Advancements in Few-Shot Nested Named Entity Recognition: The Efficacy of Meta-Learning Convolutional Approaches

Anonymous ACL submission

Abstract

Few-shot Named Entity Recognition (NER) involves the identification of new entities using a limited amount of labeled data, which may contain nested entities. Currently, mainstream few-shot NER methods are not designed to handle nested entities. This study introduces a novel span-based meta-learning framework that uses meta-learning convolution to address the challenges of few-shot nested NER. Our proposed method, called **M**eta-Learning **C**onvolution for **F**ew-**S**hot **N**ested NER (**MCFSN**), is the first to integrate meta-learning with convolutional neural networks, effectively handling nested entities with limited training examples. This study presents a two-stage processing approach: extracting span features using CNN combined with the Biaffine attention mechanism, followed by entity span classification utilizing ProtoNet and the Biaffine classifier. Our experiments demonstrate consistently superior performance across three diverse language datasets, outperforming several competing baseline models in terms of F1 scores. Specifically, our approach achieves 6.9% F1 score improvement on the Genia, 5.2% F1 value improvement on the GermEval, and 4.5% F1 value enhancement on the NEREL, thus validating the effectiveness of our proposed approach.

1 Introduction

Named Entity Recognition (NER), a core task in Natural Language Processing (NLP) (Zhang et al., 2022; Yang et al., 2017; Yan et al., 2021), is essential for identifying and classifying predefined entity categories within text. This task is particularly crucial for various downstream NLP applications such as information extraction (Lample et al., 2016a; Ma and Hovy, 2016; Peters et al., 2017; Cui and Zhang, 2019; Yamada et al., 2020). The challenge of NER is especially pronounced in specific domains such as bioinformatics and in non-English languages such as German and Russian. These fields often have limited annotated data available (Cui et al., 2021; Ma et al., 2022b; Lee et al., 2022a), leading researchers to focus on few-shot NER (Wang et al., 2022c; Ma et al., 2022a), as exemplified in Figure 1(a). A significant yet often overlooked issue in existing few-shot NER research is nested NER, where one entity may contain another, as shown in Figure 1(b). This phenomenon is more common in certain domains because of the textual characteristics of the field (Sonkar et al., 2022; Wang et al., 2022a). For instance, in bioinformatics texts, entities suck as proteins, genes, or disease names are frequently nested and interconnected, forming complex entity structures.

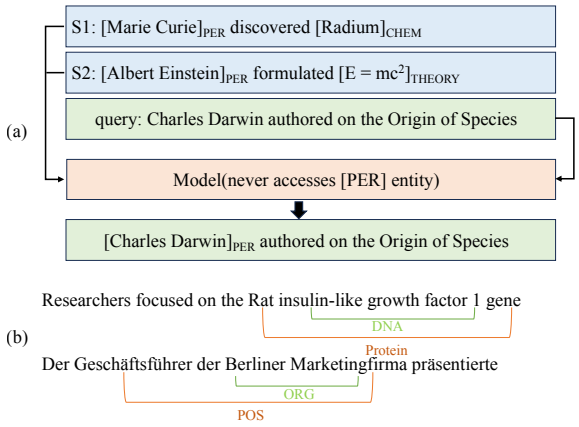

Figure 1: (a) Illustration of a 2-way, 2-shot few-shot NER task, where new entities are learned from two examples. (b) Example sentences demonstrating GENIA Nested NER and German Compound Noun Nested NER.

Most existing work either concentrates on few-shot NER while overlooking the nested structure of entities or focuses on nested NER but disregards the scarcity of data samples. Currently, the predominant approaches to few-shot NER can be broadly categorized into two types: fine-tuning-based methods (Wang et al., 2022b; Schmidt et al., 2022) and metric-based methods (Chen et al., 2022b; Ma et al., 2022b).

Fine-tuning-based methods involve adjusting the parameters of NER models using new examples, whereas metric-based methods compare query tokens with prototypes of each entity class, representing entity types as vectors within a unified representation space alongside individual tokens. Researchers have proposed numerous enhancements to these two processing approaches. For instance, Huang et al. (2021) employed a distance-based method to explore self-training techniques using external data, and Wang et al. (2023) generated counterfactual instances as interventions to augment the original dataset. Furthermore, prompt-based learning methods are widely applied in Few-shot NER (Chen et al., 2022b). Das et al. (2022) combined contrastive learning with prompt learning to better represent label dependencies.

However, the direct application of these few-shot NER methods to nested NER presents several challenges. Using fine-tuning-based methods, in a nested NER context, inconsistencies in entity labels (Straková et al., 2019) make it difficult for classifiers trained in the source domain to transfer effectively to the target domain (Wang et al., 2020). Metric-based methods struggle to distinguish semantic entities with only a few samples because of the similar semantic feature representations of nested entities. Moreover, prompt-based learning faces challenges because nested entity spans may exhibit varying dependency patterns, making it challenging to glean sufficient information from prompt learning to identify nested entities, particularly for rare or complex nesting structures (Ming et al., 2022; Huang et al., 2022).

In the context of nested NER, these methods require more nuanced adaptation and optimization to overcome the challenges posed by data complexity. Faced with the complexity of simultaneous few-shot NER and nested NER as well as the dual challenge of limited training data and the presence of nested entities, we propose a novel span-based meta-learning framework combined with a convolutional processing approach (MCFSN) to address the issue of few-shot nested NER. Our method integrates the meta-learning framework with multi-sample concatenation as soft prompts, effectively addressing challenges in few-shot NER. Specifically, in processing nested NER, we proceed in two stages. first, in the span detection stage, sentence features are extracted using Convolutional Neural Networks (CNN), and word features are regularized using a Biaffine attention mechanism to capture interactions between sentences and accurately represent word features. Second, in the entity span classification stage, the model combines the ProtoNet and Biaffine classifiers, and employs a fully connected layer output for labeling, thereby enhancing the model's span classification capability.

