SiTNER: Improving Few-Shot Cross-lingual Nested Named Entity Recognition with high-quality pseudo-labels

Anonymous ACL submission

Abstract

Few-shot named entity recognition (NER) methods have shown preliminary effectiveness in flat tasks. However, existing methods still encounter difficulties when faced with cross-lingual and nested entity challenges due to the linguistic or nested structure gap. In this work, we propose a framework named SiTNER to deal with few-shot cross-lingual nested named entity recognition tasks. SiTNER mainly comprises two components: (1) contrastive span classification which could pull entities into corresponding prototype and generate high-quality pseudo-labels, and (2) masked pseudo data self-training which refine pseudo-labels and improves the span classification via self-training strategy. We train SiTNER on the English dataset and evaluate it on the English, German, and Russian datasets, and experimental results show our method could get comparable results.

1 Introduction

The few-shot Named Entity Recognition (NER) task, which aims to recognize unlabeled instances (query set) according to only a few labeled samples (support set), has recently been studied (Das et al., 2022; Wang et al., 2022c,a). Based on N-way K-shot task setting formulated by Li (Li et al., 2020a), few-shot NER methods could always apply the transfer learning strategy to enhance the model's adaptability to other tasks, based on a small set of labeled data. This involved training the model in a rich-resource domain (aka, source domain) with high-quality annotations, followed by transferring the model to the domain with limited labeled samples (aka, target domain). hese methods could be divided into several types, including but not limited to metric-learning-based (Snell et al., 2017; Hofer et al., 2018; Yang and Katiyar, 2020), meta-learning-based (Li et al., 2020a; Sung et al., 2018), prompt-tuning-based (Ma et al., 2022a; Hou et al., 2022), and contrastive-learning-based (Das et al., 2022). While these models have demonstrated good performance in traditional few-shot NER tasks, they still face challenges in addressing issues such as cross-language and nested entity recognition, as illustrated in figure 1.

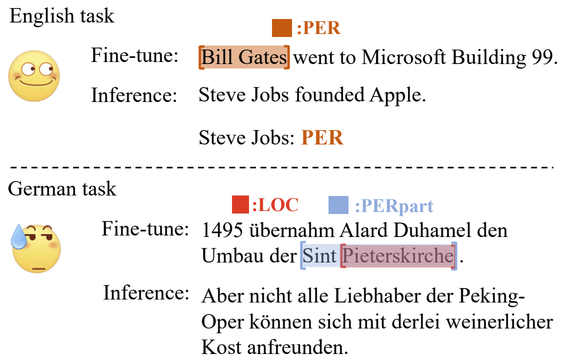

Figure 1: Traditional few-shot NER methods may perform well on flat and non-cross-lingual tasks. However, challenges persist when dealing with nested and cross-lingual tasks.

To bridge the linguistic gap between the source and target domain, semi-supervised learning (SSL) was raised to utilize the unlabeled data to enhance the labeled data and has been used in low-resource scenarios (Xie et al., 2020a; Yang et al., 2022). Self-training is a fundamental SSL strategy that can be described as a teacher-student framework. A teacher model is trained on the low-resource labeled data and generates pseudo labels based on the unlabeled data. Then, a student model is initialized and optimized by the pseudo labels of unlabeled data and shares the model parameters with the teacher model. Based on self-training, there are many works on instance-level tasks such as image classification (Wei et al., 2021; Wang et al., 2022b) and text classification (Kim et al., 2022; Tsai et al., 2022), and token-level tasks such as sequence labeling (Wang et al., 2023, 2021a). These methods mainly contribute to finding the noisy labels generated by the teacher model and avoiding error accumulation. Especially some pseudo-label sample strategies including Re-weighting (Wang et al., 2021a), Bayesian Token Selection (Wang et al., 2023), and Uncertainty-aware Selection (Rizve et al., 2021) mitigate the effect of noisy labels and alleviate the problem of confirmation bias. Although some self-training methods have been applied to deal with the few-shot sequence labeling (Wang et al., 2023; Qian and Zheng, 2022), the N-way K-shot cross-lingual nested NER tasks have not been explored previously.

To remedy this dilemma, we propose **S**elf-training h**i**gh-quality pseudo-label **T**uning, SiTNER, a novel few-shot nested NER framework for the few-shot cross-lingual nested NER task. Unlike existing data selection or re-weighting methods, SiTNER sufficiently leverages knowledge from unlabeled data in the target domain. SiTNER comprises two key components, namely contrastive span classification and masked pseudo data self-training. Firstly, we introduce a contrastive objective for cross-lingual NER tasks. Typical supervised contrastive learning methods (Das et al., 2022) treat labeled entities of the same/different class as positive/negative pairs and increase/decrease the similarity between positive/negative pairs. We further calculate the decision margin for each category of entity and force entities to fall within the decision margin via the backbone few-shot NER model. This could generate high-quality pseudo-labels for the unlabeled query set. Second, we insert high-quality pseudo-labels into the sentences in the support set and apply a masking strategy to reduce similarity with the original support set, resulting in a new dataset called the pseudo-label mask set. We then combine the pseudo-label mask set with the small support set and apply the contrastive learning strategy to refine the backbone model. As a result, the backbone few-shot NER model demonstrates improved performance on the challenging task of few-shot cross-lingual nested NER.

Our main contributions are as follows:

- The contrastive loss proposed by us enables the derivation of the prototype for each entity class and its corresponding decision margin for different tasks. Utilizing these decision boundaries, we can generate high-quality pseudo-labels for the unlabeled query set.

