Focusing, Bridging and Prompting for Few-shot Nested Named Entity Recognition

Yuanyuan Xu[1] **Zeng Yang**[1] **Linhai Zhang**[1] **Deyu Zhou**[1*] **Tiandeng Wu**[2] **Rong Zhou**[2]
[1]School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, China
[2]Huawei Technologies Co., Ltd., China
{yuanyuan-xu,yangzeng,lzhang472,d.zhou}@seu.edu.cn
{wutiandeng1,joe.zhourong}@huawei.com

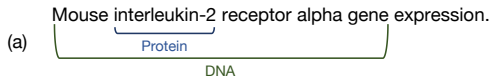
Abstract

Few-shot named entity recognition (NER), identifying named entities with a small number of labeled data, has attracted much attention. Frequently, entities are nested within each other. However, most of the existing work on few-shot NER addresses flat entities instead of nested entities. To tackle nested NER in a few-shot setting, it is crucial to utilize the limited labeled data to mine unique features of nested entities, such as the relationship between inner and outer entities and contextual position information. Therefore, in this work, we propose a novel method based on focusing, bridging and prompting for few-shot nested NER without using source domain data. Both focusing and bridging components provide accurate candidate spans for the prompting component. The prompting component leverages the unique features of nested entities to classify spans based on soft prompts and contrastive learning. Experimental results show that the proposed approach achieves state-of-the-art performance consistently on the four benchmark datasets (ACE2004, ACE2005, GENIA and KBP2017) and outperforms several competing baseline models on F1-score by 9.33% on ACE2004, 6.17% on ACE2005, 9.40% on GENIA and 5.12% on KBP2017 on the 5-shot setting.

1 Introduction

Named entity recognition (NER), aiming at identifying the spans of text and classifying them into pre-defined entity categories, is a fundamental task in natural language processing (Yan et al., 2021). NER serves as a crucial component for many downstream tasks such as information extraction, sentiment analysis and other NLP applications (Mao and Li, 2021; Peng et al., 2022).

Few-shot NER, focusing on named entity recognition with a small number of labeled data, has attracted much attention. Frequently, entities are nested within each other as shown in Figure 1(a). However, most of the existing work on few-shot NER addresses flat entities instead of nested entities. Approaches for few-shot flat NER can mainly be divided into three categories: sequence-labeling-based, generative-based and span-based methods. Sequence-labeling-based mehtods treat NER as sequence labeling that assigns a tag for each token using the BIO or IO tagging scheme (Ma et al., 2022b; Huang et al., 2022b; Das et al., 2022). Generative-based methods autoregressively generate the entity types or the pointer index sequence directly (Cui et al., 2021; Hou et al., 2022; Chen et al., 2022). Span-based methods enumerate text spans in the input text and classify each span based on its corresponding template score (Yang et al., 2022), or the similarity between the span representation and the anchor (Wang et al., 2022a; Ma et al., 2022c; Wang et al., 2022b; Ji et al., 2022).



(b)

Tags	DNA	RNA	cell_type	cell_line
nested ratio	69.17%	15.33%	5.73%	9.77%

Figure 1: (a) An example sentence marked with nested entities in GENIA. (b) The percentages of the entities of Protein being nested with the entities of other categories in GENIA.

Directly applying current few-shot flat NER methods to nested named entities suffers from some weaknesses. For sequence-labeling-based methods, extra strategies such as layering and concatenating the nested entity's multiple labels into one label (Straková et al., 2019; Wang et al., 2020) are needed. Such adaptation lacks flexibility and makes the already scarce supervision signal even more sparse. Generative-based methods can directly handle nested entities. However, due to the auto-regressive generation manner, the optimiza-

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 2621–2637
July 9-14, 2023

tion objective is not consistent with the NER task, resulting in some biases learned by the model during the training process (Zhang et al., 2022). In addition, such biases are more difficult to eliminate with limited labeled data.

By enumerating all the text spans, span-based nested NER can be converted into flat NER, which seems promising. However, such adaptation faces two challenges. First, it is crucial to utilize the relationship between inner and outer entities in nested NER, which is usually ignored in the previous work. Some types of entities in medical-related datasets are prone to be nested. As shown in Figure 1 (b), in GENIA, the frequencies of the entities of Protein type being nested with the entities of DNA type are nearly five times higher than that with the entities of RNA type. Secondly, the same mention may have different types in polysemy scenarios. Therefore, it is necessary to capture local features and precisely model contextual information.

To address the issues mentioned above, we propose a novel span-based method based on Focusing, brIdging and prompTing (FIT) for few-shot nested NER without using source domain data. In the focusing stage, inspired by the IO tagging scheme of sequence-labeling-based methods, each token is tagged whether a part of an entity or not. Then entity-concentrated parts can be obtained by concatenating **continuous** tokens marked with the I-tag. In the bridging stage, for each entity-concentrated part, all spans obtained by enumerating are chosen as candidate spans and filtered according to the boundary score of each candidate span. The bridging stage acts as a bridge connecting the flat entity-concentrated parts with nested NER since nested entities can be obtained by enumerating. In the prompting stage, to make use of the relationship information between nested entities and contextual position information, adversarial prompt-based span classification is proposed. The soft prompts directly before and after the span are inserted to make full use of the contextual position information near the span for classification. Moreover, contrastive learning is employed to shorten the distance between sentence representations to reduce the interference caused by soft prompts. In this way, we preserve the potential connections between nested entities.

Our main contributions are as follows:

- A novel span-based method based on Focusing, brIdging and prompTing (FIT) for few-shot nested NER is proposed. To the best of our knowledge, we are the first to tackle few-shot nested NER without using source domain data.

- To make use of the relationship information between nested entities and contextual position information, adversarial prompt-based span classification is proposed.

- Experimental results show that FIT achieves state-of-the-art performance consistently on the four benchmark datasets (ACE2004, ACE2005, GENIA and KBP2017) and outperforms several competing baseline models on F1-score by 9.33% on ACE2004, 6.17% on ACE2005, 9.40% on GENIA and 5.12% on KBP2017 on 5-shot setting.

2 Related Work

2.1 Nested NER

Most of the existing nested NER methods focus on the fully supervised learning paradigm. There are sequence-labeling-based methods (Straková et al., 2019; Wang et al., 2020), generative-based methods (Yan et al., 2021; Tan et al., 2021), span-based methods (Shen et al., 2021; Yuan et al., 2022; Huang et al., 2022a), anchor-based methods Lin et al. (2019) and so on. There are also methods based on hyper-graph, which adopt the hyper-graph to represent all possible nested structures in a sentence (Katiyar and Cardie, 2018; Wang and Lu, 2018). However, these supervised nested NER methods rely on plenty of labeled data to work, which is not suitable for the few-shot setting.

2.2 Few-shot NER

In recent years, several methods have been proposed to solve the few-shot flat NER task, mainly including sequence-labeling-based (Huang et al., 2021; Ma et al., 2022b,a; Yang and Katiyar, 2020; Das et al., 2022; Huang et al., 2022b), generative-based (Cui et al., 2021; Hou et al., 2022; Chen et al., 2022) and span-based (Yang et al., 2022; Wang et al., 2022b) methods. In terms of different definitions of few-shot setting, few-shot NER can also be divided into two categories: in-domain (Huang et al., 2022b) and domain transfer (Das et al., 2022) settings. The former directly uses few samples for training and tests on the complete test set; while the latter pre-trains on the rich-resource source domain

dataset and then fine-tunes on a low-resource target domain dataset. To the best of our knowledge, there is only one work dedicated to studying the few-shot nested NER (Ming et al., 2022). For each word, they design a Biaffine representation module for learning the contextual dependency representation, and then merge semantic representation by the residual module. However, they apply max pooling to extract the most important features as span representation, which loses a lot of span information. Moreover, we focus on the in-domain setting, a more difficult scenario, instead of the domain transfer setting. Our approach can be easily adapted to the domain-transfer setting by using the pre-training and fine-tuning paradigm.

3 Method

In this section, we will first introduce the task definition of nested NER, then describe the details of FIT. Finally, the training objective is introduced.

