TECA: A Two-stage Approach with Controllable Attention Soft Prompt for Few-shot Nested Named Entity Recognition

Yuanyuan Xu, Linhai Zhang, Deyu Zhou*
School of Computer Science and Engineering, Southeast University, Nanjing, China
Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China
{yuanyuan-xu,lzhang472,d.zhou}@seu.edu.cn

Abstract

Few-shot nested named entity recognition (NER), identifying named entities that are nested with a small number of labeled data, has attracted much attention. Recently, a span-based method based on three stages (focusing, bridging and prompting) has been proposed for few-shot nested NER. However, such a span-based approach for few-shot nested NER suffers from two challenges: 1) error propagation because of its 3 stage pipeline based framework; 2) ignoring the relationship between inner and outer entities, which is crucial for few-shot nested NER. Therefore, in this work, we propose a two-stage approach with a controllable attention soft prompt for few-shot nested named entity recognition (TECA). It consists of two components: span part identification and entity mention recognition. The span part identification provides possible entity mentions without an extra filtering module. The entity mention recognition pays fine-grained attention to the inner and outer entities and the corresponding adjacent context through the controllable attention soft prompt to classify the candidate entity mentions. Experimental results show that the TECA 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 5.62% on ACE04, 5.11% on ACE05, 3.41% on KBP2017 and 0.7% on GENIA on the 10-shot setting.

Keywords: few-shot nested named entity recognition, prompt-learning, attention weight

1. Introduction

Named Entity Recognition (NER) is a basic task of information extraction (Tjong Kim Sang and De Meulder, 2003), which aims to locate entity mentions and label specific entity types such as person, location, organization, or some types unique to a certain vertical scenario. It serves as a crucial component for many structured information extraction tasks, such as relation extraction (Li and Ji, 2014; Miwa and Bansal, 2016) and event extraction (McClosky et al., 2011; Wadden et al., 2019). In some situations, it makes sense to allow entities to be nested inside other entities (Ohta et al., 2002a; Alex et al., 2007), named nested NER.

However, in practice, nested NER encounters the challenge of scarce annotated data, commonly referred to as few-shot nested NER (Xu et al., 2023b). In previous work to address the issue of scarce annotations in few-shot flat NER, some opt to augment dataset labeling automatically by leveraging external specific knowledge base, such as methods like distant supervision and self-training (Xu et al., 2023a; Li et al., 2023). Others choose to mine knowledge from a limited set of annotated data, employing methods like knowledge transfer (Chen et al., 2022) or solely relying on few-shot learning (Ma et al., 2022; Yang et al., 2022).

Recently, FIT (Xu et al., 2023b), a three-stage pipeline approach is proposed for few-shot nested NER without using source domain data, which to the best of our knowledge is the only viable few-shot nested NER approach. The first two stages play the role of entity mention extraction, and the third stage classifies entity mentions based on soft prompts and contrastive learning. However, it has the following drawbacks: 1) serious error propagation issues. As shown in Figure 1 (a), in the first two stages of FIT, 30.54% and 10.22% of entities are incorrectly filtered out respectively, resulting in only 59.24% of entities entering the classification stage. 2) ignoring the relationship between inner and outer entities. Based on our observation, we found that the inner and outer nested entities tend to have a stronger semantic correlation. For

(a)

Stage	Stage1	Stage2	Total
Erroneous Filtering Rate	30.54%	10.22%	40.76%

(b)

Types	DNA	RNA	Cell_line	Cell_type
Nested Ratio	67.17%	15.33%	9.77%	5.73%

(c)

The News Agency reported Flad chairing the meeting as the delegation leader.
PER
PER

Figure 1: (a) Erroneous filtering rate of the FIT method. (b) The proportions of the entities of Protein being nested with the entities of other types in GENIA. (c) An example sentence marked with nested entities.

*Corresponding author

LREC-COLING 2024, pages 15698–15710
20-25 May, 2024.

example, as shown in Figure 1 (c): "[ ] chairing the meeting as the delegation leader", as part of the outer entity, is a description of the inner entity "Flad" that explains Flad's role and responsibilities in the meeting. This semantic correlation helps label "Flad" more like a PER (person) than a LOC (location). Other findings also indicate that some types of entities are prone to be nested with each other more frequently. For example, 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 in the GENIA dataset as shown in Figure 1 (b). It is crucial to utilize the relationship between inner and outer entities in few-shot nested NER.

To address the issues mentioned above, we propose a novel two-stage approach with controllable attention soft prompts (TECA). First, the first two stages of FIT are merged into one stage, namely the span part identification, to alleviate the error propagation issue. Specifically, the span part obtained by the IO tag classifier effectively reduces the simple negatives by filtering out the continuous O-tag parts. And then each span part is enumerated directly, to obtain the possible entity mentions without additional filtering modules. Subsequently, in the entity mention recognition stage, prompt learning with controllable attention soft prompts is conducted to label each possible entity mention. Specifically, the controllable attention soft prompts forced the soft prompt toward specific parts of the context during the initial learning phase, enabling the module to pay fine-grained attention to the adjacent context of inner and outer entity mentions capturing their relationships, including their semantic correlation. Moreover, by enumerating span parts identified in the first stage, a large set of potential entity mentions partially overlapping with actual entities (referred to as hard negatives) was generated, which helps to train a more proficient classifier.

Our contributions can be summarized as follows:

- A novel two-stage method with a controllable attention soft prompt (TECA) for few-shot nested NER was proposed to alleviate error propagation and reduce the reliance between stages.
- Controllable attention soft prompts were proposed, enabling the module to pay fine-grained attention to the inner and outer entity mentions and their corresponding adjacent context, which aims to capture the relationships between inner and outer entity mentions.
- Experimental results show that TECA 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 5.62% on ACE2004, 5.11% on ACE2005, 3.41% on KBP2017 and 0.7% on GENIA on 10-shot setting.