- We propose a method for generating pseudo-label datasets, which embeds high-quality pseudo-labels into the support set. This approach could mitigate the impact of nested structures on the model, addressing challenges in few-shot cross-lingual nested NER tasks.

- We train SiTNER on the English dataset and then make inferences on three nested NER datasets in three different languages. Our proposed SiTNER framework achieved comparable results across these three few-shot cross-lingual nested NER tasks, even using a basic pre-trained language model as the backbone.

2 Problem Definition

Following the mainstream solutions, we formulate the few-shot nested NER task as an entity span classification problem. Given a sentence x with l tokens, denoted by $x = \{w_1, \ldots, w_l\}$, we enumerate all possible spans and each span s_{pq} is a span of tokens starting from the p^{th} token and ending at the q^{th} token in x, denoted by $s_{pq} = \{w_p, \ldots, w_q\}$ $(1 \leqslant p \leqslant q \leqslant l)$. Then we represent a labeled dataset (aka. support dataset) and the unlabeled dataset (aka. query dataset) as $\mathcal{D}^{spt} = \{\mathcal{S}^{spt}, \mathcal{Y}^{spt}\}$ and $\mathcal{D}^{qry} = \{\mathcal{S}^{qry}\}$, respectively. $\mathcal{S}$ is the set of spans in sentences and $\mathcal{Y}$ is the set of corresponding labels of spans. The N-way K-shot setting of the few-shot nested NER task is making inferences for unlabeled $\mathcal{D}^{qry}$ with only a small size of $\mathcal{D}^{spt}$, which contain total N types of entity and K entities for each type.

3 Methodology

Figure 2 illustrates the overall framework of SiTNER. The framework consists of two main components: contrastive span classification and masked pseudo-data self-training.

3.1 Contrastive Span Classification

To get word embedding, we use ProtoBERT (Snell et al., 2017) as the backbone method of the SiTNER framework. This backbone method utilizes BERT (Devlin et al., 2019) as pre-trained language model (PLM) encoder to get token embeddings in the given sentence $x = \{w_1, \ldots, w_l\}$.

$$[\boldsymbol{h_1}, \boldsymbol{h_2}, \ldots, \boldsymbol{h_l}] = \mathrm{PLM}([w_1, w_2, \ldots, w_l]) \quad (1)$$

Then for a span s_{pq} which starts from the p^{th} token and ends at the q^{th} token in x, we could get the span representation

$$\boldsymbol{s_{pq}} = f(\boldsymbol{h_p} \oplus \boldsymbol{h_q}) \quad (2)$$

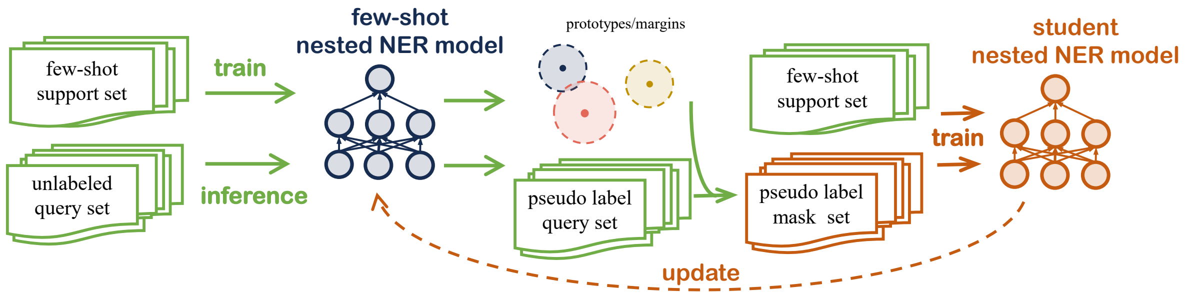

Figure 2: The overall framework of SiTNER. We begin by training a backbone model using the few-shot support set. This backbone model is then used to infer pseudo-labels for the unlabeled query set and to calculate prototypes and decision margins for each entity type in the support set. Subsequently, we employ these prototypes and decision margins to filter entities in the query set that fall within the decision margin. In the third step, the filtered entity results are combined with the small support set to create a new dataset (pseudo label mask set), which is then used to train a student model along with the support set. After the student model is trained, its parameters are shared with the backbone model.

$\oplus$ denotes the concatenation operator, and f is a non-linear activation function.

For the labeled support set, a multitude of spans is present within an input sentence, with a significant proportion of these spans belonging to the non-entity (O) category. Such a high prevalence of non-entity spans could impede the model's learning process. To mitigate this issue, we adopt a strategy of selecting all entity spans and a limited number of adjacent O-type spans for inclusion in the training sentence. After that, we generate prototypes $\boldsymbol{c}_i$ for type i in the support span set $\mathcal{S}^{spt}$:

$$\boldsymbol{c}_i = \frac{1}{|\mathcal{S}_i^{spt}|}\sum \mathcal{S}_i^{spt} \tag{3}$$

And the conventional ProtoBert methods will make inference of a span $\boldsymbol{s}^{qry}$ in the unlabeled query set and generate pseudo-label $\hat{y_s}^{qry}$ by the highest similarity with prototypes $\boldsymbol{c}$ in the support set $\mathcal{S}^{spt}$:

$$\boldsymbol{p}(\boldsymbol{s}^{qry}) = [d(\boldsymbol{c}_1, \boldsymbol{s}^{qry}), \ldots, d(\boldsymbol{c}_n, \boldsymbol{s}^{qry})] \tag{4}$$

$$\hat{y_s}^{qry} = argmax(\boldsymbol{p}(\boldsymbol{s}^{qry})) \tag{5}$$

Where $d(.)$ is the cosine similarity.

