SEE-Few: Seed, Expand and Entail for Few-shot Named Entity Recognition

Zeng Yang and **Linhai Zhang** and **Deyu Zhou***
School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, China
{yangzeng, lzhang472, d.zhou}@seu.edu.cn

Abstract

Few-shot named entity recognition (NER) aims at identifying named entities based on only few labeled instances. Current few-shot NER methods focus on leveraging existing datasets in the rich-resource domains which might fail in a training-from-scratch setting where no source-domain data is used. To tackle training-from-scratch setting, it is crucial to make full use of the annotation information (the boundaries and entity types). Therefore, in this paper, we propose a novel multi-task (Seed, Expand and Entail) learning framework, SEE-Few, for Few-shot NER without using source domain data. The seeding and expanding modules are responsible for providing as accurate candidate spans as possible for the entailing module. The entailing module reformulates span classification as a textual entailment task, leveraging both the contextual clues and entity type information. All the three modules share the same text encoder and are jointly learned. Experimental results on four benchmark datasets under the training-from-scratch setting show that the proposed method outperformed state-of-the-art few-shot NER methods with a large margin. Our code is available at https://github.com/unveiled-the-red-hat/SEE-Few.

1 Introduction

Named entity recognition (NER), focusing on identifying mention spans in text inputs and classifying them into the pre-defined entity categories, is a fundamental task in natural language processing and widely used in downstream tasks (Wang et al., 2019; Zhou et al., 2021; Peng et al., 2022). Supervised NER has been intensively studied and yielded significant progress, especially with the aid of pre-trained language models (Devlin et al., 2019; Li et al., 2020; Mengge et al., 2020; Yu et al., 2020; Shen et al., 2021; Li et al., 2021a; Chen and Kong, 2021). However, supervised NER relies on plenty of training data, which is not suitable for some specific situations with few training data.

Few-shot NER, aiming at recognizing entities based on few labeled instances, has attracted much attention in the research filed. Approaches for few-shot NER can be roughly divided into two categories, span-based and sequence-labeling-based methods. Span-based approaches enumerate text spans in input texts and classify each span based on its corresponding template score (Cui et al., 2021). Sequence-labeling-based approaches treat NER as a sequence labeling problem which assigns a tag for each token using the BIO or IO tagging scheme (Yang and Katiyar, 2020; Hou et al., 2020; Huang et al., 2021). Most of these span-based and sequence-labeling-based methods focus on leveraging existing datasets in the rich-resource domains to improve their performance in the low-resource domains. Unfortunately, the gap between the source domains and the target domains may hinder the performance of these methods (Pan and Yang, 2009; Cui et al., 2021). Moreover, these approaches might fail under the training-from-scratch setting where no source domain data is available.

Therefore, it is crucial to make full use of the in-domain annotations, which consist of two types of information: boundary information and entity type information. However, most the approaches mentioned above fail to fully utilize these information. (1) Most span-based methods simply enumerate all possible spans, ignoring the boundary information of named entities. As a large number of negative spans are generated, these approaches suffer from the bias, the tendency to classify named entities as non-entities. (2) Most sequence-labeling-based methods simply employ one-hot vectors to represent entity types while ignoring the prior knowledge of entity types.

To overcome the disadvantages mentioned above, firstly, inspired by three principles for weakly-supervised image segmentation, i.e. seed,

*Corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 2540–2550
October 12–17, 2022.

expand and constrain (Kolesnikov and Lampert, 2016), we seed with relatively high-quality unigrams and bigrams in the texts, then expand them to extract the candidate spans as accurately as possible. Secondly, we cast span classification as textual entailment to naturally incorporate the entity type information. For example, to determine whether "J. K. Rowling" in "J. K. Rowling is a British author." is a PERSON entity or a non-entity, we treat "J. K. Rowling is a British author." as a premise, then construct "J. K. Rowling is a person." and "J. K. Rowling is not an entity." as hypotheses. In such way, span classification is converted into determining which hypothesis is true. Moreover, the size of training data is increased by such converting which is beneficial for few-shot settings.