2. Method

2.1. Overall Architecture

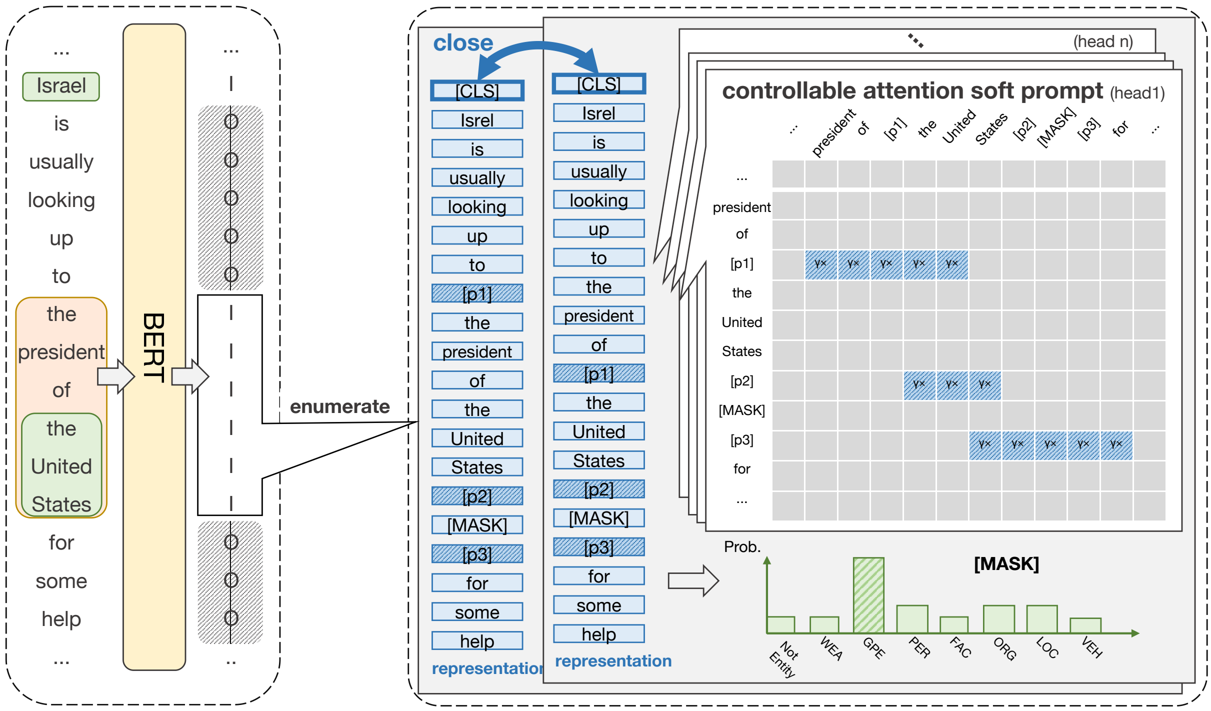
Given an input sentence $\mathcal{X} = \{x_1, \ldots, x_n\}$ of n tokens, nested NER aims to detect all the entity mentions and the corresponding types. Let $\mathcal{E} = \{\mathbf{e_1}, \ldots, \mathbf{e_n}\}$ be the set of possible entity mentions in $\mathcal{X}$. The task of nested NER is, for each entity mentions $\mathbf{e_i} \in \mathcal{E}$, to assign its label $y_i \in \mathcal{Y} \cup \{\epsilon\}$, where ϵ is the non-entity type, and $\mathcal{Y}$ is the set of pre-defined entity classes. Unlike flat NEs, nested NEs always overlap and the tokens in nested NEs may be assigned multiple types.

We formalize nested NER as span part identification and entity mention recognition. Figure 2 illustrates how the proposed approach TECA works. In the span part identification stage, the span parts, such as "the president of the United States" shown in Figure 2, are obtained. And then enumerating each span part directly to obtain the possible entity mentions without another filtering module for alleviating error propagation issues. In the entity mention recognition stage, the prompt learning with controllable attention soft prompt is conducted on the possible entity mentions. Entity mentions such as "the United States" are collected and then classified. Moreover, by enumerating span parts in the first stage, a larger set of possible entity mentions partially overlap with actual entities (referred to as hard negatives) was generated, which contributes to training a more proficient labeler, reducing its reliance on the performance of the previous stage.

2.2. Span Part Identification

Given an input text $\mathcal{X} = \{x_1, \ldots, x_n\}$ consisting of n tokens, the span part identification stage aims to locate continuous sequences of tokens marked with the I-tag and filter out the O-tag parts, as illustrated in Figure 2. These identified span parts are denoted as $\mathcal{S} = \{\mathbf{s_1}, \ldots, \mathbf{s_k}\}$, where each $\mathbf{s_i} = \{x_l, \ldots, x_r\}$ in $\mathcal{X}$ represents the i-th span part, with x_l and x_r indicating the left and right boundary tokens respectively. To achieve this, an IO classifier was trained. The specific implementation details are outlined below.

First, the input text is encoded using BERT to obtain the representation $h \in \mathbb{R}^{n \times d}$, where d represents the dimension of the BERT hidden states.



(a) Span Part Identification

(b) Entity Mention Recognition

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

The representation h_i^{tag} for each token x_i is a concatenation of token representation h_i and the representation of the [CLS] token $h^{[\mathrm{CLS}]}$.

$$h_i^{tag} = \mathrm{Concat}(h^{[\mathrm{CLS}]}, h_i) \quad (1)$$

Then, the probability p_i^{tag} is calculated as:

$$p_i^{tag} = \mathrm{Softmax}(\mathrm{MLP}_{\mathrm{tag}}(h_i^{tag})) \quad (2)$$

where MLP denotes the multilayer perceptron for binary classification. The determination of whether a token is part of a span is calculated as:

$$\hat{y}_i^{tag} = \arg\max(p_i^{tag}) \quad (3)$$

For the binary classifier, we employ the cross-entropy loss:

$$\mathcal{L}_{tag} = \sum_i \mathrm{CrossEntropyLoss}(p_i^{tag}, y_i^{tag}) \quad (4)$$

where y_i^{tag} represents the ground truth label. A value of 1 indicates that x_i is part of an entity, while 0 denotes that it is not.