However, employing these inference results directly as pseudo labels for spans in the query set could result in numerous misclassified spans as illustrated in "not using decision margin" in the Appendix A. These lower-quality predicted pseudo-labels incorporated into the existing labeled few-shot dataset during the self-training strategy will lead to harmful results during the self-training step. Thus we have devised a decision margin to retain the high-quality pseudo-labels, which reduces the

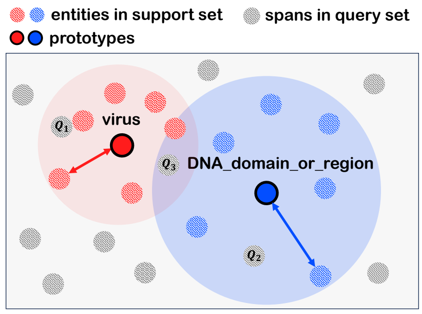

Figure 3: Illustration of the decision margin: The red/blue circles represent entity spans and their prototypes in the support set. We choose the entity span with the smallest cosine similarity to the prototype vector (i.e., the farthest Euclidean distance from the prototype) as the decision margin. The grey circles represent all spans in the unlabeled query set. If a span falls within the decision margin, the current label is assigned (as shown with Q_1 and Q_2 being assigned pseudo-labels "virus" and "DNA_domain_or_region" respectively). If a span falls at the intersection of multiple decision margins, the pseudo-label chosen is the one closer to the prototype (as seen with Q_3 being assigned "virus" in the illustration).

discrepancy between these predicted pseudo-labels and the real ground-truth labels.

After generating the prototypes $\boldsymbol{c}_i$ for each type i in the support set $\mathcal{S}^{spt}$ via Equation 3, we calculate the minimum cosine similarity (aka, farthest Euclidean distance) from the prototypes $\boldsymbol{c}_i$ to any spans within type i and utilize this minimum cosine similarity as the decision margin m_i for each type:

$$m_i = argmin(d(\boldsymbol{s}_{i1}, \boldsymbol{c}_i), d(\boldsymbol{s}_{i2}, \ldots, \boldsymbol{c}_i), d(\boldsymbol{s}_{in}, \boldsymbol{c}_i)) \tag{6}$$

Where $\boldsymbol{s}_{in}$ is the spans with type i in the $\mathcal{S}^{qry}$.

Figure 3 illustrates the process.

During the training step, we optimize the backbone model by calculating the loss for each span s:

$$\mathcal{L}_s = log\,(1 + pos * neg) \tag{7}$$

$$pos = \frac{\alpha \cdot e^{-d_p/\tau}}{1 + e^{(d_p - m_i)/\tau}} \tag{8}$$

$$neg = \sum \frac{(1-\alpha) \cdot e^{d_n/\tau} \cdot max(d_n - m_i, 0)}{1 + e^{-(d_n - m_i)/\tau}} \tag{9}$$

where α is a learnable parameter, τ is the temperature (Wang and Liu, 2021), d_p is the cosine similarity between current span s with the corresponding prototype of the same class, and d_n is the cosine similarity between current span s with the corresponding prototype of the different classes.

We adopt this loss function to maximize the similarity between spans in the query set that have the same class as their corresponding prototypes in the support set. Moreover, the further a sample is from its class center, the greater the magnitude of the pull force applied. On the other hand, for prototypes with different classes to the current span, we aim to push them away from each other and away from the corresponding class centers. If a sample is already outside the decision margin corresponding to its class center, there is no need to push it further away. Otherwise, the closer the sample is to its class center, the stronger the push force applied to move it farther away.

3.2 Masked Pseudo Data Self-training

In this section, we apply a self-training strategy to further optimize the performance of the backbone model. Specifically, we sample the spans and their corresponding pseudo labels in the unlabeled query set generated by the backbone model. In this way, we enhance the few-shot support set by sampled instances and further optimize the backbone model.

3.2.1 Self-training Instance Generation

Appendix A elucidates that for the unlabeled query dataset, we can filter entities within the decision margin m_i. This results in a relatively small number of selected entities, and through such filtering, a higher proportion of correctly predicted entities is achieved (i.e., cases where pseudo-labels match the true labels). We refer to these filtered entities as high-quality pseudo-labeled entity spans. Nevertheless, directly incorporating these high-quality

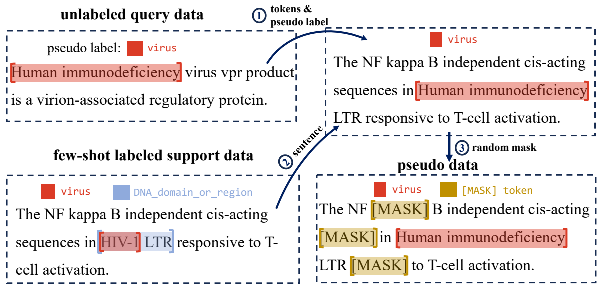

Figure 4: The process of generating self-training data: ① The backbone model identifies an entity (Human immunodeficiency) in the unlabeled query sentences and generates a pseudo-label (virus) for it. ② Based on the type of the pseudo-label, a sentence containing the chosen type is selected from the support set. The original entity (HIV-1) in the sentence is replaced with an entity corresponding to the pseudo-label. In contrast, other entities in different positions are re-labeled as "O", indicating non-entity (Human immunodeficiency LTR → "O"). ③ Random words except pseudo-label entity in the newly generated sentences are replaced with [MASK] tokens.

pseudo-labels along with their corresponding entities and sentences as self-training data and conducting contrastive learning training in comparison with the original support set is unwise. Given the nature of the nested entity task, some misidentifications may still adversely impact the model. For instance, consider a scenario where "HIV-1 LTR" is a "DNA_domain_or_region" entity but remains unrecognized by the model, while its nested subsegment "HIV-1" is identified as a "virus" entity. In such cases, directly incorporating "HIV-1" and its corresponding sentence into the model learning process is problematic as it overlooks the fact that "HIV-1 LTR" is also an entity.