In this paper, we propose SEE-Few, a novel multi-task learning framework (Seed, Expand and Entail) for Few-shot NER. The seeding and expanding modules are responsible for providing as accurate candidate spans as possible for the entailing module. Specifically, the seed selector chooses some unigrams and bigrams as seeds based on some metrics, e.g., the Intersection over Foreground. The expanding module takes a seed and the window around it into account and expands it to a candidate span. Compared with enumerating all possible n̂-gram spans, seeding and expanding can significantly reduce the number of candidate spans and alleviate the impact of negative spans in the subsequent span classification stage. The entailing module reformulates a span classification task as a textual entailment task, leveraging contextual clues and entity type information to determine whether a candidate span is an entity and what type of entity it is. All the three modules share the same text encoder and are jointly learned. Experiments were conducted on four NER datasets under training-from-scratch few-shot setting. Experimental results show that the proposed approach outperforms several state-of-the-art baselines.

The main contributions can be summarized as follows:

- A novel multi-task learning framework (Seed, Expand and Entail), SEE-Few, is proposed for few-shot NER without using source domain data. In specific, the seeding and expanding modules provide as accurate candidate spans as possible for the entailing module. The entailing module reformulates span classification as a textual entailment task, leveraging contextual clues and entity type information.
- Experiments were conducted on four NER datasets in training-from-scratch few-shot setting. Experimental results show that the proposed approach outperforms the state-of-the-art baselines by significant margins.

2 Related Work

2.1 Few-shot NER

Few-shot NER aims at recognizing entities based on only few labeled instances from each category. A few approaches have been proposed for few-shot NER. Methods based on prototypical network (Snell et al., 2017) require complex episode training (Fritzler et al., 2019; Hou et al., 2020). Yang and Katiyar (2020) abandon the complex meta-training and propose NNShot, a distance-based method with a simple nearest neighbor classifier. Huang et al. (2021) investigate three orthogonal schemes to improve the model generalization ability for few-shot NER. TemplateNER (Cui et al., 2021) enumerates all possible text spans in input text as candidate spans and classifies each span based on its corresponding template score. Ma et al. (2021) propose a template-free method to reformulate NER tasks as language modeling (LM) problems without any templates. Tong et al. (2021) propose to mine the undefined classes from miscellaneous other-class words, which also benefits few-shot NER. Ding et al. (2021) present Few-NERD, a large-scale human-annotated few-shot NER dataset to facilitate the research.

However, most of these studies follow the manner of episode training (Fritzler et al., 2019; Hou et al., 2020; Tong et al., 2021; Ding et al., 2021) or assume an available rich-resource source domain (Yang and Katiyar, 2020; Cui et al., 2021), which is in contrast to the real word application scenarios that only very limited labeled data is available for training and validation (Ma et al., 2021). EntLM (Ma et al., 2021) is implemented on training-from-scratch few-shot setting, but still needs distant supervision datasets for label word searching. The construction of distant supervision datasets requires additional expert knowledge. Some works study generating NER datasets automatically to reduce labeling costs (Kim et al., 2021; Li et al., 2021b). In this paper, we focus on the few-shot setting without source domain data