3.1 Overall Architecture

Given an input sentence $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ of n tokens, nested NER aims to correctly identify the left and right boundary tokens x_{e_l} and x_{e_r} for every entity $\boldsymbol{e} = \{x_{e_l}, \ldots, x_{e_r}\}$ in $\boldsymbol{x}$, and assign $\boldsymbol{e}$ the correct entity type y from a predefined list of categories $\mathcal{Y}$, e.g., $\mathcal{Y} = \{$"GPE", "ORG", $\ldots\}$. Unlike flat NER, there will be overlapping between entities and the tokens in entity $\boldsymbol{e}$ may be assigned multiple types in nested NER.

We formalize nested NER as span extraction and span classification which further are divided into three subtasks. Figure 2 illustrates how the proposed approach, FIT, works. In the focusing stage, the entity-concentrated parts, such as "state legislatures" shown in Figure 2, are obtained. In the bridging stage, span extraction is conducted on the parts obtained in the focusing stage. Spans such as "representatives to the electoral college" are collected. In the prompting stage, spans obtained in the bridging stage are classified.

3.2 Focusing

Given an input text $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ consisting of n tokens, the focusing stage is to find the entity-concentrated parts in $\boldsymbol{x}$, i.e., all the longest parts where named entities are adjacent as shown in Figure 2, which is important for the following bridging stage. We denote the set of entity-concentrated parts as $\boldsymbol{x_r} = \{\boldsymbol{x_{r_1}}, \ldots, \boldsymbol{x_{r_K}}\}$, where $\boldsymbol{x_{r_i}} \cap \boldsymbol{x_{r_k}} = \emptyset$, $\boldsymbol{x_{r_k}} = \{x_l, \ldots, x_r\} \subset \boldsymbol{x}$ denotes k-th part, and x_l, x_r denote the left and right boundary tokens respectively.

The focusing stage is accomplished by constructing an IO tagging module and predicting each token whether a part of an entity or not based on its tag score. Each entity-concentrated part $\boldsymbol{x_{r_k}}$ can be obtained by concatenating **continuous** tokens marked with I-tag.

The implementation details are as follows. First, we feed the input text into BERT to obtain the representation $h \in \mathbb{R}^{n \times d}$, where d is the dimension of the BERT hidden states. For each token x_i, BERT tokenizer may divide it into multiple subtokens $\boldsymbol{t}_i = (t_{i1}, \ldots, t_{ij})$. Consequently, the representation h_i^{tag} of each token x_i is the concatenation of the mean pooled subtoken representation h_i^p and the representation of the [CLS] token $h^{[\mathrm{CLS}]}$. The tag score p_i^{tag} is calculated as follows:

$$h_i^p = \mathrm{MeanPooling}(h_{t_{i1}}, \ldots, h_{t_{ij}}) \tag{1}$$

$$h_i^{tag} = \mathrm{Concat}(h_i^p, h^{[\mathrm{CLS}]}) \tag{2}$$

$$p_i^{tag} = \mathrm{Softmax}(\mathrm{MLP}_{\mathrm{tag}}(h_i^{tag})) \tag{3}$$

where MLP denotes the multilayer perceptron for binary classification. Then whether a token is a part of an entity can be calculated as:

$$\hat{y}_i^{tag} = \arg\max(p_i^{tag}) \tag{4}$$

For the binary classifier, we simply use the cross-entropy loss:

$$\mathcal{L}_{focus} = \sum_i \mathrm{CrossEntropyLoss}(p_i^{tag}, y_i^{tag}) \tag{5}$$

where y_i^{tag} is the ground truth label; y_i^{tag} being 1 denotes x_i is part of an entity and 0 denotes that x_i is not part of an entity.

3.3 Bridging

In the bridging stage, for each entity-concentrated part $\boldsymbol{x_{r_k}}$ obtained in the focusing stage, we enumerate all spans in $\boldsymbol{x_{r_k}}$ to obtain candidate nested spans. Candidate nested spans are filtered according to the boundary scores to reduce spans with low-quality.

To calculate the boundary score of each candidate nested span, we need to calculate the probabilities of each token $x_i \in \boldsymbol{x_{r_k}}$ being the left or right boundary of an entity respectively.

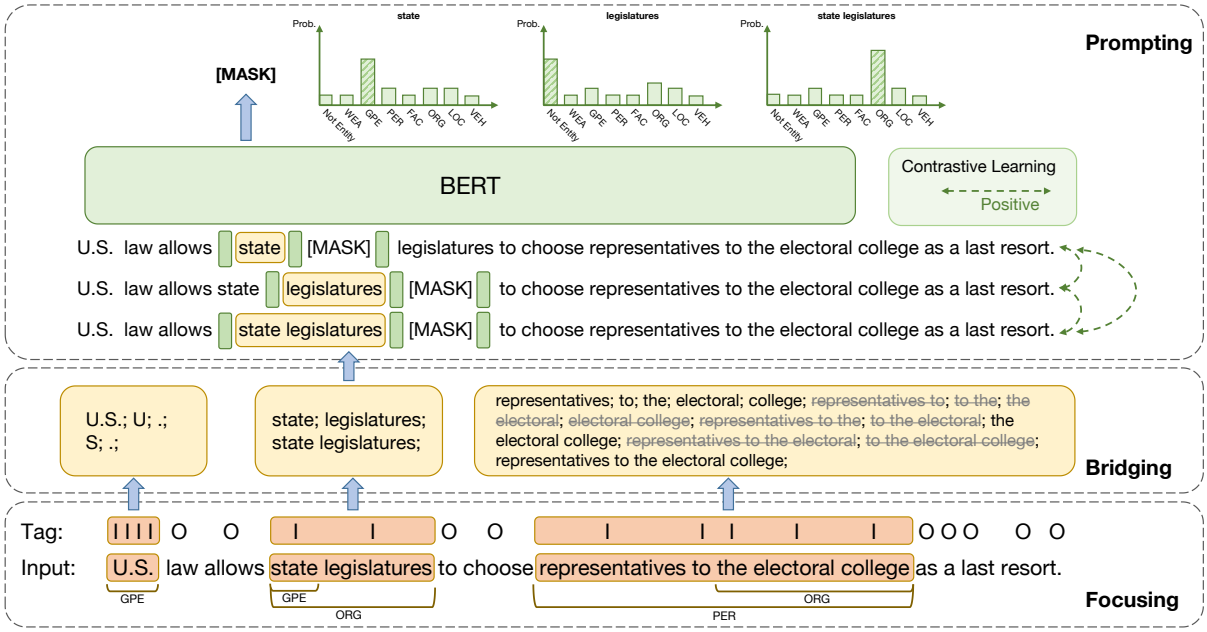


Figure 2: The architecture of the proposed approach, FIT.

For each entity-concentrated part $\boldsymbol{x_{r_k}}$, its part-representation h_i^r is the mean pooling of all tokens' representations in $\boldsymbol{x_{r_k}}$. We concatenate the part-representation h_i^r and the token representation h_i^p to obtain the representation $h_i^{boundary}$ for each token $x_i \in \boldsymbol{x_{r_k}}$, which is used to calculate whether token x_i is the left or right boundary of an entity. The probabilities of each token x_i being the left and right boundaries can be calculated as follows:

$$h_i^r = \text{MeanPooling}(h_{x_l}, \ldots, h_{x_r}) \quad (6)$$

$$h_i^{boundary} = \text{Concat}(h_i^r, h_i^p) \quad (7)$$

$$p_i^{left} = \text{Softmax}(\text{MLP}_{\text{left}}(h_i^{boundary})) \quad (8)$$

$$p_i^{right} = \text{Softmax}(\text{MLP}_{\text{right}}(h_i^{boundary})) \quad (9)$$

To train the MLP_{left} and $\text{MLP}_{\text{right}}$ classifiers, we need to pre-assign the categories y_i^{left} and y_i^{right} of x_i. 1 denotes that x_i is the left or right boundary of an entity while 0 denotes that x_i is not the boundary of an entity. We simply use the cross-entropy loss:

$$\mathcal{L}_{left} = \sum_i \text{CrossEntropyLoss}(p_i^{left}, y_i^{left}) \quad (10)$$

$$\mathcal{L}_{right} = \sum_i \text{CrossEntropyLoss}(p_i^{right}, y_i^{right}) \quad (11)$$

We denote the set of candidate nested spans obtained by enumerating $\boldsymbol{x_{r_k}}$ as $\hat{\mathbf{s}} = (\boldsymbol{s_1}, \ldots, \boldsymbol{s_w})$, where $\boldsymbol{s_i} = (s_{l_i}, \ldots, s_{r_i})$ denotes i-th candidate nested span, and s_{l_i}, s_{r_i} denote the left and right boundary tokens of the span respectively. Then the boundary score of each candidate nested span $\boldsymbol{s_i}$ can be calculated as:

$$p_{\boldsymbol{s_i}}^{span} = p_{s_{l_i}}^{left} \odot p_{s_{r_i}}^{right} \quad (12)$$

where $p_{s_{l_i}}^{left}$ denotes the probability of the left boundary token s_{l_i} of the span $\boldsymbol{s_i}$ being the left boundary of an entity. Likewise, $p_{s_{r_i}}^{right}$ denotes the probability of the right boundary token s_{r_i} of the span $\boldsymbol{s_i}$ being the right boundary of an entity. Note that $\odot$ is element-wise multiplication.