Our main contributions are as follows:

- We introduce a novel span-based approach for few-shot nested Named Entity Recognition, using a Meta-Learning Convolutional Model (MCFSN). This model is the first to leverage meta-learning in conjunction with CNNs to addressing the challenges of few-shot nested NER.

- Our approach employs CNNs for extracting high-dimensional sentence features. The meta-learning framework effectively utilizes information from few-shot samples and, in combination with soft prompts and Biaffine classifiers, further enhances the model's ability to discriminate in few-shot nested NER.

- Experimental results demonstrate that the MCFSN model achieves state-of-the-art performance on three benchmark datasets (GENIA, GermEval, NEREL). It surpasses several competing models in F1 scores, achieving 6.9 increase in F1 score on the GENIA, 5.2 improvement on the GermEval, and 4.5 enhancement on the NEREL.

2 Related Work

2.1 Few-shot NER

Current mainstream few-shot NER methodologies can be classified into two primary categories: fine-tuning-based methods (Wang et al., 2022b) and metric-based methods (Chen et al., 2022a). To improve performance, researchers have proposed various enhancements, including the use of label information to augment model recognition capabilities (Hou et al., 2022), and the design of new paradigms (Chen et al., 2022a), such as prompt-based methods (Wang and Liu, 2021; Das et al., 2022; Ma et al., 2022c). However, these approaches, which primarily focus on flat NER, are not directly applicable to nested NER.

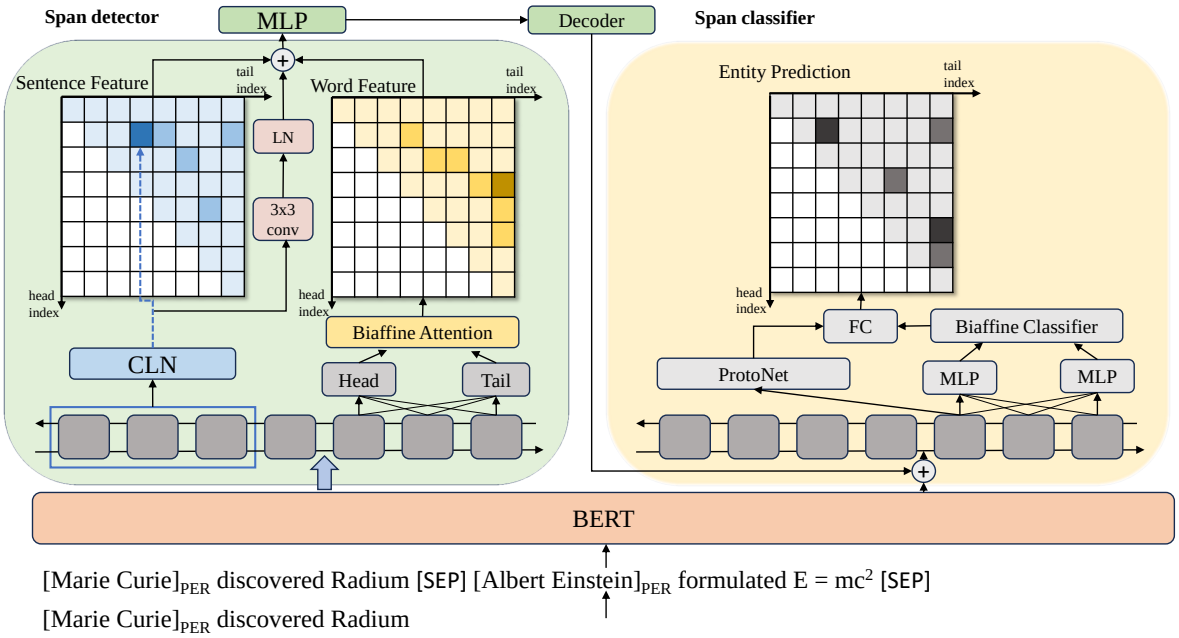

Figure 2: The few-shot nested NER meta-learning convolutional model incorporates Conditional Layer Normalization (CLN) and a Multi-Layer Perceptron (MLP). The symbol $\oplus$ denotes vector concatenation, and 'FC' refers to a fully connected layer. The Decoder employs Viterbi decoding combined with softmax prediction.

2.2 Nested NER

Early approaches to nested NER predominantly relied on rule-based methods, which depended on manually crafted rules for identifying and classifying entities (Shen et al., 2021). Although effective in certain scenarios, these rule-based methods lacked flexibility and struggled to adapt to nested entity types not covered by the rules (Patil et al., 2023). In recent years, the mainstream methodology has shifted towards fully supervised learning approaches, including neural transformer-based methods (Tual et al., 2023), hypergraph-based methods (Katiyar and Cardie, 2018; Wang and Lu, 2018), region-specific identification methods (Lin et al., 2019), and span-based methods (Shen et al., 2021; Wan et al., 2022; Zhu and Li, 2022). However, these methods require a substantial amount of labeled data and are not suitable for few-shot settings.

2.3 Few-shot Nested NER

To the best of our knowledge, existing work on few-shot nested NER primarily focuses on exploring effective methods to address the dual challenges of nested entity structures and limited training samples. Ming et al. (2022) were pioneers specifically studying the task of few-shot nested NER. They introduced a Biaffine-based Contrastive Learning Framework (BCL) to tackle this task. This framework employs a Biaffine span representation module to learn the contextual span dependencies of each entity span, merging dependency and semantic representations to differentiate nested entities.