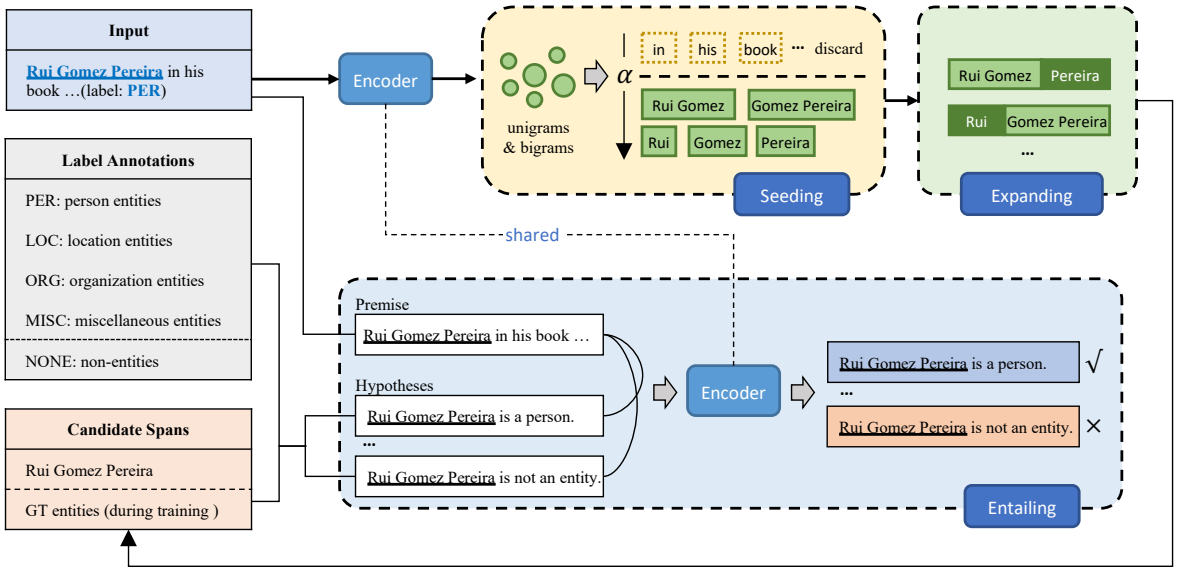


Figure 1: The architecture of the proposed approach, SEE-Few, which consists of three main modules: seeding, expanding, and entailing.

which makes minimal assumptions about available resources.

2.2 Three Principles for Weakly-Supervised Image Segmentation

Semantic image segmentation is a computer vision technique which aims at assigning a semantic class label to each pixel of an image. Kolesnikov and Lampert (2016) introduce three guiding principles for weakly-supervised semantic image segmentation: to seed with weak localization cues, to expand objects based on the information of possible classes in the image, and to constrain the segmentation with object boundaries.

3 Methodologies

3.1 Problem Setting

We decompose NER to two subtasks: span extraction and span classification. Given an input text $\mathbf{X} = \{x_1, \ldots, x_n\}$ as a sequence of tokens, a span starting from x_l and ending with x_r (i.e., $\{x_l, \ldots, x_r\}$) can be denote as $s = (l, r)$, where $1 \leq l \leq r \leq n$. The span extraction task is to obtain a candidate span set $\mathbf{C} = \{c_1, \ldots, c_m\}$ from the input text. Given an entity type set $\mathbf{T}^+ = \{t_1, \ldots, t_{v-1}\}$ and the candidate span set $\mathbf{C}$ produced by span extraction, the target of span classification is assign an entity category $t \in \mathbf{T}^+$ or the non-entity category to each candidate span. For convenience, we denote an entity type set including the non-entity type as $\mathbf{T} = \{t_1, \ldots, t_{v-1}, t_{none}\}$, where t_{none} represents the non-entity type and v is the size of $\mathbf{T}$.

3.2 The Architecture

Figure 1 illustrates the architecture of the proposed approach, SEE-Few, which consists of three main modules: seeding, expanding, and entailing. The input text will first be sent to the seeding module to generate informative seeds, then the seeds will be expanded to candidate spans in the expanding module, finally the candidate spans will be classified with an entailment task in the entailing module. We will discuss the details of each modules in the following sections.

3.2.1 Seeding

Given an input text $\mathbf{X} = \{x_1, \ldots, x_n\}$ consisting of n tokens, a unigram consists of one token and a bigram consists of two consecutive tokens. We denote the set of unigrams and bigrams in the input text as $\mathbf{S} = \{s_1, \ldots, s_{2n-1}\}$, where $s_i = (l_i, r_i)$ denotes i-th span, and l_i, r_i denote the left and right boundaries of the span respectively.

Seeding is to find the unigrams and bigrams that overlap with entities and have the potential to be expanded to named entities, which is important for the following seed expansion. It can be accomplished by constructing a seeding model and predicting the seed score for each candidate unigram

or bigram.

Firstly, 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 the span $s_i = (l_i, r_i)$, its representation h_i^{seed} is the concatenation of the mean pooled span representation h_i^p and the representation of the `[CLS]` token $h^{[\texttt{CLS}]}$. The seed score is calculated as follows:

$$h_i^p = \text{MeanPooling}(h_{l_i}, \ldots, h_{r_i}) \quad (1)$$

$$h_i^{seed} = \text{Concat}(h_i^p, h^{[\texttt{CLS}]}) \quad (2)$$

$$p_i^{seed} = \text{Sigmoid}(\text{MLP}_\text{s}(h_i^{seed})) \quad (3)$$

where MLP denotes the multilayer perceptron with a GULE function in the last layer. We set the threshold α and select the span whose seed score is above α as a seed to expand.