Now we sort the set of candidate nested spans $\hat{\mathbf{s}}$ according to the score $p_{\boldsymbol{s_i}}^{span}$. For candidate nested spans with partial overlapping, those with low scores are discarded. For simplification, we denote the set of filtered candidate nested spans as $\mathbf{S} = (\boldsymbol{s_1}, \ldots, \boldsymbol{s_f})$, where $\mathbf{S} \subset \hat{\mathbf{s}}$.

3.4 Prompting

Let $\mathcal{M}$ be a language model pre-trained on large-scale corpora, prompt learning formalizes the classification task into a masked language modeling problem. Specifically, prompt learning wraps the input text with a *template*, a piece of natural language text or some marks. The model $\mathcal{M}$ should predict the label in [MASK] position. In this work, the prompting stage follows the common prompt-learning practice (Schick and Schütze, 2021). In ad-

dition, we introduce contrastive learning to achieve adversarial prompt learning. Due to the space limitation, instead of introducing the overall process, we only list some key parts in this subsection.

Soft Prompts Setting. The first key part is how to construct soft prompts. Wikipedia usually uses the $fullname(abbreviation)$ pattern when introducing entities. For example, when introducing the "OSI model" in Wikipedia, the first sentence in the first paragraph is "The Open Systems Interconnection model (OSI model) is a conceptual model ..." [1]. Inspired by that, we build soft prompts using the same pattern $entity(tag)$, making its form closer to the form of sentences in the pre-training corpus. Specifically, for each span $\boldsymbol{s_i}$ in the filtered candidate nested spans set $\mathbf{S}$, we wrap it into

$$\boldsymbol{x}_{\mathrm{p}} = \{x_{\mathrm{part}_1}, [\mathrm{p}_1], \boldsymbol{s_i}, [\mathrm{p}_2], [\mathrm{MASK}], [\mathrm{p}_3], x_{\mathrm{part}_2}\}$$

where $[\mathrm{p_i}]$ denotes the soft prompt. For example, assuming we need to classify the span "state" in the sentence $\boldsymbol{x}$ in Figure 2, we wrap it into $\boldsymbol{x}_{\mathrm{p}}$ = "U.S. law allows $[\mathrm{p}_1]$ state $[\mathrm{p}_2][\mathrm{MASK}][\mathrm{p}_3]$ legislatures to choose representatives to the electoral college as a last resort."

Then $\mathcal{M}$ predicts the probability of each label y being filled in [MASK] token $P_{\mathcal{M}}([\mathrm{MASK}] = y \mid \boldsymbol{x}_{\mathrm{p}})$.

The predicted $\hat{y}$ is

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} P_{\mathcal{M}}([\mathrm{MASK}] = y \mid \boldsymbol{x}_{\mathrm{p}}) \quad (13)$$

This objective function is suitable for optimization by applying a cross-entropy loss on the predicted probability.

Contrastive Learning Setting. As the construction of soft prompts will interfere with nested entities, the connection between inner and outer nested entities may be cut off. To alleviate this problem, we introduce contrastive learning. Inspired by (Chen and He, 2021; Sevegnani et al., 2022), we abandon the practice of negative pairs used in traditional contrastive learning and only construct positive pairs. Positive pairs are defined as $(\boldsymbol{x}_{\mathrm{p}_1}, \boldsymbol{x}_{\mathrm{p}_2})$, where both $\boldsymbol{x}_{\mathrm{p}_1}$ and $\boldsymbol{x}_{\mathrm{p}_2}$ are different wrapped spans obtained from $\mathbf{S}$. Note that the spans in the set $\mathbf{S}$ are all paired in pairs.

Then, we calculate cosine embedding loss by:

$$\mathcal{L}_{contrast}(\boldsymbol{x}_{\mathrm{p}_1}, \boldsymbol{x}_{\mathrm{p}_2}) = 1 - \cos(\boldsymbol{x}_{\mathrm{p}_1}^{[\mathrm{CLS}]}, \boldsymbol{x}_{\mathrm{p}_2}^{[\mathrm{CLS}]}) \quad (14)$$

where $\boldsymbol{x}_{\mathrm{p_i}}^{[\mathrm{CLS}]}$ is the [CLS] token representation of $\boldsymbol{x}_{\mathrm{p_i}}$ obtained by BERT.

[1] https://en.wikipedia.org/wiki/OSI_model

3.5 Training Objectives

The overall loss function is:

$$\mathcal{L} = \alpha\mathcal{L}_{focus} + \beta\mathcal{L}_{left} + \gamma\mathcal{L}_{right} + \eta\mathcal{L}_{prompt} + \lambda\mathcal{L}_{contrast} \quad (15)$$

where $\mathcal{L}_{focus}$, $\mathcal{L}_{left}$, $\mathcal{L}_{right}$, $\mathcal{L}_{prompt}$ and $\mathcal{L}_{contrast}$ are balanced with hyper-parameters α, β, γ, η and λ respectively, and $\mathcal{L}_{prompt}$ denotes loss function used in the soft prompt-learning.

4 Experiments

In this section, we conduct experiments on four nested NER datasets to evaluate the effectiveness of the proposed method.

4.1 Datasets

Experiments are conducted on four nested NER datasets: ACE2004[2] (Doddington et al., 2004), ACE2005[3] (Walker et al., 2005), GENIA[4] (Ohta et al., 2002) and KBP2017[5] (Ji et al., 2017). Please refer to Appendix A.1 for the introduction and statistical information about the datasets.

4.2 Experiment Settings

In-Domain Setting. For few-shot learning, we conduct 5, 10, and 20-shot experiments without pre-training on the rich-resource source domain. For a k-shot experiment, all the original test sets are preserved for testing, and the training and development sets are resampled for training. Following the same sampling method as previous work (Ma et al., 2022b), we sample k instances per class from the original training set to form the few-shot training set and sample another k instances per class from the original development set to form the few-shot development set. It is worth noting that no random seed is searched when sampling. 10 sets of data were sampled for k-shot, and all subsequent metrics were taken from the **average** of these 10 sets of data. The statistical information of few-shot datasets obtained by sampling can be found in Appendix A.1. For all datasets, we train our model for 35 epochs and choose the checkpoint with the best validation performance to test. See Appendix A.2 for more detailed settings.

[2] https://catalog.ldc.upenn.edu/LDC2005T09
[3] https://catalog.ldc.upenn.edu/LDC2006T06
[4] http://www.geniaproject.org/genia-corpus
[5] https://catalog.ldc.upenn.edu/LDC2019T12

Evaluation Metrics Setting. Span-level precision, recall, and Micro-F_1 scores are used to measure the results in all experiments. Note that the nested NER datasets also contain a certain proportion of flat entities, then the standard metrics end up confusing flat and nested results and, consequently, are not able to reflect well the ability of a model to detect nesting. To alleviate this issue, we analyze the error rates for total entities e_{total}, flat entities e_{flat}, nested entities e_{nested}, the inner entities e_{inner} and the outer entities e_{outer}. See Appendix A.3 for the calculation formulae.

4.3 Baselines

We use the following models as baselines for few-shot nested NER: Locate and Label (Shen et al., 2021), Unified Generative NER (Yan et al., 2021), SEE-Few (Yang et al., 2022), SDNet (Chen et al., 2022) and ESD (Wang et al., 2022b). The first two baselines are fully supervised methods, and the last three are designed for the few-shot setting. It should be noted that since most few-shot NER methods cannot handle few-shot nested NER, the methods available to us are limited. Please refer to Appendix A.4 for detailed information.