Thus, for each high-quality pseudo label and the corresponding span, we insert the span into the original sentence in the few-shot support set which has at least one span that the type is the same as its pseudo label. To increase the dissimilarity between the new sentences and the original ones, we replace random word positions in the new sentences with "[MASK]" tokens, thus introducing a level of unpredictability. Figure 4 illustrates the process of generating masked pseudo data.

3.2.2 Self-training Algorithm

After generating Masked Pseudo Data, we apply a self-training approach to fine-tune the backbone model and improve the performance of the contrastive span classification component. The

Algorithm 1: self-training

Input: Totall self-training setps T, few-shot labeled data $\mathcal{S}^{spt}$, unlabeled data $\mathcal{S}^{qry}$

Initialize teacher model $\phi_{tea} = \theta^{(0)}$
for *self-training step* $t \leftarrow 1$ *to* T **do**
  Fine-tune teacher model on $\mathcal{S}^{spt}$
  Generate pseudo labels $\hat{y_s}^{qry}$ for spans in $\mathcal{S}^{qry}$
  Initialize the student model $\phi_{stu} = \theta^{(0)}$
  while *not converge* **do**
    generate new data $\mathcal{S}^{sudo}$ via pseudo labels and corresponding span in $\mathcal{S}^{qry}$ and the orgin sentences in $\mathcal{S}^{spt}$ according to Section 3.2.1
    Fine-tune the student model on $\mathcal{S}^{spt}$ and $\mathcal{S}^{sudo}$ Update the parameters of the student model $\phi_{stu}^{(t)}$
  end
  Update the parameters of the teacher model $\phi_{tea} = \phi_{stu}^{(t)}$
end

self-training framework involves using a teacher-student model. In our self-training strategy, we treat the backbone model as the teacher and employ the self-training algorithm to iteratively optimize the model. The overall algorithm is shown in Algorithm 1.

4 Experiments

In this section, we evaluate the performance of the proposed SiTNER framework in the few-shot nested NER setting. After introducing the rich-resource source domain dataset, three target domain datasets, and baseline models, we outline the experimental setup, present experimental results, and provide a thorough analysis.

4.1 Datasets

To better assess the performance and generality of our proposed SiTNER framework across different languages, we chose the Indo-European language family for our experiments, as obtaining datasets in these languages is readily feasible. We use English as the source language and English, German, and Russian as the target language.

As shown in Table 1, the target nested NER datasets are GENIA in English (Kim et al., 2003), GermEval in German (Benikova et al., 2014), and NEREL in Russian (Loukachevitch et al., 2021). We use a flat NER dataset, FewNERD in English (Ding et al., 2021), as the source domain dataset to train the model. All these datasets are publicly available under the licenses of CC-BY 3.0 for GENIA, CC-BY 4.0 for GermEval, CC-BY 2.5 for NEREL, and CC-BY-SA 4.0 for FewNERD. We have manually checked to guarantee these datasets are without offensive content and identifiers.

Dataset	language	Types	Sentences	Entities/Nest entities
GENIA	English	36	18.5k	55.7k / 30.0k
GermEval	German	12	18.4k	41.1k / 6.1k
NEREL	Russian	29	8.9k	56.1k / 18.7k
FewNERD	English	66	188.2k	491.7k / -

Table 1: Datasets used in experiments

4.2 Baselines

We compare SiTNER with nine baselines which can be categorized into three groups: 1) Rich-resource nested NER methods including NER-DP (Yu et al., 2020), TIdentifier (Shen et al., 2021), IoBP (Wang et al., 2021b), and PO-TreeCRFs (Fu et al., 2021); 2) Metric-based few-shot NER methods including ProtoBERT (Snell et al., 2017), NNShot (Yang and Katiyar, 2020), ESD (Wang et al., 2022c), and SpanProto (Wang et al., 2022a); 3) Contrastive-learning-based few-shot NER method CONTaiNER (Das et al., 2022). Appendix B details these baseline models.

4.3 Experiment Setup

In the training procedure, we utilized the FewNERD dataset, which could be decomposed into the inter- and intra-domain parts (Ding et al., 2021). We randomly sampled 5-way 5-shot subtasks from the FewNERD inter-domain subset for training, among which 10,000 subtasks as the training set and 500 subtasks as the validation set. We used the validation set to validate the framework for every 1000 subtasks during the training procedure. In the testing procedure, we first sampled several sentences in the target domain dataset as the support set. When sampling, we limited the number of entities in each entity category to k. Some sentences contain more than one entity. Thus, some entity categories may have more than k entities after the sampling procedure. We then fine-tuned the SiTNER on the support set and tested it on the query set. Here we chose 1-shot and 5-shot as the settings of the few-shot support set. Note that the model was trained in the FewNERD dataset and the target