Subsequently, the FIT model by Xu et al. (2023) observed that entity spans and their nested counterparts may have distinct dependency models. This model adjusts representations through contrastive learning, enhancing the similarity within spans of the same entity category and reducing it between different categories. This method, by measuring similarity, enhances transfer learning capabilities for addressing few-shot.

3 Method

In this section, the task definitions for nested NER and few-shot NER are first introduced, followed by a detailed description of the MCFSN framework. Finally, our training objectives are presented.

3.1 Overall Architecture

Definition of Few-Shot Nested NER: Given an input sentence $\mathbf{x} = \{\mathbf{x_1}, ..., \mathbf{x_n}\}$ consisting of n word tokens, the objective is to accurately identify the left and right boundary tokens $\mathbf{x}_{\text{el}}$ and $\mathbf{x}_{\text{er}}$ of each entity $\mathbf{e} = \{\mathbf{x}_{\text{el}}, ..., \mathbf{x}_{\text{er}}\}$ within x, and to assign the correct entity type to e from a predefined list of

categories Y, such as Y = {"GPE", "ORG", ...}. Unlike Flat NER, in nested NER, entities may overlap, with tokens within entity e potentially being assigned multiple types. The model follows the standard few-shot NER setting described by Ding et al. (2021), typically training on source domain data and addressing the N-way (N unseen classes) K-shot (K annotated examples per class) task in the target domain.

The process of few-shot nested NER in our study is divided into two stages: span extraction and span classification. The overall approach employs a meta-learning framework to address the challenges posed by few-shot. Figure 2 illustrates our few-shot nested NER meta-learning convolutional model.

3.2 Entity span detector

Given an input sentence $\mathbf{x} = \{\mathbf{x_1}, ..., \mathbf{x_n}\}$ containing n word tokens, the BIOES tagging scheme is used to provide more specific and granular boundary information for entity spans. This entails labeling each word $\mathbf{x_1}$ in the sentence with $\mathrm{y_i} \in \{\mathbf{B}, \mathbf{I}, \mathbf{O}, \mathbf{E}, \mathbf{S}\}$ to denote its position within an entity span.

Each entity label in the input sentence is augmented with samples of the same type because the label name contains not only entity information but also label details. This augmentation is concatenated to the sample as a soft prompt template to enhance the model's ability to utilize few-shot information. Specifically, the input sentence is formatted as $\mathbf{x} = \{\mathbf{x}[\mathbf{SEP}]\mathbf{x_{se}}\}$, where $\mathbf{x_{se}}$ represents a sample of the same type of entity, and $[\mathbf{SEP}]$ serves as a delimiter to distinguish between the enhanced instance and the input sentence.

We use BERT (Kenton and Toutanova, 2019) as the encoder for our model, as it has been proven to be one of the state-of-the-art models for representation learning in NER (Wang et al., 2021). The augmented text is input into the BERT encoder to obtain the embedding vectors $\mathbf{h} \in \mathbf{R}^{n \times d}$, where d represents the dimension of BERT's hidden states. For each token xi, the BERT token generator can divide it into multiple subtokens $\mathrm{t_i} = (\mathrm{t_{i_1}}, \ldots, \mathrm{t_{i_j}})$.

$$t_i = \mathrm{BERT}(x_i) \tag{1}$$

To further enhance context modeling, following prior work (Wadden et al., 2019), we employ a Bidirectional LSTM (BiLSTM) as described by Lample et al. (2016b). The embedding vectors $\mathrm{E} = \{\mathrm{e_1}, \ldots, \mathrm{e_n}\}$, outputted by BERT, are input into the BiLSTM to generate the final word representation vectors, denoted as $\mathrm{H} = \{\mathrm{h_1}, \ldots, \mathrm{h_n}\} \in \mathbf{R}^{\mathrm{n \times d}}$.

$$\overrightarrow{h}_i = \overrightarrow{\mathrm{LSTM}}(t_i, \overrightarrow{h}_{i-1}) \tag{2}$$

$$\overleftarrow{h}_i = \overleftarrow{\mathrm{LSTM}}(t_i, \overleftarrow{h}_{i-1}) \tag{3}$$

$$h_i = [\overrightarrow{h}_i; \overleftarrow{h}_i] \tag{4}$$

The symbol $[;]$ denotes concatenation, and h_i represents a 2d-dimensional vector.

Inspired by Li et al. (2020), the output vectors H from the BiLSTM are processed using Conditional Layer Normalization (CLN) to generate a word pairs grid. This grid can be conceptualized as a two-dimensional matrix $\mathbf{M}$, where $\mathbf{M} \in \mathbf{R}^{\mathrm{nxd}}$, to predict the relationships between word pairs $\mathbf{M}_{\mathrm{ij}}(\mathrm{x_i}, \mathrm{x_j})$.

$$\mathbf{M}_{ij} = \mathrm{CLN}(h_i, h_j) = \lambda_{ij} \odot (\frac{h_j - \mu}{\sigma}) + \phi_{ij} \tag{5}$$

where the layer normalization gain parameter is generated as $\lambda_{ij} = \mathrm{W}_\alpha \mathrm{h}_i + \mathrm{b}_\alpha$, and $\phi_{ij} = \mathrm{W}_\beta \mathrm{h}_i + \mathrm{b}_\beta$. μ and σ represent the mean and variance of the elements in $\mathrm{h_j}$, respectively. W_α , b_α , W_β and b_β are all learnable parameters.