4.4 Experiment Results

Main Results. Table 1 illustrates the performance of FIT and baselines on ACE2004, ACE2005, GENIA and KBP2017. We can see that: 1) FIT consistently outperforms all the baselines by a large margin. Especially in the 5-shot setting, the F1-scores of our model advance previous models by +9.33%, +6.17%, +9.40%, +5.12% on ACE2004, ACE2005, GENIA, and KBP2017 respectively. In the ablation study, we will investigate which components bring improvement. 2) For fully supervised methods, both Locate and Label and Unified Generative NER perform poorly. In particular, Unified Generative NER, as a generative-based method, performs more poorly in a few-shot setting. These show that fully supervised methods may inherently flaw in few-shot NER. 3) For few-shot methods, they show competitive performances as the shot rises, especially SEE-Few and ESD. SEE-Few shows competitive performances under the 20-shot setting, but its performance on the 5-shot setting is not satisfactory. The reason may be the NLI task used in SEE-Few has limitations in context utilization. ESD also shows good performance, which we attribute to its pre-training on the large-scale corpus Few-NERD (Ding et al., 2021) and a significant part of the GENIA dataset. ESD without pre-training has also been evaluated, and its performance decreases by 15%-25% on the four datasets. The performance of 1-shot experiments can be found in Appendix B.1.

Error Rates for Nested Entities. Table 2 illustrates the error rates on the GENIA dataset under few-shot settings. We can see that: Among all methods, FIT significantly reduces the error rates of nested entities. In particular, FIT significantly reduces e_{inner} and makes it even lower than e_{outer}, which shows the effectiveness of FIT for inner entities. The error rates on other datasets can be found in Appendix B.2.

4.5 Ablation Study

We conduct ablation experiments on four datasets. The results on the GENIA dataset are shown in Table 3. The results on other datasets can be found in Appendix B.3.

W/o focusing. We directly enumerate all spans in the sentence as candidate spans and filter them in the bridging stage. A significant performance drop in all settings is observed, which indicates that the focusing stage filters out most of the low-quality parts with only one binary classifier.

W/o filtering. The filtering module in the bridging stage is removed directly. The results show that the filtering module has a positive effect under the 5-shot setting. However, as the number of training data increases, the effect of w/o filtering becomes better. We think that is because the prompting stage can acquire a stronger ability to discriminate low-quality spans as the amount of training data increases, while the filtering module is relatively underfitting at this time. Consequently, some true positives are discarded in advance at the bridging stage, which causes performance loss.

W/o contrastive learning. The contrastive learning module is removed directly. The results show that contrastive learning reduces the interference caused by the soft prompt, and made the model more stable, which is reflected in the reduction of the standard deviation.

W/o series prompt setting. Three kinds of experiments are designed: **w/o soft prompt** replaces soft prompts with discrete prompts (The three prompts are "," "(" and ")" respectively); **w/o contextual prompt** does not use context-based prompts, but moves prompts to the end of the sentence (with template "$\boldsymbol{x}$. $\boldsymbol{s_i}$ is [MASK] ."). Note that the appropriate template has been searched ;

Datasets	Methods	5-shot			10-shot			20-shot		
		P	R	$F_1 \uparrow$	P	R	$F_1 \uparrow$	P	R	$F_1 \uparrow$
ACE2004	Locate and Label	51.59	3.93	$7.20_{\pm 3.34}$	65.31	14.12	$22.88_{\pm 6.81}$	67.74	29.45	$41.02_{\pm 2.79}$
	Unified NER	18.18	5.86	$8.87_{\pm 3.47}$	29.59	9.71	$14.19_{\pm 6.23}$	43.84	21.74	$28.73_{\pm 10.83}$
	SEE-Few	50.08	18.69	$26.54_{\pm 6.60}$	57.74	29.70	$38.89_{\pm 4.07}$	63.53	39.91	$48.94_{\pm 2.27}$
	SDNet	61.40	12.45	$20.55_{\pm 4.64}$	65.73	23.81	$34.82_{\pm 4.71}$	67.18	31.52	$42.87_{\pm 2.13}$
	ESD	34.51	13.69	$19.25_{\pm 5.74}$	53.95	35.44	$42.75_{\pm 5.11}$	56.94	48.27	$52.17_{\pm 3.76}$
	FIT(ours)	46.87	29.31	$\mathbf{35.87}_{\pm 4.92}$	51.43	40.18	$\mathbf{44.88}_{\pm 4.82}$	60.14	48.93	$\mathbf{53.92}_{\pm 2.99}$
ACE2005	Locate and Label	50.20	6.55	$11.43_{\pm 6.56}$	57.80	16.52	$25.13_{\pm 9.00}$	65.13	28.69	$39.61_{\pm 6.02}$
	Unified NER	17.08	5.92	$8.72_{\pm 4.42}$	18.19	9.23	$13.17_{\pm 4.01}$	36.10	18.30	$24.26_{\pm 2.59}$
	SEE-Few	49.42	17.69	$25.58_{\pm 6.61}$	55.92	27.45	$36.36_{\pm 6.63}$	61.37	44.19	$51.31_{\pm 2.27}$
	SDNet	57.46	13.81	$22.03_{\pm 6.12}$	61.17	22.08	$32.20_{\pm 4.89}$	65.84	32.03	$43.00_{\pm 3.55}$
	ESD	36.36	28.51	$31.57_{\pm 6.45}$	42.99	35.72	$38.81_{\pm 7.04}$	55.01	46.39	$50.30_{\pm 3.37}$
	FIT(ours)	44.74	33.05	$\mathbf{37.74}_{\pm 5.33}$	46.83	38.85	$\mathbf{42.25}_{\pm 10.65}$	58.02	48.5	$\mathbf{52.71}_{\pm 2.55}$
GENIA	Locate and Label	36.12	10.42	$15.57_{\pm 6.78}$	52.46	23.29	$31.65_{\pm 6.54}$	62.17	41.60	$49.67_{\pm 4.46}$
	Unified NER	13.26	2.85	$4.68_{\pm 2.27}$	17.23	7.88	$10.62_{\pm 5.48}$	30.89	15.87	$20.98_{\pm 3.64}$
	SEE-Few	30.92	14.41	$19.31_{\pm 6.95}$	52.35	29.84	$37.78_{\pm 5.04}$	59.36	45.10	$50.93_{\pm 4.66}$
	SDNet	41.25	11.36	$17.46_{\pm 6.97}$	48.57	12.18	$19.03_{\pm 7.07}$	57.03	23.54	$33.27_{\pm 3.71}$
	ESD	36.44	20.24	$25.03_{\pm 9.88}$	48.86	28.00	$35.23_{\pm 4.96}$	55.49	41.62	$47.22_{\pm 4.36}$
	FIT(ours)	40.72	30.30	$\mathbf{34.43}_{\pm 9.06}$	52.91	39.51	$\mathbf{44.95}_{\pm 3.38}$	57.00	46.81	$\mathbf{51.26}_{\pm 3.96}$
KBP2017	Locate and Label	69.95	9.57	$16.52_{\pm 7.67}$	68.33	17.54	$27.17_{\pm 9.90}$	69.36	36.40	$47.35_{\pm 7.29}$
	Unified NER	21.13	5.47	$8.49_{\pm 7.94}$	27.66	12.08	$16.00_{\pm 8.28}$	35.17	15.62	$21.30_{\pm 8.20}$
	SEE-Few	47.02	15.34	$22.87_{\pm 4.82}$	55.07	27.48	$36.26_{\pm 6.08}$	58.86	41.99	$48.65_{\pm 5.51}$
	SDNet	62.28	12.24	$20.25_{\pm 3.88}$	65.11	21.03	$31.57_{\pm 4.55}$	64.92	33.98	$44.48_{\pm 4.34}$
	ESD	34.27	24.39	$28.38_{\pm 9.02}$	49.13	38.61	$42.99_{\pm 4.20}$	54.64	51.00	$52.54_{\pm 3.76}$
	FIT(ours)	44.68	27.20	$\mathbf{33.50}_{\pm 4.37}$	50.69	39.43	$\mathbf{44.21}_{\pm 4.64}$	56.39	52.70	$\mathbf{54.27}_{\pm 5.07}$

Table 1: Performance comparison of FIT and baselines on four datasets under different shots.

w/o prompt directly abandons the prompt setting and trains a multi-class classifier to classify the candidate nested spans. Experimental results show that context-based soft prompts have a positive effect, while directly training classifiers is less effective, illustrating the importance of utilizing contextual information in few-shot nested NER.