2.3. Entity Mention Recognition

Possible entity mentions, denoted as $\mathcal{E}$, are derived by enumerating each span part $\mathrm{s_i}$ obtained in the span part identification stage. Then the prompt learning with controllable attention soft prompt is conducted to label each possible entity mention with the corresponding type. The implementation details are outlined below.

Let $\mathcal{M}$ be a pre-trained language model on a large-scale corpus. Prompt learning formalizes the classification task into a masked language modeling problem. Following the common practice in prompt learning (Schick and Schütze, 2021), the model $\mathcal{M}$ is tasked with predicting the label in the [MASK] position. Followed (Xu et al., 2023b), for each possible entity mention $\boldsymbol{e_i}$, we wrap it into template:

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

where $[\mathrm{p_i}]$ denotes the soft prompt. As an example, consider the span "the United States" in the sentence $\mathcal{X}$ shown in Figure 2, we wrap it into: $\boldsymbol{x}_{\mathrm{p}} =$ "Israel is usually looking up to the president of $[\mathrm{p}_1]$ the United States $[\mathrm{p}_2][\mathrm{MASK}][\mathrm{p}_3]$ for some help". To achieve fine-grained mining of inner and outer nested entities, we employ Controllable Attention Soft Prompts. Subsequently, $\mathcal{M}$ predicts the probability of each label y to fill the $[\mathrm{MASK}]$ position, and the predicted label $\hat{y}$ is:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} P_{\mathcal{M}}([\mathrm{MASK}] = y \mid \boldsymbol{x}_{\mathrm{p}}) \quad (5)$$

This objective function can be effectively optimized by the cross-entropy loss. Additionally, a sentence can be converted into a set of $\boldsymbol{x}_{\mathrm{p}}$ since there may be more than one possible entity mention in a sentence. The distinction between $\boldsymbol{x}_{\mathrm{p}}$ in the same set lies solely in the different positions of the soft prompts inserted. Consequently, these sentence representations should be proximate, prompting the introduction of a sentence-level proximity loss:

$$\mathcal{L}_{close}(\boldsymbol{x}_{\mathrm{p}_1}, \boldsymbol{x}_{\mathrm{p}_2}) = 1 - \cos(\boldsymbol{x}_{\mathrm{p}_1}^{[\mathrm{CLS}]}, \boldsymbol{x}_{\mathrm{p}_2}^{[\mathrm{CLS}]}) \quad (6)$$

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

2.4. Controllable Attention Soft Prompt

In order to pay fine-grained attention to the context of the inner and outer entity mentions, we proposed the controllable attention soft prompt. This involves regulating the attention weight of the soft prompt, directing it toward specific parts of the context. In particular, we refer to these specific parts as the "attention window".

As shown in Algorithm 1, denoting the position indices of the soft prompts in $\boldsymbol{x}_{\mathrm{p}}$ as $\mathcal{P} = \{p_1, p_2, p_3\}$, where p_i is the position index of the soft prompt $\mathrm{p_i}$. For each soft prompt $\mathrm{p_i}$, a distinct attention window $\mathbf{w_i} = \{w_{il}, \ldots, w_{ir}\}$ was calculated based on p_i, where w_{il} and w_{il} denote the left and right indices of the attention window respectively. For the attention weight matrix $\mathbf{A}$ of a certain layer, we enhance the weight in the attention window:

$$\mathbf{A}[:][p_i][\mathbf{w_i}] = \gamma \cdot \mathbf{A}[:][p_i][\mathbf{w_i}] \quad (7)$$

where γ represents the multiplier, and $[:]$ denotes all of attention heads.

This approach enables the soft prompts $\mathrm{p_1}$ and $\mathrm{p_3}$ on the left and right sides of the entity to serve a dual purpose. On the one hand, they serve as segmentation cues, aiding the model in understanding the boundaries of the entity mention required labeling. On the other hand, they pay more fine-grained attention to the adjacent context of the entity mention to learn their relationship, and learn the segmentation information of inner and outer entities (since the segmentation boundaries of inner and outer nested entities may also be included in the attention window). For the soft prompt $\mathrm{p_2}$ in the middle, it emphasizes the entity itself and the shared context of inner and outer nested entities, since we forced the window for $\mathrm{p_2}$ not to extend beyond the entity's boundary and must include the intersection of the inner and outer nested entities.

2.5. Training Objectives

The overall loss function is:

$$\mathcal{L} = \alpha\mathcal{L}_{tag} + \beta\mathcal{L}_{close} + \eta\mathcal{L}_{prompt} \quad (8)$$

Algorithm 1: Controllable Attention Soft Prompt

Input: $\mathcal{P} = \{p_1, p_2, p_3\}$, attention weight matrix $\mathbf{A}$ of a certain layer, sentence length len, the multiplier γ

Output: enhanced attention weight matrix $\mathbf{A}$

```
left ← p1, right ← p3;
δ ← max(1, (right − left + 1)//2);
w1 = [max(0, left − δ), min(left + δ, len));
w2 = (left, right);
w3 = [max(0, right − δ), min(right + δ, len));
for each head i in A do
    for each p_i in P do
        A[i][p_i][w_i] = γ · A[i][p_i][w_i];
    end
end
```

---

where $\mathcal{L}_{tag}$, $\mathcal{L}_{close}$ and $\mathcal{L}_{prompt}$ are balanced with hyper-parameters α, β, and η respectively, and $\mathcal{L}_{prompt}$ denotes the loss function used in the prompt-learning. Note that we train both two stages at the same time since the BERT embedding is shared between both two stages, and concurrent training aids the model in obtaining a more suitable embedding.

3. Experimental Settings

3.1. Datasets

Experiments are conducted on four widely used nested NER datasets: ACE2004[1] (Mitchell et al., 2005), ACE2005[2] (Walker et al., 2006), GENIA[3] (Ohta et al., 2002) and KBP2017[4] (Ellis et al., 2019). Please refer to Appendix A for the introduction. For a fair comparison, we directly used the data (Xu et al., 2023b) sampled.