Given that CNNs are well-suited for performing 2D convolutions on grids and exhibit excellent characteristics in processing representational relationships (Zeng et al., 2018), we employ a 3x3 convolution as a feature refiner. Coupled with layer normalization, this approach aims to capture the interactions of different spans within a sentence.

$$\mathrm{C} = \mathrm{CLN}\left(\mathrm{Conv}(\mathbf{M})\right) \tag{6}$$

Subsequently, Biaffine Attention is employed to represent the current word's features through head and tail characteristics, enhancing MLP prediction (Li et al., 2021). Biaffine Attention can be viewed as a method for modeling the pairwise interaction relationships between elements in the sequence H output by BiLSTM.

$$A_{ij} = h_i^T W h_j + \mathrm{U}^T (h_i \oplus h_j) + b \tag{7}$$

Wherein W represents the weight matrix for the bilinear terms, U is the weight matrix for the linear terms, and b denotes the bias. $\oplus$ indicates vector concatenation.

The outputs from Biaffine Attention, CLN, and layer normalization are then concatenated and fed into an MLP to amalgamate information, with the

expectation of capturing both the regional information and internal patterns necessary for predicting spans.

$$F = \mathrm{MLP}(A \oplus C \oplus M) \tag{8}$$

Entity spans are obtained by employing Viterbi decoding and softmax prediction, selecting the results with the highest probability.

$$P(f_i) = \frac{e^{f_i}}{\sum_{j=1}^{n} e^{f_j}} \tag{9}$$

$$\widehat{S} = \operatorname{argmax} \prod_{i=1}^{m} P\left(f_i\right) \tag{10}$$

Herein, m denotes the length of the sequence, and f_i represents the state at the ith position in the sequence.

Our training objective is to minimize the discrepancy between predicted probabilities and actual labels, thereby enabling the model to accurately identify entity spans. The loss function used in the entity span detection stage employs cross-entropy loss.

$$\mathcal{L}_{detector} = -\frac{1}{n^2} \sum_{ij} \mathrm{CrossEntropyLoss}(y_{ij}, p_{ij}) \tag{11}$$

The actual label y_{ij}, which is either 0 or 1, signifies whether the word pairs formed by the i_{th} and j_{th} words in a sentence are part of a valid entity span. p_{ij} is the probability predicted by the model for these word pairs to belong to a valid entity span. The term n^2 denotes the total number of possible word pairs combinations within the sentence.

3.3 Entity span classify

For the entity spans extracted in the entity span detection stage, they are concatenated and integrated with the output from BERT, and then processed through a BiLSTM. This approach is expected to enable the model to fully utilize the information from the existing few-shot instances.

$$h_{new} = [t_1 \oplus s_1, t_2 \oplus s_2, ..., t_n \oplus s_m] \tag{12}$$

$$\overrightarrow{h_l} = \overrightarrow{\mathrm{LSTM}}\left(\overrightarrow{h_{l-1}}, h_{new,i}\right) \tag{13}$$

$$\overleftarrow{h_l} = \overleftarrow{\mathrm{LSTM}}\left(\overleftarrow{h_{l+1}}, h_{new,i}\right) \tag{14}$$

$$H_{new} = \left[\overrightarrow{h_1} \oplus \overleftarrow{h_1}, \overrightarrow{h_2} \oplus \overleftarrow{h_2}, ..., \overrightarrow{h_n} \oplus \overleftarrow{h_n}\right] \tag{15}$$

In this context, t_i and s_i respectively represent the embedding vectors output by BERT and the entity spans extracted during the entity span detection stage, with $\oplus$ indicating vector concatenation.

FlatNER is processed using ProtoNet.Assuming $\mathrm{H}_{\mathrm{new,[i,j]}}$ is the entity span output from the entity span detection stage, spanning from h_i to h_j, the span representation of $\mathrm{H}_{\mathrm{new,[i,j]}}$ is calculated by averaging the representations of all tokens within$\mathrm{H}_{\mathrm{new,[i,j]}}$.

$$\sup_{[i,j]} = \frac{1}{j-i+1} \sum_{k=i}^{j} h_k \tag{16}$$

Let $\mathrm{S_k} = \{\mathrm{H}_{\mathrm{new,[i,j]}}\}$ denote the set of entity spans contained in the given support set S, corresponding to the entity class γ_{k} in the set γ. For each entity class γ_{k}, the average span representation is computed to serve as the prototype p_k.

$$p_k(S) = \frac{1}{|S_k|} \sum_{x_{[i,j]} \in S_k} sup_{[i,j]} \tag{17}$$

Utilizing the given training set (S_{train}, Q_{train}, γ_{train}), the prototypes for all entity classes in γ_{train} are calculated in S_{train} using Equation 17. For each span's $\mathrm{H}_{\mathrm{new,[i,j]}}$ in the query set Q_{train}, the Euclidean distance between $\mathrm{H}_{\mathrm{new,[i,j]}}$ and the prototype of each category is computed to determine the class of $\mathrm{H}_{\mathrm{new,[i,j]}}$.

$$dist = \parallel H_{new,[i,j]} - p_k(S_{train}) \parallel_2 \tag{18}$$

$$P_{Proto}\left(p_k|\Pi_{new,[i,j]}\right) = \frac{\exp\left(-dist\right)}{\sum_{k'} \exp\left(-dist'\right)} \tag{19}$$

For nested NER, Biaffine classifier is used, with the category of each span being determined through calculations performed by the Biaffine layer.

$$Bia = h_s^T W_1 h_e + W_2(h_s \oplus h_e) + b \tag{20}$$

$$P_{Biaffine}\left(p_k|H_{new,[i,j]}\right) = softmax\left(Bia\right) \tag{21}$$

Wherein h_s and h_e denote the feature vectors of the span's start and end positions, respectively. W_1 and W_2 are weight matrices, and b represents the bias.