5 Time Complexity

Theoretically, the number of possible spans in a sentence of length N is $\frac{N(N+1)}{2}$. If we classify almost all spans into corresponding categories, it will lead to a high computational cost with $O(N^2)$ time complexity. However, the focusing stage makes the model only focus on the entity-concentrated part, reducing the time complexity. Although in the worst case, the model keeps the whole sentence as an entity-concentrated part, generating $\frac{N(N+1)}{2}$ candidate nested spans. The number of candidate spans is reduced as some partial overlap spans are discarded according to the boundary scores.

We also evaluate the efficiency of FIT. In the 5-shot setting of the ACE2004 dataset, compared with few-shot span-based methods SEE-Few training that takes 159.39s, the FIT takes 122.37s for the same 35 epochs, which leads to approximately 23.23% speedup. In the inference phase, FIT also spends 31.99ms for each sample on average, which is 15.50% faster than other results-competitive methods. Time usage on four datasets can be found in Appendix C.

6 Discussion

The F_1 scores of 10 sets of 20-shot data sampled on the ACE2005 dataset are compared in Figure 3. The horizontal coordinate is sorted in ascending order by the nested ratio (the lower bound is 22.86%, and the upper bound is 42.14%). The nested ratio of each set can be found in Appendix A.1. It shows that under a single data set, the performance of the model is more closely related to the quality of the sampled data rather than the nested ratio. Nevertheless, FIT works better than other methods.

To further explore the effect of the different

Methods	5-shot					10-shot					20-shot				
	$e_{total}\downarrow$	$e_{flat}\downarrow$	$e_{nested}\downarrow$	$e_{inner}\downarrow$	$e_{outer}\downarrow$	$e_{total}\downarrow$	$e_{flat}\downarrow$	$e_{nested}\downarrow$	$e_{inner}\downarrow$	$e_{outer}\downarrow$	$e_{total}\downarrow$	$e_{flat}\downarrow$	$e_{nested}\downarrow$	$e_{inner}\downarrow$	$e_{outer}\downarrow$
SEE-Few	85.58	86.07	83.84	83.73	84.10	70.17	69.11	73.97	74.39	73.58	54.90	52.20	64.62	65.84	63.49
SDNet	88.64	86.09	97.79	98.08	97.59	87.82	85.12	97.50	97.96	97.13	76.46	70.92	96.36	96.84	96.03
ESD	79.76	78.96	82.64	81.27	84.41	72.00	70.59	77.08	74.46	80.05	58.38	55.67	68.11	63.81	72.52
FIT(ours)	**69.70**	**68.11**	**75.40**	**73.82**	**77.50**	**60.49**	**57.91**	**69.77**	**65.81**	**73.99**	**53.19**	**49.89**	**65.04**	**60.07**	**70.17**

Table 2: The error rates comparison of FIT and baselines on the GENIA dataset under different shots. Orange indicates that e_{inner} is smaller than e_{outer}. Note that: 1) We follow Wang et al. (2022b) and pre-train ESD on part of the GENIA dataset. 2) We did not mark SDNet's e_{inner} because the values are too large to be informative.

Methods	5-shot			10-shot			20-shot		
	P	R	$F_1\uparrow$	P	R	$F_1\uparrow$	P	R	$F_1\uparrow$
Full model	40.72	30.30	**34.43**$_{\pm 9.06}$	52.91	39.51	**44.95**$_{\pm 3.38}$	57.00	46.81	**51.26**$_{\pm 3.96}$
-w/o focusing	33.57	9.80	14.21$_{\pm 8.54}$	49.90	13.25	19.57$_{\pm 7.44}$	57.22	14.40	22.74$_{\pm 4.23}$
-w/o filtering	33.62	22.11	26.56$_{\pm 7.97}$	52.79	37.52	43.58$_{\pm 4.85}$	57.47	48.63	**52.42**$_{\pm 4.03}$
-w/o contrastive learning	41.41	24.43	30.17$_{\pm 9.78}$	52.39	38.94	44.23$_{\pm 5.01}$	59.71	44.16	50.47$_{\pm 4.48}$
-w/o soft prompt	43.02	27.00	32.45$_{\pm 6.87}$	49.22	38.16	42.67$_{\pm 4.55}$	59.23	45.28	51.14$_{\pm 3.98}$
-w/o contextual prompt	36.93	21.74	26.94$_{\pm 8.04}$	53.51	31.90	39.63$_{\pm 7.08}$	59.19	44.09	50.35$_{\pm 4.54}$
-w/o prompt	18.05	9.20	10.99$_{\pm 6.02}$	29.81	21.30	23.94$_{\pm 4.80}$	40.61	35.06	37.39$_{\pm 2.68}$

Table 3: Ablation study of FIT and baselines on the GENIA dataset under different shots.

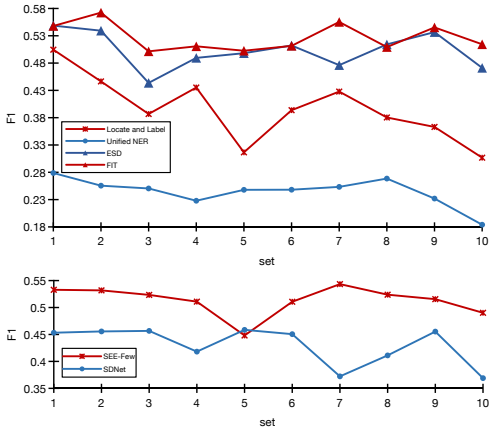


Figure 3: Comparison of F_1 scores of 10 sets of 20-shot data sampled on the ACE2005 dataset.

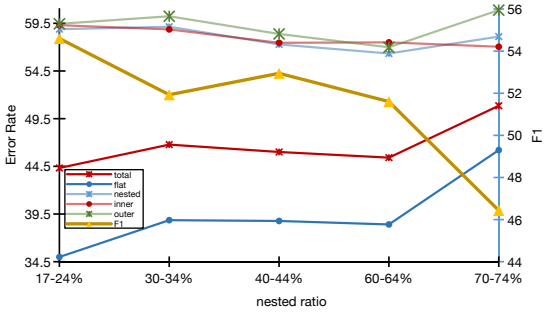


Figure 4: Comparison of F_1 scores and error rates of different nested ratios.

nested ratios of the training sets on FIT, we randomly sample 200 sets of 20-shot data from the ACE2005 dataset and preserve sets that satisfy specific nested ratios. Finally, 50 sets are kept and divided into 5 groups. (Note that each group contains 10 sets and 5 groups corresponding to nested ratios of 17-24%, 20-34%, 40-44%, 60-64%, and 70-74%.) As shown in Figure 4, the F_1 score tends to decrease as the nested ratio increases. However, the e_{nested} maintains a decreasing trend while the e_{flat} increases. It can be seen that the increase in the nested ratio may help the model learn nested entities better, and most of the decrease in F_1 is due to the misjudgment of flat entities.

7 Conclusion

In this work, we propose a span-based method for few-shot nested NER without using source domain data. First, the candidate nested spans are generated by the focusing and bridging components. Then the adversarial prompt-based span classification method is proposed to classify candidate spans into the corresponding categories. Our proposed method, FIT, can make full use of the unique features of nested entities while reducing the computational cost and the impact of low-quality candidate spans. Experimental results show that our method achieves state-of-the-art performance consistently on the four benchmark datasets (ACE2004, ACE2005, GENIA, and KBP2017), and outperforms several competing baseline models on F1-score and the error rates of nested entities.

Limitations

Although our method achieves state-of-the-art performance consistently on the four benchmark datasets, it suffers from the following limitations:

- No optimization for the verbalizer. The verbalizer we use in the prompting stage is just a simple 1-to-1 mapping, This simple design does not fully exploit the capabilities of MLM.
- No explicit modeling of the relationship information between nested entities. We consider that in some other scenarios, the relationship information between nested entities is not very significant. Consequently, explicitly modeling the relationship may introduce new biases. So we just utilize the potential information. But in practice, it is worth exploring how to model such a relationship from a novel perspective.

Acknowledgements

We would like to thank anonymous reviewers for their valuable comments and helpful suggestions and we thank Huawei for supporting this project. This work was funded by the National Natural Science Foundation of China (62176053). This work is supported by the Big Data Computing Center of Southeast University.