3.2. Baselines

We select recent competitive models as our baselines: Biaffine-CNN-NER (Yan et al., 2023), ChatGPT-NER (Han et al., 2023), FIT (Xu et al., 2023b), SEE-Few (Yang et al., 2022), SDNet (Chen et al., 2022) and ESD (Wang et al., 2022). Biaffine-CNN-NER is a fully supervised method, and the last four are designed for the few-shot setting. In addition, we compare our method with the performance of ChatGPT reported by others (Han et al.,

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

2023). It should be noted that since most few-shot NER methods cannot solve few-shot nested NER, the methods available to us are limited. Please refer to Appendix B for detailed information.

3.3. Evaluation

Span-level precision, recall, and Micro-F_1 scores are used to measure the results. 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 measure this, following (Xu et al., 2023b) we analyze the error rates for nested entities and flat entities respectively.

3.4. Implementation Details

For few-shot learning, we conduct 1, 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 sampled for training. For a fair comparison, we asked (Xu et al., 2023b) and directly used the data they sampled. 10 sets of data for each shot and all subsequent metrics are taken from the average of these 10 sets of data. For all datasets, we train our model for 35 epochs and choose the checkpoint with the best validation performance to test. Please refer to Appendix C for more detailed settings [5].

4. Results and Analysis

4.1. Main Results

Table 1 illustrates the performance of TECA as well as baselines on ACE04, ACE05, GENIA and KBP2017. We can observe that: 1) TECA consistently outperforms all the baselines on ACE04, ACE05, GENIA, and KBP2017 datasets. In particular, TECA outperforms the FIT method, which is also dedicated to processing the few-shot nested NER task. 2) For fully supervised methods, the idea of fusing information around nested entities, as proposed by the Biaffine-CNN-NER approach, bears a strong resemblance to our starting point. However, they perform poorly on few-shot nested NER, suggesting that complex modules such as the combination of Biaffine and CNN may have inherent flaws in few-shot NER. Table 2 illustrates the error rates on the ACE04 dataset under few-shot settings. We can observe that: Among all methods, TECA significantly reduces the error rates of nested entities on the ACE04 dataset. In addition, we also calculated the erroneous filtering rate of TECA on the ACE04 dataset, which is 31.7%, smaller than FIT's 40.76%, showing our approach alleviates the error propagation issue. We also used the gold span to evaluate the performance of the second stage and the results show +1.21% over FIT on 5-shot setting in ACE04. Moreover, to illustrate the decoupling effect of our method, we enumerate the entire sentence to get the possible entity mentions but retain the training of the IO tag classifier module to simulate the situation where there is a failure to filter negatives well in the first stage. The result shows that the F1 score of our model outperforms the FIT by +15.14% on average. This shows that our labeler works better and is less dependent on the previous stage.

4.2. Comparison with ChatGPT

We compare the performance of TECA with that of ChatGPT reported by (Han et al., 2023) and the results are shown in Table 3. The proposed method, TECA, is competitive to ChatGPT by +1.66% and +3.02% on the ACE04 and ACE05 datasets in the 5-shot setting respectively. On the GENIA dataset, TECA performs significantly worse than ChatGPT, which we attribute to the fact that ChatGPT exhibits much better performance on flat NER than it does on nested NER due to their autoregressive architecture (Han et al., 2023; Bubeck et al., 2023). The GENIA dataset has a nesting rate of only 21.78% on the test set, lower than the 46.69% on ACE04 and 39.08% on ACE05, and thus ChatGPT has a natural advantage on the GENIA dataset.

4.3. Ablation Study

We conduct ablation experiments and the results are shown in Table 4.

W/o sentence-level closing. Directly remove the sentence-level closing loss during training and the rest remains the same. The results show that sentence-level closing improves the model's performance by shortening the distance of sentence-level representation [CLS].

W/o controllable attention soft prompt. Directly remove the controllable attention soft prompt and retain the vanilla soft prompt. The results show that the controllable attention soft prompt plays an important role, which we attribute to its learning of fine-grained contextual information and interactive information of inner and outer nested entities.

4.4. Parameter Analysis

Controllable attention soft prompts were added to different layers to explore their impact. Figure 5 reports the performance of the proposed

[5] Our code is available at `https://github.com/xuyy6789/TECA-NER`.

Datasets	Methods	5-shot			10-shot			20-shot		
		P	R	$F_1 \uparrow$	P	R	$F_1 \uparrow$	P	R	$F_1 \uparrow$
ACE04	Biaffine-CNN-NER	56.41	4.51	$8.20_{\pm 3.65}$	58.09	5.09	$9.27_{\pm 3.88}$	57.23	18.68	$28.05_{\pm 2.98}$
	FIT†	46.87	29.31	$35.87_{\pm 4.92}$	51.43	40.18	$44.88_{\pm 4.82}$	60.14	48.93	$53.92_{\pm 2.99}$
	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	$52.17_{\pm 3.76}$
	TECA(ours)	49.59	34.25	$\mathbf{40.18}_{\pm 4.32}$	60.03	43.78	$\mathbf{50.50}_{\pm 1.81}$	63.32	54.02	$\mathbf{58.19}_{\pm 1.57}$
ACE05	Biaffine-CNN-NER	51.92	5.83	$7.76_{\pm 2.99}$	63.21	5.06	$9.37_{\pm 2.27}$	56.01	19.35	$28.52_{\pm 5.26}$
	FIT†	44.74	33.05	$37.74_{\pm 5.33}$	46.83	38.85	$42.25_{\pm 10.65}$	58.02	48.50	$52.71_{\pm 2.55}$
	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	$50.30_{\pm 3.37}$
	TECA(ours)	49.35	32.68	$\mathbf{39.19}_{\pm 6.24}$	54.29	42.05	$\mathbf{47.36}_{\pm 2.87}$	60.33	52.43	$\mathbf{55.99}_{\pm 3.50}$
GENIA	Biaffine-CNN-NER	53.09	3.02	$5.59_{\pm 3.59}$	52.36	7.12	$12.33_{\pm 4.04}$	53.79	21.62	$30.58_{\pm 4.59}$
	FIT†	40.72	30.30	$34.43_{\pm 9.06}$	52.91	39.51	$44.95_{\pm 3.38}$	57.00	46.81	$51.26_{\pm 3.96}$
	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	$47.22_{\pm 4.36}$
	TECA(ours)	44.92	28.63	$\mathbf{34.60}_{\pm 5.88}$	54.16	40.27	$\mathbf{45.65}_{\pm 3.71}$	59.23	47.81	$\mathbf{52.69}_{\pm 3.92}$
KBP2017	Biaffine-CNN-NER	54.43	4.31	$7.82_{\pm 3.59}$	56.32	4.85	$8.83_{\pm 4.42}$	57.74	19.62	$29.18_{\pm 2.90}$
	FIT†	44.68	27.20	$33.50_{\pm 4.37}$	50.69	39.43	$44.21_{\pm 4.64}$	56.39	52.70	$54.27_{\pm 5.07}$
	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	$52.54_{\pm 3.76}$
	TECA(ours)	49.38	29.04	$\mathbf{36.14}_{\pm 6.85}$	55.58	41.92	$\mathbf{47.62}_{\pm 2.71}$	58.88	54.00	$\mathbf{56.10}_{\pm 3.05}$