Model	GENIA (32-way)		GermEval (12-way)		NEREL (29-way)		Avg
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	
NER-DP	15.26±2.78	31.89±4.01	7.12±2.61	24.89±3.92	15.86±5.77	42.25±2.42	22.87
TIdentifier	9.73±5.36	23.90±4.48	12.26±8.13	41.11±4.86	30.06±7.44	53.29±5.56	28.39
IoBP	16.09±2.07	31.67±3.31	3.32±2.04	12.86±2.60	8.61±1.23	18.50±1.46	15.17
PO-TreeCRFs	22.37±5.08	35.13±3.33	8.87±8.08	45.83±3.88	22.06±6.55	52.25±2.40	31.08
CONTaiNER	16.76±6.00	17.60±6.61	29.18±7.05	37.05±1.01	26.61±1.75	44.37±1.27	28.60
ProtoBERT	21.83±3.39	37.18±1.81	33.20±9.00	47.95±4.06	38.70±4.62	50.22±1.28	38.18
NNShot	25.72±4.75	33.77±2.57	28.58±6.76	41.26±2.50	38.58±1.30	46.54±1.93	35.74
ESD	19.96±3.93	25.31±3.17	34.00±8.75	34.75±6.03	28.56±5.18	47.68±2.20	31.71
SpanProto	**31.39**±2.86	**43.14**±1.37	34.12±6.64	51.11±5.89	44.20±3.55	56.16±2.15	43.35
SiTNER	29.53±2.96	39.92±6.82	**46.53**±6.57	**55.44**±2.80	**45.39**±2.97	**57.53**±1.13	**45.72**

Table 2: F_1 performance on GENIA, GermEval, and NEREL datasets with 1-shot and 5-shot settings (%).

dataset was from other languages. For the GENIA dataset, we dropped four entity types with several entities less than 50, thus the number of total types is 32. For the NEREL and GermEval datasets, the sampled datasets are from the given test part of the original datasets.

To encode words in different languages into vectors, we used the PLM $\text{BERT}_{\text{base\_multilingual}}$ which has 12 heads of attention layers and 768 word-embedding dimensions. The learning rate is set to 5e-5 and 1e-8 during the training and self-training process, respectively. The temperature τ is set to 10. The ratio of the "[MASK]" token in the pseudo data is 10%. We implemented SiTNER with PyTorch 1.12.1, and the experiments were performed on a Nvidia Tesla A10 GPU.

4.4 Experimental Results

Table 2 shows the average micro F_1 results over ten experiments with different random seeds on three target domain nested NER datasets including GENIA, GermEval, and NEREL. The micro F_1 represents the aggregation performance on all entity types by using the total number of true positives, false positives, and false negatives for all entity types in the calculation of F_1 scores. Compared to baseline models, the SiTNER achieves the best performance for each setting on GermEval and NEREL datasets. For example, compared with SpanProto, the SiTNER achieves an increase of 3.33% and 1.37% on the 5-shot setting on GermEval and NEREL datasets in terms of micro F_1 score, respectively. SiTNER achieves an increase of 12.41% and 1.19% on the 1-shot setting on GermEval and NEREL datasets in terms of micro F_1 score, respectively.

However, for the GENIA dataset, our model did not outperform the best-performing baseline model, whether in the 1-shot or 5-shot settings. This is because our backbone model is the simplest ProtoBERT model. The performance of this module is not sufficient to compete with the best baseline models. Besides, as shown in Table 5, the GENIA dataset comprises a more diverse range of categories, leading to the observation that the performance of Protobert is less effective in discerning high-quality labels compared to the other two datasets. In the case of GENIA, less than half of the pseudo-labels are identified as high-quality labels, while in GERM and NEREL, 90.31% and 73.35% of the pseudo-labels, respectively, are categorized as high-quality labels. This is also a contributing factor to the lower performance of the GENIA dataset compared to the baseline.

5 Analysis

5.1 Effect of Replace Strategy

In Section 3.2 and Figure 4, we explained that we aim to select as many true entities from the query set as possible and place them in sentences from the support set. To validate the effectiveness of this strategy, we designed a comparative experiment: we directly included the original entities from the query set along with their corresponding original sentences in the self-training process (**RSS**), as shown in Table 3.

Table 3 shows that if we directly construct a self-training dataset by including possible entities from the query set along with their corresponding sentences, the results are not as effective as our self-training method, which involves placing these entities into sentences from the support set. We believe this is due to the issue of entity nesting. Even if the predicted entities from the query set are within the decision margin and have higher accuracy, the surrounding nested entities may still be predicted incorrectly. Including these incorrectly predicted entities with their inaccurate pseudo-labels in the

Dataset		Before self-training	SiTNER		SiTNER w **RSS**	
			result	improvement	result	improvement
NEREL	1-shot	44.79	46.09	1.29	45.15	0.36
	5-shot	51.20	54.06	2.86	52.86	1.66

Table 3: F_1 performance obtained using different self-training data generation strategies. We set our SiTNER model to a ratio of 0, meaning that no words are replaced with "[MASK]" in the support set sentences. As a comparison, we directly included the original entities from the query set along with their corresponding original sentences in the self-training process (**RSS**).

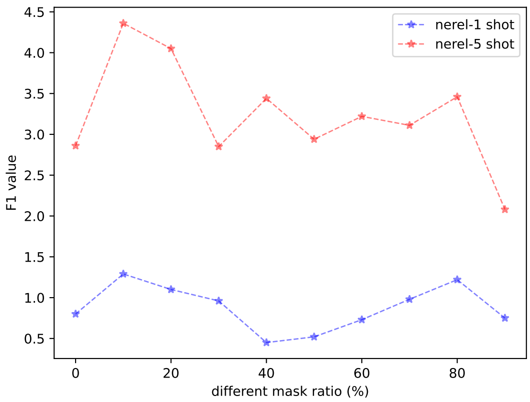

Figure 5: The results of different "[MASK]" ratios in sentences.

self-training dataset can have a detrimental impact on the model. On the other hand, when we place these entities into sentences from the support set to create new sentences, the labels for the nested entities around them are correctly positioned as "O" resulting in greater accuracy.