References

Jiawei Chen, Qing Liu, Hongyu Lin, Xianpei Han, and Le Sun. 2022. 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, Dublin, Ireland. Association for Computational Linguistics.

Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15750–15758.

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

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

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

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3198–3213, Online. Association for Computational Linguistics.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

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.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Peixin Huang, Xiang Zhao, Minghao Hu, Yang Fang, Xinyi Li, and Weidong Xiao. 2022a. Extract-select: A span selection framework for nested named entity recognition with generative adversarial training. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 85–96, Dublin, Ireland. Association for Computational Linguistics.

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

Bin Ji, Shasha Li, Shaoduo Gan, Jie Yu, Jun Ma, Huijun Liu, and Jing Yang. 2022. Few-shot named entity recognition with entity-level prototypical network enhanced by dispersedly distributed prototypes. In *Proceedings of the 29th International Conference*

on Computational Linguistics, pages 1842–1854, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Heng Ji, Xiaoman Pan, Boliang Zhang, Joel Nothman, James Mayfield, Paul McNamee, Cash Costello, and Sydney Informatics Hub. 2017. Overview of tac-kbp2017 13 languages entity discovery and linking. In *TAC*.

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.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

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, Florence, Italy. Association for Computational Linguistics.

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, Dublin, Ireland. Association for Computational Linguistics.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuanjing 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, Seattle, United States. Association for Computational Linguistics.

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, Dublin, Ireland. Association for Computational Linguistics.

Rui Mao and Xiao Li. 2021. Bridging towers of multi-task learning with a gating mechanism for aspect-based sentiment analysis and sequential metaphor identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13534–13542.

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

Tomoko Ohta, Yuka Tateisi, Jin-Dong Kim, Hideki Mima, and Junichi Tsujii. 2002. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the human language technology conference*, pages 73–77. Citeseer.

Keqin Peng, Chuantao Yin, Wenge Rong, Chenghua Lin, Deyu Zhou, and Zhang Xiong. 2022. Named entity aware transfer learning for biomedical factoid question answering. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19(4):2365–2376.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Karin Sevegnani, Arjun Seshadri, Tian Wang, Anurag Beniwal, Julian McAuley, Alan Lu, and Gérard Medioni. 2022. Contrastive learning for interactive recommendation in fashion. In *SIGIR 2022 Workshop on eCommerce*.

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.

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, Florence, Italy. Association for Computational Linguistics.

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.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2005. Ace 2005 multilingual training corpus-linguistic data consortium. *URL: https://catalog. ldc. upenn. edu/LDC2006T06*.

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.

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. *CoRR*, abs/2210.09049.

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.

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, Seattle, United States. 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, Online. Association for Computational Linguistics.

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, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Zheng Yuan, Chuanqi Tan, Songfang Huang, and Fei Huang. 2022. Fusing heterogeneous factors with triaffine mechanism for nested named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3174–3186, Dublin, Ireland. Association for Computational Linguistics.

Shuai Zhang, Yongliang Shen, Zeqi Tan, Yiquan Wu, and Weiming Lu. 2022. De-bias for generative extraction in unified NER task. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 808–818, Dublin, Ireland. Association for Computational Linguistics.

A Experiment Settings on Nested NER

A.1 Statistics of Nested Datasets

We conduct experiments on four nested NER datasets: ACE2004[6], ACE2005[7], GENIA[8] and KBP2017[9]. GENIA dataset is available under the license of CC-BY 3.0, whereas ACE2004, ACE2005, and KBP2017 require a license from LDC. The details are as follows:

ACE 2004 and ACE 2005 (Doddington et al., 2004; Walker et al., 2005) are two nested datasets, each of them containing 7 entity categories. The two nested datasets also contain more than two layers of nesting and the proportion of long entities is relatively large. Following (Katiyar and Cardie, 2018; Lin et al., 2019; Shen et al., 2021), we split them into the train, dev, and test sets by 8:1:1.

GENIA (Ohta et al., 2002) is a biology nested named entity dataset and contains five entity types, including DNA, RNA, protein, cell line, and cell type categories. We use the original division provided by the official[10], which is nearly 8/1/1 for the train/dev/test set.

KBP2017 (Ji et al., 2017) has 5 entity categories, including GPE, ORG, PER, LOC, and FAC. We randomly split them into train, dev, and test sets by 6:2:2.

In Table 4, We report the number of sentences, the number of sentences containing nested entities, the average sentence length, the total number of entities, the number of nested entities, and the nested ratio on the ACE2004, ACE2005, GENIA, and KBP2017 datasets. In Table 5, We report the nested ratio of our randomly sampled training sets on the ACE2004, ACE2005, GENIA, and KBP2017 datasets.

A.2 Detailed Parameter Settings

We implement FIT with Huggingface Transformers 4.11.3 and PyTorch 1.7.1. In most exper-

[6] https://catalog.ldc.upenn.edu/LDC2005T09
[7] https://catalog.ldc.upenn.edu/LDC2006T06
[8] http://www.geniaproject.org/genia-corpus
[9] https://catalog.ldc.upenn.edu/LDC2019T12
[10] http://www.geniaproject.org/genia-corpus/relation-corpus

Dataset Statistics	ACE2004			ACE2005			GENIA			KBP2017		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
# sentences	6202	745	812	7299	971	1060	15023	1669	1854	2126	722	720
# sent. nested entities	2712	294	388	2799	352	340	3197	325	446	622	208	217
avg sentence length	22.50	23.02	23.05	19.94	19.71	17.90	25.43	24.63	25.99	24.11	25.41	25.10
# total entities	22202	2514	3035	24708	3218	3030	46142	4367	5506	7515	2630	2564
# nested entities	10148	1092	1417	9940	1189	1184	8265	799	1199	2145	725	726
nested ratio (%)	45.71	43.44	46.69	40.23	36.95	39.08	17.91	18.30	21.78	28.54	27.57	28.32

Table 4: Statistics of the four datasets used in the experiments.

Groups	ACE2004			ACE2005			GENIA			KBP2017		
	5-shot	10-shot	20-shot	5-shot	10-shot	20-shot	5-shot	10-shot	20-shot	5-shot	10-shot	20-shot
# 1	40.00	33.33	35.34	22.86	25.71	42.14	8.00	8.00	11.00	8.33	42.86	23.08
# 2	6.45	32.79	34.51	31.43	25.71	26.43	8.00	16.00	6.00	27.27	5.26	34.21
# 3	19.35	33.33	38.14	20.00	41.43	35.71	8.00	26.00	12.00	21.74	28.95	25.97
# 4	19.36	21.31	40.17	34.29	30.00	31.43	0.00	14.00	15.00	34.78	23.26	12.82
# 5	25.81	24.59	40.65	25.71	25.71	28.57	16.00	10.00	12.00	24.00	20.93	21.62
# 6	25.82	20.97	34.45	34.29	54.29	30.71	16.00	20.00	18.00	8.33	24.39	30.14
# 7	29.03	32.79	38.46	45.71	38.57	26.43	16.00	6.00	13.00	12.50	14.29	18.57
# 8	19.35	36.07	33.06	31.43	28.57	22.86	0.00	14.00	7.00	16.00	30.00	27.03
# 9	6.06	29.51	31.30	34.29	40.00	33.57	0.00	12.00	16.00	13.64	24.39	30.14
# 10	38.71	33.85	27.12	11.43	31.43	41.43	16.00	18.00	10.00	9.09	21.74	18.57
avg nested ratio	22.99	29.85	35.32	29.14	34.14	31.93	8.80	14.40	12.00	17.57	23.61	24.22

Table 5: Nested ratio(%) of the few-shot training datasets used in the experiments.

iments, we use BERT (Devlin et al., 2019) as PLM. For the GENIA dataset, we replace BERT with BioBERT (Lee et al., 2019). In the experimental details, we use `bert-base-uncased`[11] for ACE2004, ACE2005 and KBP2017 datasets and `dmis-lab/biobert-base-cased-v1.2`[12] for GENIA dataset (the two model sizes: all about 110M). The soft prompts are initialized by the embedding of "," "(" and ")". The verbalizer is just a simple 1-to-1 mapping as shown in Table 6, that is, only the word corresponding to the semantics of the tag is used as a mapping. We use the Adam Optimizer with a linear warmup-decay learning rate schedule, a dropout before the tag, left boundary and right boundary classifiers with a rate of 0.1. Please see Table 7 for details. We train our model on a single NVIDIA 3090 GPU with 24GB memory.