Table 1: Performance comparison of TECA and baselines on four datasets under different shots. The existing results marked with † are retrieved from (Xu et al., 2023b).

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†	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†	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
TECA(ours)	**65.75**	**58.22**	**74.35**	**76.11**	**73.61**	**56.22**	**48.42**	**65.12**	**67.96**	**63.63**	**45.98**	**38.02**	**55.06**	**59.45**	**52.28**

Table 2: The error rates comparison of TECA and baselines on the ACE04 dataset under different shots. The existing results marked with † are retrieved from (Xu et al., 2023b).

Datasets	ChatGPT	TECA
	$F_1 \uparrow$	$F_1 \uparrow$
ACE04	$38.52_{\pm 2.51}$‡	$40.18_{\pm 4.32}$
ACE05	$36.17_{\pm 1.78}$‡	$39.19_{\pm 6.24}$
GENIA	$48.82_{\pm 1.31}$‡	$34.60_{\pm 5.88}$

Table 3: Performance comparison between TECA and ChatGPT on 5-shot setting. ‡ are retrieved from (Han et al., 2023) with different data

model TECA after adding controllable attention soft prompts to the first n layers under the ACE04 dataset, where n is 0 for not adding and n is 4 for adding all the first 4 layers. We can observe that: 1) In the 1-shot setting, the model outperforms by not adding or only adding controllable attention soft prompts in the lower layers rather than by adding them up to the higher layers. The possible reason is that adding controllable attention soft prompts also introduces some interference. However, the labeled data is too few to fully train the model. 2) As n increases, the performance improves to a certain extent. However, when n is 4, the performance decreases on the 5-shot, from which we believe that there is a limit to the number of layers for adding controllable attention soft prompts, i.e.,

Methods	5-shot			10-shot			20-shot		
	P	R	$F_1 \uparrow$	P	R	$F_1 \uparrow$	P	R	$F_1 \uparrow$
Full model (ACE04)	49.59	34.25	**40.18**$_{\pm 4.32}$	60.03	43.78	**50.50**$_{\pm 1.81}$	63.32	54.02	**58.19**$_{\pm 1.57}$
-w/o sentence-level closing	44.25	31.80	$36.73_{\pm 3.98}$	57.90	40.74	$47.73_{\pm 3.31}$	63.42	50.11	$55.84_{\pm 2.02}$
-w/o attention soft prompt p_1	46.11	32.98	$38.10_{\pm 5.12}$	57.61	42.02	$48.50_{\pm 3.01}$	63.10	53.17	$57.60_{\pm 1.66}$
-w/o attention soft prompt p_2	51.47	32.06	$39.16_{\pm 3.97}$	62.09	41.80	$49.88_{\pm 2.57}$	63.27	51.79	$56.83_{\pm 3.56}$
-w/o attention soft prompt p_3	50.07	33.58	$40.05_{\pm 4.48}$	56.86	44.40	$49.60_{\pm 3.34}$	63.05	51.93	$56.81_{\pm 3.10}$

Table 4: Ablation study of TECA and baselines on the ACE04 dataset under different shots.

Datasets	Layers n	1-shot			5-shot			10-shot			20-shot		
		P	R	$F_1 \uparrow$	P	R	$F_1 \uparrow$	P	R	$F_1 \uparrow$	P	R	$F_1 \uparrow$
	$n=0$	32.88	10.11	$15.18_{\pm 6.51}$	46.05	30.68	$35.35_{\pm 5.11}$	57.35	43.80	$49.48_{\pm 3.15}$	63.40	52.08	$57.15_{\pm 1.96}$
	$n=1$	36.19	12.14	**16.90**$_{\pm 7.59}$	51.22	27.89	$36.11_{\pm 3.70}$	58.05	43.67	$49.74_{\pm 2.52}$	63.71	52.92	$57.68_{\pm 2.35}$
ACE04	$n=2$	32.70	11.75	$16.64_{\pm 5.72}$	49.59	34.25	**40.18**$_{\pm 4.32}$	60.03	43.78	**50.50**$_{\pm 1.81}$	63.32	54.02	**58.19**$_{\pm 1.57}$
	$n=3$	32.75	10.95	$14.77_{\pm 5.86}$	49.77	33.23	$39.67_{\pm 5.00}$	58.09	44.78	$50.41_{\pm 2.62}$	64.12	52.31	$57.56_{\pm 2.41}$
	$n=4$	32.84	9.47	$14.12_{\pm 5.24}$	50.31	31.86	$38.40_{\pm 4.94}$	59.79	43.80	$50.44_{\pm 2.59}$	63.05	51.76	$56.75_{\pm 2.05}$

Table 5: Performance comparison of adding controllable attention soft prompts to different layers.

adding it at higher levels may disrupt the continuity of parameter learning. We only add controllable attention soft prompts at lower layers to direct the model to the specific parts during the initial learning phase to force the model towards specific parts of the context.