For instance, in the sentences shown in Figure 4, "Human immunodeficiency" is accurately identified as the "virus" entity. However, due to its nested structure, "Human immunodeficiency virus" can easily be misclassified as the "virus" entity, even though its true label should be "O" . If this sentence is directly used as self-training pseudo data, it would have a detrimental effect on the model. On the contrary, if we include "Human immunodeficiency" in a sentence from the support set and replace "HIV-1" the nested entity "HIV-1 LTR" becomes "Human immunodeficiency LTR" and its label changes from "DNA_domain_or_region" to "O". This way, the impact of misclassified spans on the model would be smaller.

5.2 Effect of MASK tokens

After incorporating possible entities from the query set into sentences from the support set, to further reduce the similarity with the original support set sentences, we randomly replaced different numbers of words in the new sentences with "[MASK]" tokens as mentioned in Section 3.2. To investigate the impact of varying the number of replacements, we designed a comparative experiment, and the results are shown in Figure 5.

We observe that the varying proportions of different masked tokens in sentences have a discernible impact on the experimental performance of F_1 value. In comparison to the 1-shot setting, the 5-shot setting demonstrates a more pronounced effect on the results. Additionally, even without replacing words in the sentence with masks, the influence of self-training contributes to improved outcomes. However, considering the overall perspective, favorable results are achieved when the masked tokens are present in lower quantities (10%) or higher proportions (80%).

6 Related Work

6.1 Rich-resource Nested NER

Nested NER aims to recognize entities with nested structures. Most of the current methods for nested NER are established on rich-resource datasets, and they require a large number of instances for training the model. These methods could be categorized into span-based, hypergraph-based, and layered-based (Wan et al., 2022).

Span-based methods treat sequences of tokens as spans and then label all possible spans by classification models (Shen et al., 2021; Li et al., 2020b; Tan et al., 2021). Hypergraph-based methods analyze the dependence of words in a sentence and then construct a dependency tree (Yu et al., 2020) or other structures (Wang and Lu, 2018; Katiyar and Cardie, 2018) to help identify nested entities.

These methods may be stuck in overfitting due to sophisticated models and the limited number of instances for training in the few-shot setting.

6.2 Few-shot NER

Few-shot NER requires recognizing entities with the support of only very few labeled instances (Hofer et al., 2018; Fritzler et al., 2019). Due to limited information in labeled instances, methods for few-shot NER mainly resort to a rich-resource source domain to help train models, resulting in meta-learning frameworks that train models on adequate subtasks to make the model acquire the learning ability on few-shot tasks (Ma et al., 2022b).

Within the meta-learning framework, various kinds of models are designed. For example, metric-based methods, including ProtoBERT (Snell et al., 2017), NNShot (Yang and Katiyar, 2020), and SpanProto (Wang et al., 2022a), measure distances between prototypes in the support set and instances in the query set. Optimization-based methods, such as MAML (Finn et al., 2017) and FEWNER (Li et al., 2020a), train the model by a special optimizer. And Contrastive-learning methods, such as CONTaiNER (Das et al., 2022), aim to maximize similarities of the same type and minimize similarities between different types.

Besides, prompt-based methods have gained attention due to the ability to guide models focused on the information of interests through various templates (Hou et al., 2022; Hu et al., 2022).

These few-shot NER methods mostly focus on flat entities. Few works have discussed the few-shot nested NER setting. Wang et al. converted sequence labeling to span-level matching and showed their method could handle nested entities (Wang et al., 2022c). However, it is not designed for the few-shot nested NER specifically.

6.3 Semi-supervised Learning

In recent years, there has been a considerable amount of research in the field of semi-supervised learning (Xie et al., 2020b; Berthelot et al., 2019), and a subset of this research involves the utilization of pseudo-labels (Sohn et al., 2020) and self-training (Wang et al., 2023, 2021a). Some of these efforts are focused on applying semi-supervised learning methods to address the issue of class imbalance (Wei et al., 2021; Yang and Xu, 2020; Hyun et al., 2020).

To make full use of unlabeled data in NER tasks, the self-training method could use contextualized augmentations to improve the generalization ability of the NER model (Meng et al., 2021). The combination of transfer learning and self-training strategy shows a boost in performance in low-resource biomedical applications (Gao et al., 2021).

These semi-supervised learning methods neither study N-way K-shot setting scenario of few-shot nested NER tasks.

7 Conclusion

In this work, we propose SiTNER as a novel contrastive and self-training framework for the unexplored few-shot cross-lingual NER tasks. Specifically, diverging from conventional data selection or re-weighting methods, SiTNER effectively harnesses knowledge from unlabeled data within the target domain. SiTNER consists of two primary components: contrastive span classification and masked pseudo-data self-training.

Firstly, we present a contrastive objective tailored for few-shot cross-lingual NER tasks. We extend typical supervised contrastive learning methods by calculating a decision margin for each entity category and generating high-quality pseudo-labels for the unlabeled query set. Secondly, we incorporate these pseudo-labels into sentences within the support set and employ a masking strategy to diminish similarity with the original support set. Experiments on three cross-lingual nested NER datasets validate the effectiveness of SiTNER.