All baselines follow the settings of their original work. Among them, Locate and Label, SEE-Few, and ESD all uses `bert-base-uncased` for ACE2004, ACE2005 and KBP2017 datasets, and uses `dmis-lab/biobert-base-cased-v1.2` for GENIA dataset. While Unified NER uses `facebook/bart-large`[13] (Lewis et al., 2019) (model size: about 406M), and SDNet uses `t5-base`[14] (Raffel et al., 2020) (model size: about 220M).

Tags	ACE2004	ACE2005	GENIA	KBP2017
# WEA	weapon	weapon	-	-
# GPE	geography	geography	-	geography
# PER	person	person	-	person
# FAC	facility	facility	-	facility
# ORG	organization	organization	-	organization
# LOC	location	location	-	location
# VEH	vehicle	vehicle	-	-
# DNA	-	-	DNA	-
# RNA	-	-	RNA	-
# cell_type	-	-	cell	-
# protein	-	-	protein	-
# cell_line	-	-	group	-
# No Entity	none	none	none	none

Table 6: Verbalizer used in the prompting stage.

P	ACE2004	ACE2005	KBP2017	GENIA
lr	3e-05	3e-05	3e-05	3e-05
Focus&Bridge batch size	1	1	1	1
Prompt batch size	8	8	8	8
α	1.0			
β	1.0			
γ	1.0			
η	1.0			
λ	1.0			
drop out rate	0.1			
lr_warmup	0.1			
weight_decay	0.01			

Table 7: Detailed Parameter(P) Settings

A.3 Error Rates Calculation

We analyze the error rates for total entities e_{total}, flat entities e_{flat}, nested entities e_{nested}, the inner entities e_{inner}, and the outer entities e_{outer}. Specif-

[11] https://huggingface.co/bert-base-uncased

[12] https://huggingface.co/dmis-lab/biobert-base-cased-v1.2

[13] https://huggingface.co/facebook/bart-large

[14] https://huggingface.co/t5-base

ically, we calculate these metrics by dividing the total number of misjudged entities belonging to that entity type by the total number of that entity type. For example, e_{nested} can be calculated by dividing the total number of nested entities misjudged by the total number of nested entities. All the metrics are calculated in the test set. The formulae are as follows:

$$e_{total} = \frac{n_{\text{misjudged\_entities}}}{n_{\text{all\_entities}}} \tag{16}$$

$$e_{flat} = \frac{n_{\text{misjudged\_flat\_entities}}}{n_{\text{all\_flat\_entities}}} \tag{17}$$

$$e_{nested} = \frac{n_{\text{misjudged\_nested\_entities}}}{n_{\text{all\_nested\_entities}}} \tag{18}$$

$$e_{inner} = \frac{n_{\text{misjudged\_inner\_nested\_entities}}}{n_{\text{all\_inner\_nested\_entities}}} \tag{19}$$

$$e_{outer} = \frac{n_{\text{misjudged\_outer\_nested\_entities}}}{n_{\text{all\_outer\_nested\_entities}}} \tag{20}$$

A.4 Baselines

We use the following models as baselines for few-shot nested NER. The first two are models under the fully supervised setting, and the last three are models under the few-shot setting. It should be noted that since most few-shot NER methods cannot handle few-shot nested NER, the methods available to us are limited.

- **Locate and Label** (Shen et al., 2021) generates candidate spans by filtering and boundary regression on the seed spans, and then labels the boundary-adjusted candidate spans with the corresponding categories. The two-stage method achieves good results on fully supervised nested NER.

- **Unified Generative NER** (Yan et al., 2021) formulates the NER task as an entity span sequence generation task, which can directly generate nested entity categories.

- **SEE-Few** (Yang et al., 2022) is a span-based method applied to the few-shot flat NER, which extracts spans with seeding and expanding, then classifies them via natural language inference. It can be naturally extended to few-shot nested NER.

- **SDNet** (Chen et al., 2022) is a self-describing generation model for few-shot NER. In the pre-training stage, the external data is used to jointly train mention describing and entity generation tasks. In the fine-tuning stage, SDNet first conducts mention describing to summarize type concept descriptions, and then conducts entity generation based on the generated descriptions.

- **ESD** (Wang et al., 2022b) formulates the few-shot sequence labeling task as a span-level similarity matching problem between test query and supporting instances to solve few-shot NER. Wang et al. (2022b) mentions that their approach can be extended to few-shot nested NER by modifying pre-training datasets. Specifically, they sample from Few-NERD (Ding et al., 2021) dataset and GENIA dataset in a certain proportion to form the FewNERD-nested dataset, and then pre-trained on it. In our experiments, we control the sampling ratio of the two at 6:4 (FewNERD:GENIA).

B Experiment Results on Nested NER

B.1 1-shot Experiments

We show the performance of 1-shot experiments on ACE2004, ACE2005, and KBP2017 datasets in Table 8. We can see that FIT significantly outperforms all methods.

B.2 Error Rates

We show the error rates on ACE2004, ACE2005, and KBP2017 datasets in Table 9. We can see that FIT significantly reduces the error rates of nested entities among all methods.

B.3 Ablation Studys

We conduct ablation experiments to elucidate the main components of our proposed method FIT. The results on ACE2004, ACE2005 and KBP2017 datasets are shown in Table 10.

C Time Usage

The time usage on ACE2004, ACE2005, and KBP2017 datasets is shown in Table 11.

Datasets	Methods	1-shot		
		P	R	$F_1 \uparrow$
ACE2004	SEE-Few	37.65	2.27	$4.15_{\pm 2.65}$
	SDNet	55.26	7.54	$12.98_{\pm 5.39}$
	ESD	10.70	3.71	$5.24_{\pm 9.62}$
	FIT(ours)	30.21	11.99	$\mathbf{16.67}_{\pm 8.94}$
ACE2005	SEE-Few	38.64	3.01	$5.35_{\pm 5.83}$
	SDNet	48.62	6.21	$10.85_{\pm 4.59}$
	ESD	7.18	2.50	$3.52_{\pm 7.31}$
	FIT(ours)	28.92	9.77	$\mathbf{13.76}_{\pm 7.36}$
GENIA	SEE-Few	14.63	0.97	$1.77_{\pm 1.53}$
	SDNet	32.26	6.53	$10.72_{\pm 3.89}$
	ESD	4.63	3.81	$4.14_{\pm 7.63}$
	FIT(ours)	25.20	14.12	$\mathbf{17.74}_{\pm 6.74}$
KBP2017	SEE-Few	39.45	0.55	$1.08_{\pm 1.28}$
	SDNet	54.65	6.77	$11.90_{\pm 4.15}$
	ESD	10.60	3.39	$5.04_{\pm 9.07}$
	FIT(ours)	32.19	10.43	$\mathbf{15.30}_{\pm 8.52}$

Table 8: Performance comparison of FIT and baselines on four datasets under 1-shot setting.