4.5. Attention Weight Visualization

In order to analyze the attention weight, attention heads in BERT are visualized for an example input sentence. We visualize the soft prompts in each attention head of the last layer of BERT (while $n = 3$, e.g., only the first three layers add controllable attention soft prompt). As shown in Figure 3, in the last layer, $[\mathrm{p}_2]$ in head2 focuses on the entity to be classified “his", while in head3, $[\mathrm{p}_2]$ pays more attention to the outer part “family". It can be seen that the attention weight on these heads reflects the model's attention to the relationship between the inner and outer entities to a certain extent. As well as on head4, when classifying the inner entity “his", the attention weights of all the soft prompts focus on the “family", which is part of the outer entity “his family". In addition, similar findings were also found in head5, head6, head9 and head12.

5. Case Study

Examples of model predictions are shown in Table 6. The first line illustrates that our model can recognize entities with nested structures. We can see that the nested entities from inside to outside are “her” and “her husband”, both of which can be accurately recognized by our model. The second line illustrates that our model still falls short in identifying long entities. As shown in the second line, our model incorrectly identifies and classifies the phrase “parts of Nebraska, Iowa, Minnesota, Wisconsin, northern Missouri, Illinois, Indiana, and Michigan”. This could be addressed through the pre-training and fine-tuning paradigm.

6. Related Work

6.1. Nested NER

Most of the existing nested NER methods focus on the fully supervised learning paradigm. According to the models used, they can be divided into: sequence-labeling-based methods (Straková et al., 2019; Wang et al., 2020; Ma et al., 2022; Huang et al., 2022b; Das et al., 2022); generative-based methods (Cui et al., 2021; Hou et al., 2022; Chen et al., 2022); span-based methods (Yan et al., 2023; Nguyen et al., 2023; Yuan et al., 2022; Huang et al., 2022a); programming-algorithm-based methods (Corro, 2023); and so on. However, these supervised nested NER methods are not suitable for the few-shot setting because of plenty of labeled data needed.

6.2. Few-shot NER

In recent years, several methods have been proposed to solve the few-shot flat NER task, which can be divided into two categories according to whether to expand the datasets. some approaches augment dataset labeling automatically by leveraging external specific knowledge base. These weakly supervised datasets are then used to train

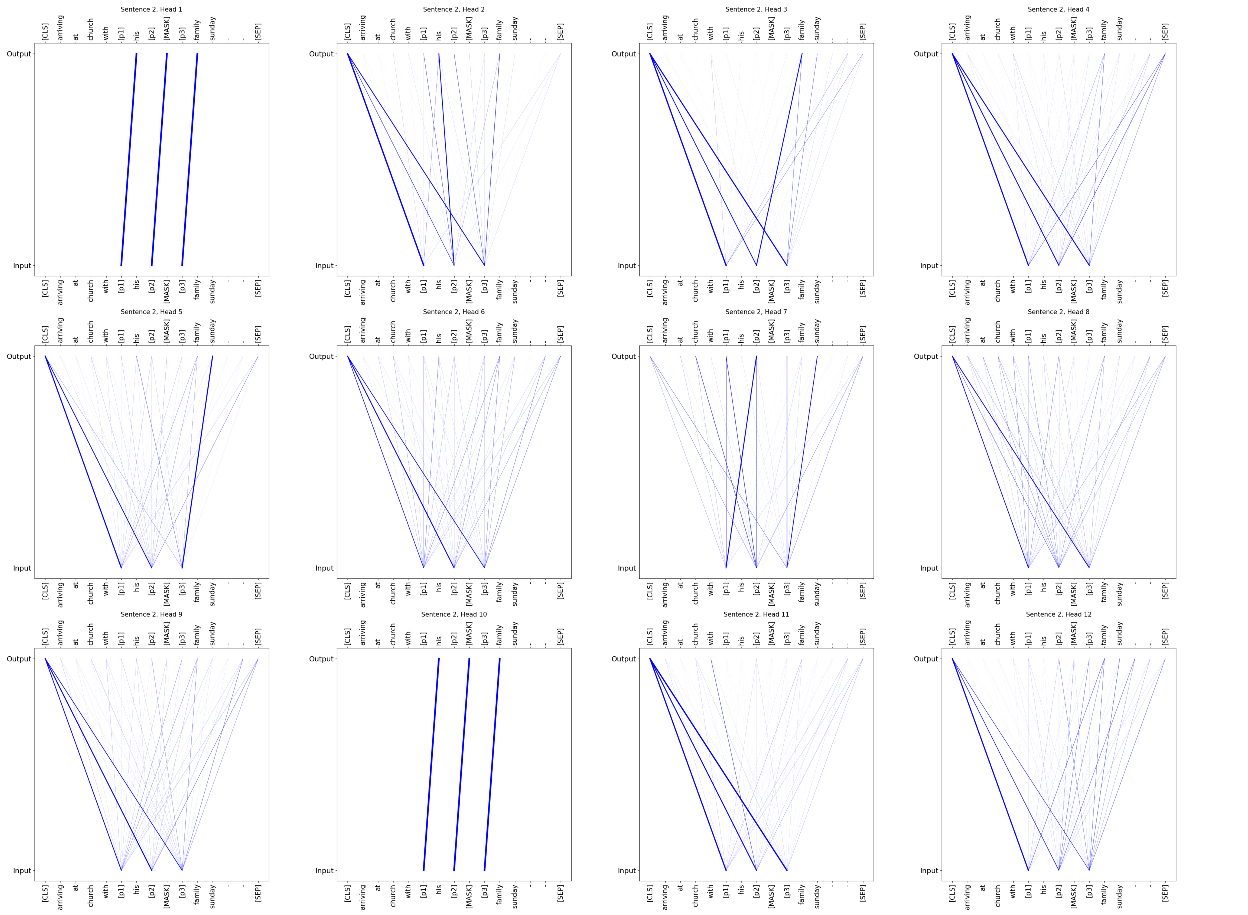


Figure 3: Attention heads in BERT are visualized for an example input sentence in the ACE04 dataset. In this example sentence, his is the inner entity to be classified and $his family$ is the corresponding outer entity. The darker the blue line, the greater the attention weight.