The probability of an entity span's category is generated by concatenating the outputs of ProtoNet and Biaffine classifier, and then inputting them into a fully connected layer. This method is employed to predict the category of an entity span more accurately.

$$P = \mathrm{FC}\Big(P_{\mathrm{Proto}} \oplus P_{\mathrm{Biaffine}}\Big) \tag{22}$$

Dataset	5-shot Nested ratio(%)	10-shot Nested ratio(%)	Sentence	Entities/Nest entities
GENIA	16	18	18k	54.3k/28.4k
GermEval	10	11	17.7k	39.8k/5.3k
NEREL	23	28	8.6k	53.8k/17.2k
FewNERD	-	-	188.2k	491.7k/-

Table 1: Data Scale of the Datasets Used in the Experiment.

Parameter	Values
Learning rate	{1e-5, 3e-5, 1e-4}
ML fine-tune steps	{1, 2, 3, 5, 9, 10, 20}
Dropout	{0.1, 0.3, 0.5}
BERT learning rate	{5e-6, 1e-5, 2e-5}
n and λ	{0.35-0.65}
Batch size	{16, 32}
Type similarity threshold	{1, 2.5, 3, 4, 5}

Table 2: Hyper-parameters search space used in our experiments.

The loss function employed in the classification stage utilizes cross-entropy loss.

$$\mathcal{L}_{classify} = -\sum_{c} y_c \log(P_c) \tag{23}$$

where y_c represents the $c_t h$ element of y, indicating whether the entity span belongs to category c. P_c is the $c_t h$ element of P, denoting the probability of being predicted as category c.

3.4 Meta-learning framework

The objective of meta-learning is to enable the model to quickly adapt to few-shot tasks that it has never encountered before. Meta-learning frameworks consist of two stages: meta-training and meta-testing. The existing model M_θ undergoes repeated meta -training, followed by fine-tuning the trained model $M_{\theta'}$ using the novel episode support set. This is then evaluated on the corresponding query set.

During the meta-training phase, the model randomly samples an episode $(S_{train}, Q_{train}, \gamma_{train})$ from the source domain dataset ϵ_{train} to simulate test. Subsequently, the parameters θ of the model M_θ undergo n steps of inner updates, with the update rule being:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta; S_{\text{train}}) \tag{24}$$

$$\mathcal{L}_{total} = \eta \mathcal{L}_{detect} + \lambda \mathcal{L}_{classify} \tag{25}$$

Herein, α represents the learning rate, $\mathcal{L}_{total}$ is the loss function, and η and λ are hyperparameters.

Subsequently, the updated model M_θ is evaluated on the query set Q_{train} for meta-updates.

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_i \mathcal{L}\left(\theta; Q^{(i)}_{\text{train}}\right) \tag{26}$$

Here, β is the meta-learning rate. During the meta-testing phase, the model is evaluated on test tasks. After meta-training with $\mathcal{L}_{total}$, the model is fine-tuned to obtain M_{θ^*}, and then this fine-tuned model is used to make predictions on new query examples Q_{new}.

4 Experiments

4.1 Settings

4.1.1 Datasets

The FewNERD[1](Ding et al., 2021) dataset was utilized as the source domain dataset, and experiments were conducted on three datasets GENIA[2](Kim et al., 2003), NEREL[3](Loukachevitch et al., 2021), GermEval[4](Benikova et al., 2014) to evaluate the proposed method. Refer to Appendix A.1 for an overview of the datasets.

4.1.2 Domain settings

FewNERD serves as the source domain dataset, which can be divided into inter and intra parts. The experiments on FewNERD inter subset follow the processing method described in the original paper by Ding et al. (2021), involving random sampling to extract N-way N-shot subtasks for meta-learning training. Similarly, during the testing process, the FewNERD sampling procedure is followed, randomly selecting N-way N-shot as the fine-tune dataset on the target domain datasets. For the NEREL, GermEval, and GENIA datasets, the sampled datasets are derived from the test portions provided in the original datasets, excluding some entity categories with fewer than 50 entities. The final data scale is presented in Table 1.

[1] https://github.com/thunlp/Few-NERD
[2] http://www.geniaproject.org/genia-corpus
[3] https://github.com/nerel-ds/NEREL
[4] https://germeval.github.io/

Dataset	Method	5-shot			10-shot		
		P	R	F1	P	R	F1
GENIA	SEE-Few	30.92	14.41	$19.31_{\pm 6.95}$	52.35	29.84	$37.78_{\pm 5.04}$
	SDNet	41.25	11.36	$17.48_{\pm 6.97}$	48.57	12.18	$19.03_{\pm 7.07}$
	ESD	36.44	20.24	$25.03_{\pm 9.88}$	48.84	28.00	$35.23_{\pm 4.96}$
	FIT	40.72	30.30	$34.43_{\pm 9.06}$	52.91	39.51	$44.95_{\pm 3.38}$
	BCL	\	\	$46.06_{\pm 1.22}$	\	\	$62.33_{\pm 1.86}$
	Ours	**60.87**	**48.62**	$\mathbf{52.96}_{\pm 6.96}$	**73.82**	**60.38**	$\mathbf{67.45}_{\pm 9.33}$

Table 3: Performance comparison of MCFSN and baselines on GENIA datasets under different shots.