8 Limitations

Given that few-shot nested cross-lingual NER is a nascent task, this paper provides only a preliminary exploration and acknowledges several limitations that warrant further consideration. The foremost concern pertains to the multi-language dimension. Our evaluation of the SiTNER framework relies on English, German, and Russian datasets. Despite the substantial linguistic distinctions among these languages, they share a common lineage within the Indo-European language family. This raises a potential language bias, necessitating an assessment of SiTNER's generalization capability across different language families.

The second limitation revolves around the imbalanced distribution of entity types. The stringent K-shot setting proves challenging to uphold, leading to difficulties in achieving a balanced performance across entity types that exhibit notable quantitative disparities. Addressing this challenge remains an ongoing task.

References

Darina Benikova, Chris Biemann, and Marc Reznicek. 2014. Nosta-d named entity annotation for german: Guidelines and dataset. In *Proceedings of the Ninth International Conference on Language Resources* *and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 2524–2531. European Language Resources Association (ELRA).

David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5050–5060.

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

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, 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. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3198–3213. Association for Computational Linguistics.

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

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

Yao Fu, Chuanqi Tan, Mosha Chen, Songfang Huang, and Fei Huang. 2021. Nested named entity recognition with partially-observed treecrfs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12839–12847.

Shang Gao, Olivera Kotevska, Alexandre Sorokine, and J Blair Christian. 2021. A pre-training and self-training approach for biomedical named entity recognition. *PloS one*, 16(2):e0246310.

Maximilian Hofer, Andrey Kormilitzin, Paul Goldberg, and Alejo J. Nevado-Holgado. 2018. Few-shot learning for named entity recognition in medical text. *CoRR*, abs/1811.05468.

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, Dublin, Ireland. Association for Computational Linguistics.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2240, Dublin, Ireland. Association for Computational Linguistics.

Minsung Hyun, Jisoo Jeong, and Nojun Kwak. 2020. Class-imbalanced semi-supervised learning. *CoRR*, abs/2002.06815.

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 (Long Papers)*, pages 861–871, New Orleans, Louisiana. Association for Computational Linguistics.

Hazel Kim, Jaeman Son, and Yo-Sub Han. 2022. LST: lexicon-guided self-training for few-shot text classification. *CoRR*, abs/2202.02566.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus - a semantically annotated corpus for bio-textmining. In *Proceedings of the Eleventh International Conference on Intelligent Systems for Molecular Biology, June 29 - July 3, 2003, Brisbane, Australia*, pages 180–182.

Jing Li, Billy Chiu, Shanshan Feng, and Hao Wang. 2020a. Few-shot named entity recognition via meta-learning. *IEEE Transactions on Knowledge and Data Engineering*.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020b. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Natalia V. 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), Held Online, 1-3September, 2021*, pages 876–885. INCOMA Ltd.

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

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

Yu Meng, Yunyi Zhang, Jiaxin Huang, Xuan Wang, Yu Zhang, Heng Ji, and Jiawei Han. 2021. Distantly-supervised named entity recognition with noise-robust learning and language model augmented self-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 10367–10378. Association for Computational Linguistics.

Yudong Qian and Weiguo Zheng. 2022. A self-training approach for few-shot named entity recognition. In *Web and Big Data - 6th International Joint Conference, APWeb-WAIM 2022, Nanjing, China, November 25-27, 2022, Proceedings, Part II*, volume 13422 of *Lecture Notes in Computer Science*, pages 183–191. Springer.

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S. Rawat, and Mubarak Shah. 2021. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021. Locate and label: A two-stage identifier for nested named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2782–2794, Online. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4077–4087.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208.

Zeqi Tan, Yongliang Shen, Shuai Zhang, Weiming Lu, and Yueting Zhuang. 2021. A sequence-to-set network for nested named entity recognition. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3936–3942. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Austin Cheng-Yun Tsai, Sheng-Ya Lin, and Li-Chen Fu. 2022. Contrast-enhanced semi-supervised text classification with few labels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11394–11402.

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, Brussels, Belgium. Association for Computational Linguistics.

Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2495–2504. Computer Vision Foundation / IEEE.

Jianing Wang, Chengyu Wang, Jun Huang, Ming Gao, and Aoying Zhou. 2023. Uncertainty-aware self-training for low-resource neural sequence labeling. *CoRR*, abs/2302.08659.

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, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3466–3476. Association for Computational Linguistics.

Kuo Wang, Yuxiang Nie, Chaowei Fang, Chengzhi Han, Xuewen Wu, Xiaohui Wang, Liang Lin, Fan Zhou, and Guanbin Li. 2022b. Double-check soft teacher for semi-supervised object detection. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

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

Yaqing Wang, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah. 2021a. Meta self-training for few-shot neural sequence labeling. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1737–1747.

Yiran Wang, Hiroyuki Shindo, Yuji Matsumoto, and Taro Watanabe. 2021b. Nested named entity recognition via explicitly excluding the influence of the best path. 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 3547–3557, Online. Association for Computational Linguistics.

Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. 2021. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10857–10866.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020a. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268. Curran Associates, Inc.

Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le. 2020b. Self-training with noisy student improves imagenet classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10684–10695. Computer Vision Foundation / IEEE.

Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. 2022. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*.

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 2020, Online, November 16-20, 2020*, pages 6365–6375. Association for Computational Linguistics.

Yuzhe Yang and Zhi Xu. 2020. Rethinking the value of labels for improving class-imbalanced learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

A Effect of Decision Margin

Table 4 and 5 illustrate the impact of employing decision margin on the classification results of the backbone model. **total spans** denotes the number of total spans predicted during the inference step among three datasets. "✓, O" denotes the number of spans that the true label is O and the inference label is O. "✓, E" denotes the number of spans that the true label is the entity type and the inference label is the same entity type. "✗, O→E" denotes the number of spans that the true label is O but the inference label is the entity type. "✗, E→O" denotes the number of spans that the true label is the entity type but the inference label is O. "✗, E→oE" denotes the number of spans that the true label is the entity type but the inference label is a different entity type.