In a press release, $[^1[^1\text{LVMH}^1]_{\text{ORG}}{}^1]_{\text{ORG}}$ said $[^1[^1\text{it}^1]_{\text{ORG}}{}^1]_{\text{ORG}}$ aimed to combine $[^1[^1\text{Gabrielle}^1]_{\text{PER}}{}^1]_{\text{ORG}}$ and $[^1[^1\text{Donna Karan International}^1]_{\text{ORG}}{}^1]_{\text{ORG}}$ and that $[^1[^1\text{it}^1]_{\text{ORG}}{}^1]_{\text{ORG}}$ expected that $[^1[^1\text{Karan}^1]_{\text{PER}}{}^1]_{\text{PER}}$ and $[^2[^2[^1[^1\text{her}^1]_{\text{PER}}{}^1]_{\text{PER}}\text{ husband}^2]_{\text{PER}}{}^2]_{\text{PER}}$ "will exchange a significant portion of their $[^1\text{DKI}^1]_{\text{ORG}}$ shares for, and purchase additional stock in, $[^1[^1\text{the combined entity}^1]_{\text{ORG}}{}^1]_{\text{ORG}}$."
An area of low pressure area over $[^1[^1\text{the Midwest}^1]_{\text{LOC}}{}^1]_{\text{LOC}}$ carried light to moderate snow across $[^2[^2\text{parts of } [^1[^1\text{Nebraska}^1]_{\text{GPE}}{}^1]_{\text{GPE}}{}^2]_{\text{GPE}}$, $[^1[^1\text{Iowa}^1]_{\text{GPE}}{}^1]_{\text{GPE}}$, $[^1[^1\text{Minnesota}^1]_{\text{GPE}}{}^1]_{\text{GPE}}$, $[^1[^1\text{Wisconsin}^1]_{\text{GPE}}{}^1]_{\text{GPE}}$, $[^1[^1\text{northern Missouri}^1]_{\text{LOC}}{}^1]_{\text{LOC}}$, $[^1[^1\text{Illinois}^1]_{\text{GPE}}{}^1]_{\text{GPE}}$, $[^1[^1\text{Indiana}^1]_{\text{GPE}}{}^1]_{\text{GPE}}$, and $[^1[^1\text{Michigan}^1]_{\text{GPE}}{}^1]_{\text{GPE}}{}^2]_{\text{LOC}}$.

Table 6: Cases Study. Blue brackets indicate entities predicted by the model, red brackets indicate true entities, the labels in the lower right corner indicate the type of entity, and the superscripts indicate the level of the nesting.

the model, such as methods like distant supervision and self-training (Xu et al., 2023a; Li et al., 2023; Si et al., 2023; Ma et al., 2023). Others choose to thoroughly mine knowledge from a limited set of annotated samples, employing methods like knowledge transfer (Chen et al., 2022; Das et al., 2022; Zhang et al., 2023; Chen et al., 2023; Fang et al., 2023) or solely relying on few-shot learning (Ma et al., 2022; Xu et al., 2023b; Huang et al., 2022b; Yang et al., 2022). To the best of our knowledge, FIT (Xu et al., 2023b) is the only viable few-shot nested NER approach. FIT is a three-stage pipeline method for few-shot nested NER without using source domain data. Both focusing and bridging stages play the role of entity mentions extraction, and the prompting stage classifies entity mentions based on soft prompts and contrastive learning. However, the three-stage pipeline method has serious error propagation issues.

7. Conclusion

In this work, we propose a two-stage method for few-shot nested NER without using source domain data. The span part identification stage, with an IO tag classifier and enumerating without an extra filtering module, provides possible entity mentions. The entity mention recognition stage pays fine-grained attention to the inner and outer entities and the corresponding adjacent context through the controllable attention soft prompt to classify the possible entity mentions. Our proposed method, TECA, alleviates the error propagation issues effectively and learns the relationship between inner and outer entities. Experimental results show that our method achieves state-of-the-art performance consistently on the four benchmark datasets (including ACE2004, ACE2005, GENIA, and KBP2017), and outperforms several competing baseline models on F1-score and the corresponding error rates of nested entities.

8. Acknowledgements

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

9. Bibliographical References

Beatrice Alex, Barry Haddow, and Claire Grover. 2007. Recognising nested named entities in biomedical text. In *Biological, translational, and clinical language processing*, pages 65–72, Prague, Czech Republic. Association for Computational Linguistics.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

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.

Yanru Chen, Yanan Zheng, and Zhilin Yang. 2023. Prompt-based metric learning for few-shot NER. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7199–7212, Toronto, Canada. Association for Computational Linguistics.

Caio Corro. 2023. A dynamic programming algorithm for span-based nested named-entity recognition in $o(n^2)$. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10712–10724, Toronto, Canada. Association for Computational Linguistics.

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.

Jinyuan Fang, Xiaobin Wang, Zaiqiao Meng, Pengjun Xie, Fei Huang, and Yong Jiang. 2023. MANNER: A variational memory-augmented model for cross domain few-shot named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4261–4276, Toronto, Canada. Association for Computational Linguistics.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *arXiv preprint arXiv:2305.14450*.

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.

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.

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.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

Qi Li, Tingyu Xie, Peng Peng, Hongwei Wang, and Gaoang Wang. 2023. A class-rebalancing self-training framework for distantly-supervised named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11054–11068, Toronto, Canada. Association for Computational Linguistics.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Linyang Li, Qi Zhang, and Xuanjing Huang. 2022. 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.