Datasets	Method	1-shot(F1)	5-shot(F1)
GermEval	NNShot	$28.58_{\pm 6.76}$	$41.26_{\pm 2.50}$
	ProtoNet	$19.05_{\pm 1.71}$	$28.59_{\pm 2.32}$
	CONTaiNER	$33.18_{\pm 6.03}$	$42.38_{\pm 2.61}$
	BCL	$39.56_{\pm 5.69}$	$47.07_{\pm 2.94}$
	Ours	$\mathbf{44.76}_{\pm 11.73}$	$\mathbf{50.22}_{\pm 8.96}$
NEREL	NNShot	$38.58_{\pm 1.30}$	$46.54_{\pm 1.93}$
	ProtoNet	$17.76_{\pm 1.78}$	$23.16_{\pm 3.19}$
	CONTaiNER	$35.23_{\pm 2.31}$	$53.55_{\pm 1.14}$
	BCL	$44.47_{\pm 1.60}$	$\mathbf{58.95}_{\pm 1.64}$
	Ours	$\mathbf{48.95}_{\pm 8.76}$	$54.33_{\pm 6.95}$

Table 4: Performance comparison of MCFSN and baselines on GermEval and NEREL datasets under different shots.

4.1.3 Implementation Details

In our experimental setup, the model was implemented using the Pytorch framework, version 1.9.0. The Pretrained Language Model (PLM) BERT-base-uncased from the Huggingface Library (Wolf et al., 2020) was employed as our primary encoder, in accordance with the requirements set by Ding et al. (2021). The embedding layer parameters were frozen during optimization, with a learning rate of 5e-6 during the Few-NERD learning period. For optimization, we use AdamW (Loshchilov and Hutter, 2018) with a learning rate of 3e-5 as the optimizer, complemented by a linear warm-up phase accounting for 1% of the training. Hyperparameter settings are determined using grid search, with the search space outlined in Table 2. For more details, please refer to the Appendix A.2

4.1.4 Baselines

The following models are used as baselines for few-shot nested NER: **ProtoNet** (Snell et al., 2017), **NNShot** (Yang and Katiyar, 2020), **CONTaiNER** (Das et al., 2022), **SEE-Few** (Yang et al., 2022), **SDNet** (Chen et al., 2022a), **ESD** (Wang et al., 2022b), **FIT** (Xu et al., 2023), **BCL** (Ming et al., 2022). The range of applicable methods is constrained by the inability of most few-shot NER approaches to address few-shot nested NER challenges. For more details, Appendix A.3 should be consulted.

4.2 Main Results

Tables 3 and 4 display the main results of our method, averaged over five experiments, and compare them with the results of previous advanced methods.

It is evident that our proposed method significantly outperforms previous approaches. On the Genia, it achieved 6.9% increase in F1 score, on the GermEval, 5.2% enhancement in F1 value, and on the NEREL , 4.5% improvement in F1 score, effectively demonstrating the effectiveness of our approach.

Tables 3 and 4 illustrate the model's results on multilingual datasets, where our method achieves impressive performance, particularly in the presence of significant language domain differences compared to most other methods. For instance, it facilitates the transfer and alignment of information across different language datasets and even

Method	5-shot			10-shot		
	P	R	F1	P	R	F1
Full model	58.61	48.73	**52.96**$_{\pm 6.96}$	72.82	62.38	**67.45**$_{\pm 9.33}$
-w/o CNN feature	44.25	24.14	31.21$_{\pm 9.97(-21.75)}$	50.69	32.55	39.75$_{\pm 8.47(-27.7)}$
-w/o Biaffine classifier	51.84	36.27	43.07$_{\pm 7.88(-9.89)}$	66.06	52.91	58.23$_{\pm 6.96(-9.22)}$
-w/o meta learning	56.72	43.08	48.73$_{\pm 8.06(-4.23)}$	73.71	62.47	**68.35**$_{\pm 4.19(+1.1)}$
-w/o soft prompts+Viterbi	57.76	45.44	51.06$_{\pm 6.22(-1.9)}$	73.66	62.05	67.13$_{\pm 9.86(-0.32)}$

Table 5: Ablation study of MCFSN and baselines on the GENIA dataset under different shots.

disparate entity classes, thereby constructing a unified model for diverse domains.

It demonstrates the strong adaptability of our approach.

4.3 Ablation Study

To verify the contribution of different components in the proposed method, ablation experiments were conducted on the Genia dataset from the perspective of modules and implementation. The following variants and baselines for ablation study were introduced:

w/o CNN feature: This variant involves extracting features solely using word features, without employing CNN for feature extraction of the combined information. Subsequently, the model is used for detecting nested NER and fine-tuned with few-shot samples.

w/o biaffine classifier: In this variant, span classification for nested NER is conducted solely using ProtoNet, without employing a biaffine classifier.

w/o meta learning: This variant involves training the detection and classification models using traditional gradient descent methods, without the use of meta-learning.

w/o soft prompts+Viterbi: In this variant, classification is performed using the original sentence input and softmax prediction, without employing soft prompts and Viterbi decoding.

As shown in Table 5. Generally, the removal of any single module leads to a decrease in performance. Furthermore, Table 5 also allows for some in-depth observational conclusions.

1) The omission of CNN features for feature extraction results in a significant decline in performance. This indicates that relying solely on word features during the feature extraction stage captures only a limited amount of useful information, insufficient to meet the demands of few-shot NER and nested NER.

2) Upon removing the biaffine classifier, results indicate that this module exhibits good nested NER classification performance in the 5-shot. As the training data increases, the classification effectiveness of the biaffine classifier module also improves. This improvement is due to the span classification stage gaining a stronger ability to differentiate nested NER as the volume of data increases.