To train the seeding model, we need to construct a dataset consisting of unigrams (bigrams) and their seeding scores. We construct the seed score based on Intersection over Foreground (IoF). Intersection over Union (IoU) is used to measure the overlap between objects in object detection which is defined as $\text{IoU}(A, B) = \frac{A \cap B}{A \cup B}$ in NER, where A and B are two spans (Chen et al., 2020; Shen et al., 2021). However, IoU is not suitable for the seeding stage. Considering an entity consisting of five words, e.g., "International Conference on Computational Linguistics", IoU between the bigram "International Conference" and the entity is 0.4, not significant. Intersection over Foreground (IoF) can be a better choice which is defined as $\text{IoF}(A, B) = \frac{A \cap B}{A}$, where A is the foreground (i.e., a unigram or bigram) and B is the background (i.e., a named entity). In the above example, IoF between "International Conference" and the entity is 1.0, indicating that it is part of the entity and has the potential to be expanded to the whole entity. We assign each s the IoF between it and its closed named entity as the ground-truth seed score $\hat{y}_i^{seed}$ which indicates the potential to be expanded to the whole entity.

3.2.2 Expanding

For named entities consisting of more than two words, the seeds generated in seeding stage are only part of them and need to be expanded to the whole entities. Expanding is a regression task to learn the boundary offsets $\hat{o}$ between a seed and the named entity closed to the seed.

Expanding is allowed to offset the left and right boundaries of the seed by up to λ, respectively, which means that the longest entity we can get is an entity of length $2 + 2\lambda$. Besides, expanding needs to consider a window around the seed in addition to the seed itself. For the seed $s_i = (l_i, r_i)$, the maximum expansion is denoted as:

$$s_i^{exp\_max} = (\min(1, l_i - \lambda), \max(n, r_i + \lambda)) \quad (4)$$

If we use $s_i^{exp\_max}$ as the window around s_i, it may not provide enough information to distinguish the boundaries for the maximum expansion. Thus, the window for s_i should be larger than $s_i^{exp\_max}$, defined as w_i:

$$w_i^l = \min(1, l_i - 2\lambda) \quad (5)$$

$$w_i^r = \max(n, r_i + 2\lambda) \quad (6)$$

$$w_i = (w_i^l, w_i^r) \quad (7)$$

We concatenate the mean pooled span representation h_i^p of seed s_i and the mean pooled span representation h_i^w of window w_i. Then the offsets o_i of left and right boundaries are calculated as follows:

$$h_i^w = \text{MeanPooling}(h_{w_i^l}, \ldots, h_{w_i^r}) \quad (8)$$

$$h_i^{exp} = \text{Concat}(h_i^p, h_i^w) \quad (9)$$

$$o_i = \lambda \cdot (2 \cdot \text{Sigmoid}(\text{MLP}_\text{e}(h_i^{exp})) - 1) \quad (10)$$

where $o_i \in \mathbb{R}^2$. The first element of o_i can be denoted as o_i^l, indicating the offset of the seed's left boundary. Likewise, the second element o_i^r indicates the offset of the seed's right boundary, and $o_i^l, o_i^r \in [-\lambda, \lambda]$. We can obtain the result of expanding, i.e., a candidate span with the new boundaries l_i' and r_i':

$$l_i' = \max(1, l_i + \left\lfloor o_i^l + \frac{1}{2} \right\rfloor) \quad (11)$$

$$r_i' = \min(n, r_i + \left\lfloor o_i^r + \frac{1}{2} \right\rfloor) \quad (12)$$

The duplicate results and invalid results that $l_i' > r_i'$ are discarded. At this point, a set of candidate spans are produced for span classification.

3.2.3 Entailing

The entailing module reformulates span classification as a textual entailment task, leveraging contextual clues and entity type information. To cast span classification as textual entailment, we need to construct textual entailment pairs. For the i-th candidate span c_i, the entailment pair is constructed as $(\mathbf{X}, \mathbf{E}_i^j)$, where $\mathbf{E}_i^j = \{c_i, \text{is}, \text{a}, t_j\}$ and $t_j \in \mathbf{T}$. Please refer to Appendix A for detailed templates used to construct entailment pairs. The entailment label $\hat{y}_{i,j}^{entail}$ for $(\mathbf{X}, \mathbf{E}_i^j)$ can be obtained by:

$$\hat{y}_{i,j}^{entail} = \begin{cases} \texttt{entail}, & \text{if } c_i \text{ belongs to } t_j \\ \texttt{not entail}, & \text{otherwise} \end{cases} \tag{13}$$

The entailment pair $(\mathbf{X}, \mathbf{E}_i^j)$ is fed into the shared text encoder to obtain the representation of the `[CLS]` token $h_{\mathbf{E}_{i,j}}^{\texttt{[CLS]}} \in \mathbb{R}^d$ and the binary textual entailment classification can be performed:

$$p_{i,j}^{entail} = \text{Softmax}(\text{MLP}_{\text{entail}}(h_{\mathbf{E}_{i,j}}^{\texttt{[CLS]}})) \tag{14}$$

To ensure that all ground-truth entities are learned, we add ground-truth entities to the candidate span set $\mathbf{C}$ during the training phase.