Zhiyuan Ma, Jintao Du, and Shuheng Zhou. 2023. Noise-robust training with dynamic loss and contrastive learning for distantly-supervised named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10119–10128, Toronto, Canada. Association for Computational Linguistics.

David McClosky, Mihai Surdeanu, and Christopher Manning. 2011. Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1626–1635, Portland, Oregon, USA. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Nhung T. H. Nguyen, Makoto Miwa, and Sophia Ananiadou. 2023. Span-based named entity recognition by generating and compressing information. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1984–1996, Dubrovnik, Croatia. Association for Computational Linguistics.

Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. 2002a. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the Second International*

Conference on Human Language Technology Research, HLT '02, page 82–86, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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

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.

Shuzheng Si, Zefan Cai, Shuang Zeng, Guoqiang Feng, Jiaxing Lin, and Baobao Chang. 2023. SANTA: Separate strategies for inaccurate and incomplete annotation noise in distantly-supervised named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3883–3896, Toronto, Canada. 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.

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

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

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.

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. 2022. An enhanced span-based decomposition method for few-shot sequence labeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5012–5024, Seattle, United States. Association for Computational Linguistics.

Lu Xu, Lidong Bing, and Wei Lu. 2023a. Sampling better negatives for distantly supervised named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4874–4882, Toronto, Canada. Association for Computational Linguistics.

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

Hang Yan, Yu Sun, Xiaonan Li, and Xipeng Qiu. 2023. An embarrassingly easy but strong baseline for nested named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1442–1452, Toronto, Canada. Association for Computational Linguistics.

Zeng Yang, Linhai Zhang, and Deyu Zhou. 2022. SEE-few: 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.

Shan Zhang, Bin Cao, Tianming Zhang, Yuqi Liu, and Jing Fan. 2023. Task-adaptive label dependency transfer for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3280–3293, Toronto, Canada. Association for Computational Linguistics.

10. Language Resource References

Joe Ellis and Jeremy Getman and Stephanie Strassel. 2019. *TAC KBP Evaluation Source Corpora 2016-2017*. LDC. Linguistic Data Consortium: LDC2019T12, ISLRN 221-511-301-129-0.

Alexis Mitchell and Stephanie Strassel and Shudong Huang and Ramez Zakhary. 2005. *ACE 2004 Multilingual Training Corpus*. LDC. Linguistic Data Consortium: LDC2005T09, ISLRN 789-870-824-708-5.

Tomoko Ohta and Yuka Tateisi and Jin-Dong Kim and Hideki Mima and Junichi Tsujii. 2002. *GENIA corpus*. GENIA Project. GENIA Project, ISLRN 905-770-498-692-0.

Christopher Walker and Stephanie Strassel and Julie Medero and Kazuaki Maeda. 2006. *ACE 2005 Multilingual Training Corpus*. LDC. Linguistic Data Consortium: LDC2006T06, ISLRN 458-031-085-383-4.

A. 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.

GENIA (Ohta et al., 2002b) is a biology nested named entity dataset and contains five entity types, including DNA, RNA, protein, cell line, and cell type categories.

KBP2017 (Ji et al., 2017) has 5 entity categories, including GPE, ORG, PER, LOC, and FAC.

[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

Table 7 reports 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.

B. Baselines

We select the following models as baselines for few-shot nested NER. The first one is a model under the fully supervised setting, and the last four are models under the few-shot setting.

- **Biaffine-CNN-NER** (Yan et al., 2023) combine the biaffine and CNN to recognize NEs. First, a multi-head Biaffine decoder is used to generate the representation of each adjacent span, and then CNN is used to model the interaction of adjacent spans. Lastly, the representation incorporating information from adjacent spans is used for classification.
- **FIT** (Xu et al., 2023b) is based on focusing, bridging, and prompting pipeline for few shot nested NER without using source domain data. Both focusing and bridging stages play the role of entity mentions extraction, and the prompting stage classifies entity mentions based on soft prompts and contrastive learning.
- **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., 2022) 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. (2022) 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 pretrained on it.

Dataset Statistics	ACE04			ACE05			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 7: Statistics of the four datasets used in the experiments.

Tags	ACE04	ACE05	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 8: Verbalizer used in the prompting stage.

C. Implementation Details

We implement TECA with Huggingface Transformers 4.11.3 and PyTorch 1.7.1. In most experiments, we use BERT (Devlin et al., 2019) as PLM. For the GENIA dataset, replacing BERT with BioBERT (Lee et al., 2019). In the experimental details, we use `bert-base-uncased`[10] for ACE2004, ACE2005 and KBP2017 datasets and `dmis-lab/biobert-base-cased-v1.2`[11] for GENIA dataset (the two model sizes: all about 110M). The soft prompts are initialized by the embedding of "," "(" and ")". The verbalizer is a simple 1-to-1 mapping as shown in Table 8, 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 io classifier with a rate of 0.1. Please see Table 9 for details. We train our model on a single NVIDIA 3090 GPU with 24GB memory. For all datasets, we train our model for 35 epochs and choose the checkpoint with the best validation performance to test. The model usually converges in less than 35 epochs. Taking the 5-shot of the ACE04 dataset as an example, the model converges in the 13th epoch on average in 10 groups of samples (variance is 15.96).

P	**ACE04**	**ACE05**	**KBP17**	**GENIA**
lr	3e-05	3e-05	3e-05	3e-05
The first stage batch size	1	1	1	1
The second stage batch size	8	8	8	8
n	2	3	3/2/3	3
α	1.0			
β	1.0			
η	1.0			
γ	1.1			
drop out rate	0.1			
lr_warmup	0.1			
weight_TECAy	0.01			

Table 9: Detailed Parameter(P) Settings. 3/2/3 means $n = 3$ for 5-shot and 20-shot, $n = 2$ for 10-shot.

[10] https://huggingface.co/bert-base-uncased
[11] https://huggingface.co/dmis-lab/biobert-base-cased-v1.2