3) Meta-learning is essential for the application of models in few-shot NER. The fact that the positive effect in a 5-shot setting surpasses that in a 10-shot setting indicates that the removal of meta-learning leads to a noticeable decrease in results when training data is scarce. Concurrently, meta-learning also aids the model in effectively detecting and classifying NER in few-shot contexts.

4) Ablation studies have shown that the use of soft prompt templates and Viterbi decoding aids in extracting valuable information from a limited number of samples, thereby enhancing the model's capacity to learn nested NER. As a result, the model can maintain a robust capability for feature extraction and language modeling.

5 Conclusion

This study introduces an innovative meta-learning convolutional approach for few-shot NER, adeptly combining the strengths of meta-learning and convolutional neural networks. This approach, demonstrated through a two-stage process involving CNN and dual affine attention mechanisms for span feature extraction followed by effective entity span classification, shows considerable promise. Notably, the inclusion of mete-learning and CNN has proven crucial, enabling the model to utilize context effectively for entity knowledge recall. The results from ablation studies and performance evaluations indicate a substantial enhancement in the model's capability to learn nested NER. Our method not only achieves significant F1 score improvements but also demonstrates robust adaptability and efficiency in few-shot scenarios.

6 Limitations

- Dataset Specificity: Although effective across the tested datasets, the proposed MCFSN method may not generalize equally well to all types of nested NER, particularly those with highly idiosyncratic or domain-specific language structures.

- Dependency on High-Quality Span Features: The performance of MCFSN method heavily relies on the accurate extraction of span features using CNN and Biaffine attention mechanisms. If these initial features are not extracted effectively, this could significantly impact the overall accuracy of nested entity recognition. Moreover, it is important to investigate alternative approaches for modeling such relationships, exploring novel perspectives that could enhance the model's adaptability and effectiveness in diverse scenarios.

References

Darina Benikova, Chris Biemann, and Marc Reznicek. 2014. Nosta-d named entity annotation for german: Guidelines and dataset. In *LREC*, pages 2524–2531.

Jiawei Chen, Qing Liu, Hongyu Lin, Xianpei Han, and Le Sun. 2022a. Few-shot named entity recognition with self-describing networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5711–5722.

Yanru Chen, Yanan Zheng, and Zhilin Yang. 2022b. Prompt-based metric learning for few-shot ner. *arXiv preprint arXiv:2211.04337*.

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.

Leyang Cui and Yue Zhang. 2019. Hierarchically-refined label attention network for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4115–4128, Hong Kong, China. Association for Computational Linguistics.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca J 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.

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.

Yutai Hou, Cheng Chen, Xianzhen Luo, Bohan Li, and Wanxiang Che. 2022. Inverse is better! fast and accurate prompt for few-shot slot tagging. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 637–647.

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

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.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016a. 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.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016b. 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.

Dong-Ho Lee, Akshen Kadakia, Kangmin Tan, Mahak Agarwal, Xinyu Feng, Takashi Shibuya, Ryosuke Mitani, Toshiyuki Sekiya, Jay Pujara, and Xiang Ren. 2022a. 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.

Dong-Ho Lee, Akshen Kadakia, Kangmin Tan, Mahak Agarwal, Xinyu Feng, Takashi Shibuya, Ryosuke Mitani, Toshiyuki Sekiya, Jay Pujara, and Xiang Ren. 2022b. 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.

Jing Li, Aixin Sun, and Yukun Ma. 2020. Neural named entity boundary detection. *IEEE Transactions on Knowledge and Data Engineering*, PP(99):1–1.

Jingye Li, Kang Xu, Fei Li, Hao Fei, Yafeng Ren, and Donghong Ji. 2021. Mrn: A locally and globally mention-based reasoning network for document-level relation extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1359–1370.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5182–5192.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Natalia Loukachevitch, Ekaterina Artemova, Tatiana Batura, Pavel Braslavski, Ilia Denisov, Vladimir Ivanov, Suresh Manandhar, Alexander Pugachev, and Elena Tutubalina. 2021. Nerel: A russian dataset with nested named entities, relations and events. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 876–885.

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.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuan-Jing Huang. 2022b. 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.

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

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.

Hong Ming, Jiaoyun Yang, Lili Jiang, Yan Pan, and Ning An. 2022. Few-shot nested named entity recognition. *arXiv preprint arXiv:2212.00953*.

Archana Patil, Shashikant Ghumbre, and Vahida Attar. 2023. Named entity recognition over dialog dataset using pre-trained transformers. In *International Conference on Data Management, Analytics & Innovation*, pages 583–591. Springer.

Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108*.

Fabian David Schmidt, Ivan Vulic, and Goran Glavas. 2022. Don't stop fine-tuning: On training regimes for few-shot cross-lingual transfer with multilingual language models. In *EMNLP 2022*.

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.

Shashank Sonkar, Zichao Wang, and Richard G Baraniuk. 2022. Maner: Mask augmented named entity recognition for extreme low-resource languages. *arXiv preprint arXiv:2212.09723*.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested ner through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331.

Solenn Tual, Nathalie Abadie, Joseph Chazalon, Bertrand Duménieu, and Edwin Carlinet. 2023. A benchmark of nested named entity recognition approaches in historical structured documents. In *International Conference on Document Analysis and Recognition*, pages 115–131. Springer.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789.

Juncheng Wan, Dongyu Ru, Weinan Zhang, and Yong Yu. 2022. Nested named entity recognition with span-level graphs. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 892–903.