Table 4 presents the impact of incorporating decision margin on the final prediction outcomes of our backbone models across three datasets. The use of decision margin leads to an increase in the number of O→O cases and a decrease in E→E cases, where some true entity-labeled data points fall outside the decision margin and are misclassified as O. Consequently, the overall predictive performance of the model decreases compared to the scenario where decision margin are not employed. Additionally, concerning misclassifications, the model tends to reduce the instances classified as E (entity) and increase those classified as O.

Table 5 provides a breakdown of the components within segments classified as entities by the backbone model, comparing the proportions with and without the use of decision margin. It can be observed that although the number of segments classified as entities decreases when decision margins are employed, the proportion of correctly classified segments among these entity segments increases. Therefore, utilizing these correctly classified entity segments to augment the training data for the few-shot support set ensures the quality of the added data.

total spans	GENIA (32-way) 5119635		GERM (12-way) 617650		NEREL (29-way) 247149	
decision margin?	✗	✓	✗	✓	✗	✓
✓, O	4953007	5006369	608832	610871	238578	241029
✓, E	45674/27.41%	23868/21.07%	4174/47.33%	2415/35.62%	3457/40.33%	2051/33.51%
✗, O→E	74324/44.60%	20962/18.50%	2224/25.22%	185/2.72%	3083/35.97%	632/10.32%
✗, E→O	12717/7.63%	62682/55.34%	1073/12.16%	4105/60.55%	1302/15.19%	3324/54.31%
✗, E→oE	33913/20.35%	5754/5.08%	1347/15.27%	74/1.09%	729/8.50%	113/1.84%
F_1	37.10	33.41	58.22	52.11	54.20	49.52

Table 4: Statistical results by using decision margin or not.

	GENIA (32-way)		GERM (12-way)		NEREL (29-way)	
decision margin?	✗	✓	✗	✓	✗	✓
entity spans	153911	50584	7745	2674	7296	2796
✓, E	45674/29.67%	23868/47.18%	4174/53.89%	2145/90.31%	3457/47.55%	2051/73.35%
✗, O→E	74324/48.29%	20962/41.43%	2224/28.71%	185/6.91%	3083/42.41%	632/22.60%
✗, E→oE	33913/22.03%	5754/11.375%	1347/17.39%	74/2.76%	729/10.02%	113/4.04%

Table 5: Statistical results of the spans which are prediceted as an entity by using decision margin or not.

B Baseline Models

We compare our SiTNER with the following baseline models:

- NER-DP (Yu et al., 2020) is a rich-resource-based nested NER method. It uses the idea of graph-based dependency parsing and applies a biaffine model to establish the dependency of the start and end words for each span. For the few-shot nested NER task, we train the model via the support set on the target domain.

- TIdentifier (Shen et al., 2021) is also a rich-resource-based nested NER method. It utilizes a Two-stage Identifier (TIdentifier) to identify nested entities. It first locates entities by seed spans through a seed span generation module and then classifies them by a span proposal module. We also train it via the support set on the target domain.

- IoBP (Wang et al., 2021b) is an extension of the second-best path recognition method, which eliminates the impact of the best path. It is a layered approach that maintains a set of hidden states at each time step and employs them to construct a unique potential function for recognition at each level.

- PO-TreeCRFs (Fu et al., 2021) treats nested NER as constituency parsing with partially observed trees. It proposes a model called partially observed TreeCRFs to handle this task. Labeled entity spans are considered observed nodes in a constituency tree, while other spans are latent nodes. The TreeCRF model allows for joint modeling of observed and latent nodes. This model supports different inference operations for different nodes, enabling efficient parallelized implementation.

- CONTaiNER (Das et al., 2022) is a contrastive-learning-based few-shot flat NER method. It assumes the word embeddings follow the Gaussian distributions and uses KL-divergence to measure the similarity between words. It applies a contrastive loss function of the average of similarities between positive samples dividing similarities between all samples. We adapt this method to handle the nested NER task by applying the entity span formulation.

- ProtoBERT (Snell et al., 2017) is a metric-learning-based few-shot flat NER method. It identifies the prototype for each entity type and makes inferences according to the distances between prototypes and query samples. It applies the cross-entropy loss to optimize the model. We also adapt it with the entity span formulation.

- NNShot (Yang and Katiyar, 2020) is also a metric-learning-based method for the few-shot flat NER. It makes inferences according to the word-level distance from the labeled support set. We adapt it to handle nested entities by utilizing entity spans rather than sequence labeling, therefore, the CRF (Conditional Random Field) layer is not needed to label the words. Consequently, our experiment did not use the StructShot method mentioned by Yang et al. (Yang and Katiyar, 2020).

- ESD (Wang et al., 2022c) is a metric-learning-based few-shot flat NER method that constructs prototypes by applying intra-span and cross-span attention to enhance span representation. Based on enhanced representations, it classifies spans according to the prototypes from the support set. The authors showed this method could handle nested entities due to the entity span formulation. We apply it directly in our experiment.

- SpanProto (Wang et al., 2022a) is also a metric-learning-based method designed for the few-shot flat NER scenario. It applies a two-stage strategy to recognize entities, including a span extractor stage to determine candidate entity spans and a mention classifier stage to identify entity labels. This method applies the entity span formulation and could handle nested entities, although the authors do not validate it. We also apply it directly in our experiment.