3.3 Training Objective

Both seeding and expanding are regression tasks, the seeding loss $\mathcal{L}_{seed}$ and expansion loss $\mathcal{L}_{exp}$ are defined as follows:

$$\mathcal{L}_{seed} = \sum_i \text{SmoothL1}\left(\hat{y}_i^{seed}, p_i^{seed}\right) \tag{15}$$

$$\mathcal{L}_{exp} = \sum_i \sum_{j \in \{l,r\}} \text{SmoothL1}\left(\hat{o}_i^j, o_i^j\right) \tag{16}$$

For the entailing module, since the number of instances with the `not entail` label is bigger than the number of instances with the `entail` label, we use focal loss (Lin et al., 2017) to solve the label imbalance problem:

$$FL(p, y) = \begin{cases} -(1-p_{i,j})^\gamma \log(p_{i,j}), & \text{if } y = 1 \\ -(p_{i,j})^\gamma \log(1-p_{i,j}), & \text{otherwise} \end{cases} \tag{17}$$

$$\mathcal{L}_{entail} = \sum_i \sum_j FL(p_{i,j}^{entail}, \hat{y}_{i,j}^{entail}) \tag{18}$$

where γ denotes the focusing parameter of focal loss.

The multi-task framework is trained by minimizing the combined loss defined as follows:

$$\mathcal{L} = \beta_1 \mathcal{L}_{seed} + \beta_2 \mathcal{L}_{exp} + \beta_3 \mathcal{L}_{entail} \tag{19}$$

where β_1, β_2 and β_3 are hyperparameters controlling the relative contribution of the respective loss term.

3.4 Entity Decoding

The entailing module will output an entailment score $p_{i,j}^{entail}$ for the entailment pair $(\mathbf{X}, \mathbf{E}_i^j)$, where $\mathbf{E}_i^j = \{c_i, \text{is}, \text{a}, t_j\}$, $c_i \in \mathbf{C}$ and $t_j \in \mathbf{T}$. We collect all entailment pairs associated with the candidate span c_i, then assign c_i the entity type with the highest entailment score. If two candidate spans have overlap, the span with a higher score will be selected as the final result.

4 Experiment Settings

4.1 Training-from-scratch Few-shot Settings

Different from most previous few-shot NER studies that assume source-domain data is available, we consider a training-from-scratch setting, which is more practical and challenging. Specifically, we assume only K examples for each entity class in the training set and validation set respectively, where $K \in \{5, 10, 20\}$.

4.2 Datasets Construction

For fair comparison, we manually construct the few-shot datasets. With $K \in \{5, 10, 20\}$, we follow the greedy sampling strategy in (Yang and Katiyar, 2020) to ensure the sample number K of each category. To make the experimental results more convincing and credible, we randomly sample 5 different groups of training sets and validation sets for each K. We employ these strategies on four NER datasets from different domains: CoNLL2003 dataset (Sang and De Meulder, 2003) in news domain, MIT-Restaurant dataset (Liu et al., 2013) in review domain, WikiGold dataset (Balasuriya et al., 2009) in general domain and Weibo dataset (He and Sun, 2016) in social media domain. Table 2 shows the statistics on these original datasets. The self-constructed datasets are public available with the code for reproducibility.