Datasets	Methods	5-shot					10-shot					20-shot				
		$e_{total} \downarrow$	$e_{flat} \downarrow$	$e_{nested} \downarrow$	$e_{inner} \downarrow$	$e_{outer} \downarrow$	$e_{total} \downarrow$	$e_{flat} \downarrow$	$e_{nested} \downarrow$	$e_{inner} \downarrow$	$e_{outer} \downarrow$	$e_{total} \downarrow$	$e_{flat} \downarrow$	$e_{nested} \downarrow$	$e_{inner} \downarrow$	$e_{outer} \downarrow$
ACE2004	SEE-Few	81.31	77.71	85.42	89.26	83.40	70.30	64.81	76.58	81.73	74.06	60.09	51.91	69.43	75.71	66.18
	SDNet	87.54	77.31	99.24	98.99	99.55	76.19	56.89	98.23	98.03	98.66	68.48	43.05	97.51	97.38	97.96
	ESD	86.31	82.39	90.78	94.44	88.78	64.56	57.89	72.17	76.53	70.41	51.73	42.13	62.68	65.22	62.16
	FIT(ours)	**70.69**	**63.81**	**78.53**	**78.30**	**78.99**	**59.83**	**51.73**	**69.07**	**71.43**	**68.24**	**51.07**	**41.57**	**61.91**	**64.26**	**61.58**
ACE2005	SEE-Few	82.31	78.95	87.55	89.37	86.82	72.55	66.68	81.70	83.84	80.53	55.81	45.86	71.33	76.21	68.64
	SDNet	86.19	78.17	98.71	98.63	98.97	77.92	65.22	97.71	98.00	97.83	67.97	49.61	96.59	97.50	96.39
	ESD	71.50	65.36	81.06	82.47	80.57	64.28	56.84	75.87	77.37	75.25	53.61	45.11	66.86	67.37	67.41
	FIT(ours)	**66.95**	**60.77**	**76.59**	**77.39**	**76.42**	**61.15**	**53.22**	**73.51**	**74.85**	**73.06**	**51.50**	**43.04**	**64.68**	**63.83**	**66.25**
KBP2017	SEE-Few	84.42	83.54	86.67	91.63	81.72	72.35	69.54	79.42	89.25	70.76	57.87	53.31	69.44	79.22	60.32
	SDNet	87.75	83.23	99.19	99.04	99.44	78.88	71.14	98.47	98.32	98.82	65.89	53.43	97.44	97.14	98.06
	ESD	75.43	72.36	83.20	90.73	76.47	61.25	54.91	77.30	87.59	68.53	48.88	41.59	67.36	76.39	59.73
	FIT(ours)	**72.63**	**68.88**	**82.12**	**88.87**	**76.05**	**60.43**	**55.06**	**74.04**	**84.95**	**64.83**	**47.19**	**40.11**	**65.37**	**74.76**	**57.52**

Table 9: The error rates comparison of FIT and baselines on the three datasets under different shots. Orange indicates that e_{inner} is smaller than e_{outer}. Note that: We did not mark SDNet's e_{inner} because the values are too large to be informative.

Datasets	Methods	5-shot			10-shot			20-shot		
		P	R	$F_1 \uparrow$	P	R	$F_1 \uparrow$	P	R	$F_1 \uparrow$
ACE2004	Full model	46.87	29.31	**35.87**$_{\pm 4.92}$	51.43	40.18	**44.88**$_{\pm 4.82}$	60.14	48.93	**53.92**$_{\pm 2.99}$
	-w/o focusing	57.11	15.26	23.39$_{\pm 5.17}$	58.45	24.63	34.39$_{\pm 5.01}$	67.56	27.85	39.28$_{\pm 3.35}$
	-w/o filtering	47.69	28.22	35.03$_{\pm 7.65}$	53.55	38.35	44.56$_{\pm 3.58}$	62.10	52.51	**56.76**$_{\pm 1.97}$
	-w/o contrastive learning	44.73	28.39	34.60$_{\pm 8.27}$	52.41	39.18	44.71$_{\pm 5.70}$	58.67	49.48	53.58$_{\pm 3.11}$
	-w/o soft prompt	45.27	28.90	34.83$_{\pm 8.28}$	51.02	37.42	43.11$_{\pm 4.18}$	58.94	48.37	53.00$_{\pm 3.38}$
	-w/o contextual prompt	48.35	27.29	33.82$_{\pm 5.07}$	52.13	38.43	43.94$_{\pm 4.75}$	57.63	44.29	49.97$_{\pm 4.00}$
	-w/o prompt	23.55	9.82	12.64$_{\pm 4.08}$	34.65	22.66	26.06$_{\pm 5.85}$	43.45	34.12	37.65$_{\pm 3.88}$
ACE2005	Full model	44.74	33.05	**37.74**$_{\pm 5.33}$	46.83	38.85	**42.25**$_{\pm 10.65}$	58.02	48.5	**52.71**$_{\pm 2.55}$
	-w/o focusing	44.85	19.13	26.27$_{\pm 10.72}$	55.20	24.38	33.11$_{\pm 6.11}$	68.60	31.95	43.52$_{\pm 2.81}$
	-w/o filtering	39.96	26.13	31.40$_{\pm 10.17}$	52.36	41.32	**45.93**$_{\pm 4.27}$	58.26	51.72	**54.67**$_{\pm 2.52}$
	-w/o contrastive learning	41.56	30.92	35.35$_{\pm 7.52}$	45.87	35.09	39.53$_{\pm 11.51}$	53.90	49.97	51.82$_{\pm 2.79}$
	-w/o soft prompt	40.86	30.23	34.32$_{\pm 6.92}$	48.88	35.80	40.53$_{\pm 8.66}$	55.42	49.25	52.46$_{\pm 4.16}$
	-w/o contextual prompt	39.73	27.46	32.25$_{\pm 10.21}$	51.49	36.31	41.87$_{\pm 10.25}$	55.57	47.80	51.31$_{\pm 3.19}$
	-w/o prompt	22.02	12.37	13.59$_{\pm 8.54}$	34.05	19.86	24.09$_{\pm 7.54}$	46.02	35.69	39.88$_{\pm 3.48}$
KBP2017	Full model	44.68	27.20	**33.50**$_{\pm 4.37}$	50.69	39.43	**44.21**$_{\pm 4.64}$	56.39	52.70	**54.27**$_{\pm 5.07}$
	-w/o focusing	52.21	24.08	32.59$_{\pm 7.89}$	60.27	33.99	43.24$_{\pm 7.55}$	59.75	42.31	48.79$_{\pm 4.75}$
	-w/o filtering	41.18	21.37	27.14$_{\pm 6.96}$	52.24	33.21	40.38$_{\pm 7.48}$	57.06	51.26	53.73$_{\pm 5.33}$
	-w/o contrastive learning	46.21	25.35	32.45$_{\pm 4.54}$	51.53	38.53	44.04$_{\pm 4.75}$	54.94	51.59	52.99$_{\pm 5.65}$
	-w/o soft prompt	47.76	25.77	33.16$_{\pm 6.00}$	52.11	41.64	**45.87**$_{\pm 5.28}$	54.97	51.56	53.16$_{\pm 5.45}$
	-w/o contextual prompt	46.13	26.04	32.86$_{\pm 6.21}$	55.75	36.00	43.32$_{\pm 4.79}$	55.21	50.28	52.41$_{\pm 3.53}$
	-w/o prompt	13.55	7.74	9.34$_{\pm 4.36}$	20.21	14.65	16.33$_{\pm 9.35}$	24.60	20.73	22.05$_{\pm 7.41}$

Table 10: Ablation study of FIT and baselines on the three datasets under different shots.

Methods	ACE2004		ACE2005		GENIA		KBP2017	
	train	test	train	test	train	test	train	test
Locate and Label	88.41s	39.28ms	88.95s	24.97ms	89.57s	35.28ms	65.30s	39.12ms
SEE-Few	159.39s	61.82ms	207.70s	36.14ms	160.18s	37.00ms	150.72s	52.44ms
SDNet	58.04s	37.86ms	71.86s	46.66ms	121.05s	50.97ms	40.04s	31.63ms
FIT(ours)	122.37s	31.99ms	147.40s	26.94ms	156.24s	36.89ms	105.15s	42.57ms

Table 11: Time usage on the 5-shot setting. Note that: 1) Locate and Label is a fully supervised method, and the other three are few-shot setting methods. 2) For SDNet, the time usage for training does not include the time usage for validation while others include it. 3) The ESD in the baseline is not included in the discussion because the method needs to be pre-trained on a large-scale dataset in advance.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
8 (Limitations)

☒ A2. Did you discuss any potential risks of your work?
We do not discuss them due to the space limitation, but we believe our study does not involve these potential risks.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract; 1 Introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

4 Experiments

☑ B1. Did you cite the creators of artifacts you used?
4.1 Datasets; 4.2 Experiment Settings; 4.3 Baselines; A.1 Statistics of Nested Datasets; A.2 Detailed Parameter Settings; A.4 Baselines

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
A.1 Statistics of Nested Datasets;

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
We do not discuss them due to the space limitation, but the artifacts we have used are consistent with their intended use.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The data we use are from datasets commonly used in the previous work, and sensitive information has been handled in the previous work.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
A.1 Statistics of Nested Datasets

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
A.1 Statistics of Nested Datasets

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

4 Experiment

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
A.2 Detailed Parameter Settings; 5 Time Complexity

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
A.2 Detailed Parameter Settings

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4.2 Experiment Settings; 4.4 Experiment Results; 4.5 Ablation Study; 6 Discussion

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
A.2 Detailed Parameter Settings

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.