, Jun Huang, and Ming Gao. 2022a. Spanproto: A two-stage span-based prototypical network for few-shot named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3466–3476.

Peiyi Wang, Runxin Xu, Tianyu Liu, Qingyu Zhou, Yunbo Cao, Baobao Chang, and Zhifang Sui. 2022b. 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.

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*.

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.

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

Yafeng Zhang, Zilan Yu, Yuang Huang, and Jing Tang. 2024. Cllmfs: A contrastive learning enhanced large language model framework for few-shot named entity recognition. In *ECAI 2024*, pages 1985–1992. IOS Press.

Xingyu Zhu, Feifei Dai, Xiaoyan Gu, Bo Li, Meiou Zhang, and Weiping Wang. 2024. Gl-ner: Generation-aware large language models for few-shot named entity recognition. In *International Conference on Artificial Neural Networks*, pages 433–448. Springer.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.