Datasets	Methods	$K=5$			$K=10$			$K=20$		
		P	R	F_1	P	R	F_1	P	R	F_1
CoNLL03	LC-BERT	42.83	30.72	$35.06_{(6.09)}$	50.36	52.13	$51.20_{(6.39)}$	56.33	63.85	$59.84_{(1.43)}$
	Prototype	38.26	43.14	$40.37_{(8.06)}$	45.08	64.02	$52.82_{(3.22)}$	43.94	69.72	$53.89_{(1.95)}$
	NNShot	32.11	38.42	$34.92_{(3.30)}$	34.10	40.98	$37.18_{(5.82)}$	38.43	47.85	$42.61_{(2.23)}$
	StructShot	30.04	21.33	$23.43_{(4.52)}$	38.62	19.72	$26.09_{(7.23)}$	44.96	28.59	$34.87_{(1.30)}$
	TemplateNER	26.90	23.46	$23.13_{(8.40)}$	44.51	43.99	$44.01_{(4.82)}$	52.16	56.46	$54.01_{(5.09)}$
	Ours	60.45	51.27	$\mathbf{55.21}_{(3.93)}$	66.19	58.68	$\mathbf{61.99}_{(1.73)}$	69.49	67.07	$\mathbf{68.21}_{(2.60)}$
MIT-Restaurant	LC-BERT	41.21	38.65	$39.88_{(3.79)}$	43.60	48.93	$46.08_{(3.75)}$	56.24	60.04	$58.07_{(1.50)}$
	Prototype	27.77	46.79	$34.84_{(1.63)}$	30.37	50.64	$37.97_{(2.29)}$	37.91	59.31	$46.25_{(1.62)}$
	NNShot	28.15	34.81	$31.11_{(2.30)}$	30.28	37.65	$33.56_{(1.48)}$	36.72	45.55	$40.66_{(1.26)}$
	StructShot	45.13	25.00	$31.93_{(4.32)}$	43.94	28.19	$34.30_{(2.56)}$	52.08	36.18	$42.69_{(1.12)}$
	TemplateNER	23.11	20.78	$21.53_{(4.66)}$	39.45	28.77	$32.71_{(8.14)}$	46.93	37.00	$41.26_{(6.80)}$
	Ours	53.08	39.47	$\mathbf{45.25}_{(3.18)}$	57.19	46.41	$\mathbf{51.20}_{(1.48)}$	64.79	57.22	$\mathbf{60.75}_{(2.07)}$
WikiGold	LC-BERT	36.02	8.02	$12.57_{(7.81)}$	43.13	8.95	$37.72_{(7.20)}$	50.68	50.73	$50.68_{(5.94)}$
	Prototype	20.55	21.46	$19.28_{(8.12)}$	23.31	45.21	$30.59_{(3.95)}$	27.31	56.22	$36.56_{(8.65)}$
	NNShot	27.81	34.16	$30.63_{(1.91)}$	26.36	37.92	$30.93_{(4.89)}$	28.33	39.07	$32.81_{(5.41)}$
	StructShot	49.00	13.37	$20.88_{(4.61)}$	43.21	14.19	$21.28_{(2.96)}$	43.51	15.94	$23.16_{(2.18)}$
	TemplateNER	18.45	19.45	$17.26_{(12.73)}$	38.33	45.37	$41.04_{(13.19)}$	57.39	56.00	$56.60_{(3.22)}$
	Ours	61.23	41.01	$\mathbf{48.87}_{(8.01)}$	63.36	48.74	$\mathbf{54.98}_{(3.24)}$	69.06	58.25	$\mathbf{63.19}_{(1.28)}$
Weibo	LC-BERT	36.93	26.32	$29.95_{(13.93)}$	46.49	53.19	$49.54_{(3.96)}$	54.27	58.53	$56.23_{(1.48)}$
	Prototype	14.32	37.68	$20.64_{(7.07)}$	21.27	59.42	$31.25_{(2.64)}$	21.27	59.42	$37.39_{(2.58)}$
	NNShot	4.64	10.57	$06.45_{(2.65)}$	6.58	13.73	$08.90_{(1.27)}$	11.77	26.61	$16.32_{(0.80)}$
	StructShot	16.77	1.53	$02.80_{(1.63)}$	38.48	3.21	$05.91_{(1.93)}$	52.05	5.93	$10.65_{(1.73)}$
	TemplateNER	4.12	16.70	$04.41_{(4.67)}$	5.12	27.27	$08.31_{(3.11)}$	10.70	29.57	$15.24_{(7.09)}$
	Ours	49.51	48.51	$\mathbf{48.67}_{(4.05)}$	55.12	57.65	$\mathbf{56.07}_{(1.62)}$	57.10	57.70	$\mathbf{57.21}_{(1.62)}$

Table 1: Performance comparison of SEE-Few and baselines on four datasets under different Ks.