Bailin Wang and Wei Lu. 2018. Neural segmental hypergraphs for overlapping mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 204–214.

Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504.

Huiming Wang, Liying Cheng, Wenxuan Zhang, De Wen Soh, and Lidong Bing. 2023. Enhancing few-shot ner with prompt ordering based data augmentation. *arXiv preprint arXiv:2305.11791*.

Jue Wang, Lidan Shou, Ke Chen, and Gang Chen. 2020. Pyramid: A layered model for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5918–5928.

Liwen Wang, Rumei Li, Yang Yan, Yuanmeng Yan, Sirui Wang, Wei Wu, and Weiran Xu. 2022a. Instructionner: A multi-task instruction-based generative framework for few-shot ner. *arXiv preprint arXiv:2203.03903*.

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.

Yaqing Wang, Haoda Chu, Chao Zhang, and Jing Gao. 2021. Learning from language description: Low-shot named entity recognition via decomposed framework. *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, et al. 2022c. Usb: A unified semi-supervised learning benchmark for classification. *Advances in Neural Information Processing Systems*, 35:3938–3961.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Yuanyuan Xu, Zeng Yang, Linhai Zhang, Deyu Zhou, Tiandeng Wu, and Rong Zhou. 2023. Focusing, bridging and prompting for few-shot nested named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2621–2637, Toronto, Canada. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. 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 5808–5822, Online. Association for Computational Linguistics.

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.

Zeng Yang, Linhai Zhang, and Deyu Zhou. 2022. Seefew: Seed, expand and entail for few-shot named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2540–2550.

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *arXiv preprint arXiv:1703.06345*.

Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 506–514, Melbourne, Australia. Association for Computational Linguistics.

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.

Enwei Zhu and Jinpeng Li. 2022. Boundary smoothing for named entity recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7096–7108.

A Appendix

A.1 Datasets

GENIA is a corpus focused on the biomedical field, predominantly covering articles in biology and molecular genetics. This dataset comprises 5 entity types, primarily centered around concepts such as proteins, genes, and cells. We adhere to the official partitioning scheme provided, which roughly maintains an 8:1:1 ratio for the tra/dev/test sets.

GermEval is a German nested NER dataset that includes 12 entity types. These are divided into four main categories: Person, Location, Organization, and Other, each with its subcategories.We split them into the train, dev, and test sets by 8:1:1.

NEREL is a Russian nested NER dataset, encompassing a total of 29 entity types. These types include basic categories such as Person, Organization, Location, Facility, and Geopolitical Entities. NEREL is currently the largest Russian corpus with annotations for entities and relationships, featuring nesting of named entities up to six layers. We roughly maintains an 8:1:1 ratio for the tra/dev/test sets.

FewNERD is a NER dataset specifically designed for few-shot learning, comprising data collected from multiple domains, such as news, literature, and academia. The dataset contains 66 entity types, covering a wide range of entity categories. For this dataset, we employ the same training and testing splits as outlined in the original paper.

All datasets are available under a usage license and can be downloaded online. In terms of data partitioning within the meta-learning training framework, the FewNERD dataset is used as the source domain data, while datasets from other domains serve as target datasets.

A.2 Implementation Details

The batch size is set to 32, with a maximum sequence length limit of 1024, and a dropout rate maintained at 0.1. The hidden state size of the BiLSTM encoder is set to 400. The output dimension of the MLP and the dropout rate are set to 150 and 0.2, respectively. In the total loss function, both η and λ are set to 0.5. During the meta-training phase, the number of inner update steps is set to 2. During the span detector tuning stage, the number of fine-tuning steps on the Few-NERD dataset is set to 3, while for other datasets, it is set to 10. The number of steps for the span classify phase is set similarly. In the meta-training query set evaluation phase, the maximum loss coefficient λ is set to 2. The model's effectiveness is validated on the development set every 100 steps. To optimize training speed, we retain only those entities whose similarity score to the nearest prototype exceeds a threshold of 2. The training is conducted on two Nvidia RTX 3090 GPUs, each with 24GB of memory, taking approximately 6 hours.

A.3 Baselines

ProtoNet learned a metric space in which classification could be performed by calculating distances to prototype representations of each class, offering a simple yet effective inductive bias.

NNShot utilized a supervised NER model, trained on the source domain, as a feature extractor. In the feature space, a nearest neighbor classifier was applied, employing a straightforward method to capture label dependencies among entity labels.

CONTaiNER optimized the distributional distances between tokens through contrastive learning. It leveraged Gaussian distribution embeddings to distinguish token categories, thereby effectively addressing the issue of domain overfitting in environmental training.

SEE-Few leveraged the context of mentioned entities and their types, utilizing a shared text encoder for joint learning to enhance performance.

SDNet utilized external data for joint training of mention descriptions and entity generation tasks. During the fine-tuning phase, mention descriptions were used to summarize type concept descriptions, followed by entity generation based on the generated descriptions.

ESD framed few-shot sequence tasks as a problem of matching span similarities between test queries and support entities. Sampling was conducted from the FewNERD and GENIA datasets at a specific ratio, followed by pretraining.

FIT achieved outstanding performance on four datasets by using a focus component and a bridge component to provide an accurate candidate range for the prompt component, without utilizing source domain data. The prompt component capitalized on the unique features of nested entities, classifying spans based on soft prompts and contrastive learning.

BCL employs Biaffine-based contrastive learning to differentiate nested entities using context dependencies. A Biaffine span representation mod-

ule is utilized to learn the context span dependency representation of each entity span. Nested entities are distinguished by merging the two representations through residual connections.