Dataset	Domain	Language	# Class	# Train	# Test
CoNLL03	News	English	4	14,987	3,684
MIT-Restaurant	Review	English	8	7,660	1,521
WikiGold	General	English	4	1,017	339
Weibo	Social Media	Chinese	8	1,350	270

Table 2: Statistics on the original datasets used to construct our few-shot datasets.

4.3 Baselines

We compare the proposed model with five competitive baselines.

LC-BERT (Devlin et al., 2019) BERT with a linear classifier which is applied to project the contextualized representation of each token into the label space.

Prototype (Huang et al., 2021) A method based on prototypical network (Snell et al., 2017), represents the entity categories as vectors in the same representation space of individual tokens and utilizes the nearest neighbor criterion to assign the entity category.

NNShot and **StructShot** (Yang and Katiyar, 2020) NNShot is a metric-based few-shot NER method that leverages a nearest neighbor classifier for few-shot prediction. StructShot is based on NNShot and use the Viterbi algorithm for decoding predictions. These methods pre-train the model with a dataset from other rich-resource domain (source domain) which is unavailable in our training-from-scratch setting. We re-implement them and directly apply them on target domains.

TemplateNER (Cui et al., 2021) A template-based prompt learning method which fine-tunes BART (Lewis et al., 2020) to generate pre-defined templates filled by enumerating text spans from input texts.

4.4 Implementation Details

For the proposed model and all the baselines except TemplateNER, we implement them based on "bert-base-uncased" for English datasets and "bert-base-chinese" for Chinese datasets. TemplateNER uses BART-large (Lewis et al., 2020) as the backbone on English datasets and Chinese BART-large (Shao et al., 2021) as the backbone on Chinese datasets. For all the baselines, we use the recommended parameters provided by the original paper or the official implementation.

For the proposed model, the number of epochs is 35. The batch sizes of seeding and expanding are 1, and the batch sizes of entailing are 16, 16, 16, 8 on CoNLL03, MIT-Restaurant, WikiGold and Weibo, respectively. The threshold αs on CoNLL03, MIT-Restaurant, WikiGold and Weibo, are set to 0.5, 0.6,

Model	CoNLL03			MIT-Restaurant			WikiGold			Weibo		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Full model	60.45	51.27	**55.21**	53.08	39.47	**45.25**	61.23	41.01	**48.87**	49.51	48.51	**48.67**
w/o seeding	63.20	45.46	52.36	53.45	38.11	44.15	61.79	36.66	45.68	56.46	40.72	47.23
w/o expanding	61.70	43.41	50.66	52.13	34.30	41.27	61.93	27.48	37.58	48.62	28.13	35.22
w/o entailing	50.93	41.15	45.01	46.60	31.43	37.13	69.39	28.47	40.02	38.32	22.58	27.14
w/o seed & exp	57.29	41.35	47.59	54.73	33.77	41.55	71.87	23.10	34.43	49.06	24.26	31.97
repl IoF with IoU	60.10	44.14	50.44	36.65	31.07	33.58	51.47	27.21	34.50	66.92	13.49	20.97

Table 3: Ablation study on 5-shot setting with the metrics of precision, recall and F1-score.

Model	CoNLL03		MIT-Restaurant		WikiGold		Weibo	
	$\|\mathbf{C}\|$	$\|\mathbf{C}\|$ / #Sen	$\|\mathbf{C}\|$	$\|\mathbf{C}\|$ / #Sen	$\|\mathbf{C}\|$	$\|\mathbf{C}\|$ / #Sen	$\|\mathbf{C}\|$	$\|\mathbf{C}\|$ / #Sen
Full model	13464 (×0.15)	3.65	7435 (×0.28)	4.48	1461 (×0.11)	4.31	1030 (×0.04)	3.81
w/o seeding	59053 (×0.66)	16.03	17780 (×0.66)	15.74	9924 (×0.72)	29.27	19772 (×0.67)	73.23
w/o expanding	15055 (×0.17)	4.09	8281 (×0.31)	5.11	1662 (×0.12)	4.90	2093 (×0.07)	7.75
w/o entailing	19350 (×0.22)	5.25	7625 (×0.28)	4.68	1656 (×0.12)	4.88	1272 (×0.04)	4.71
w/o seed & exp	89648 (×1.00)	24.33	26991 (×1.00)	17.75	13833 (×1.00)	40.81	29352 (×1.00)	108.71
repl IoF with IoU	11190 (×0.12)	3.04	4098 (×0.15)	1.76	490 (×0.04)	1.45	276 (×0.01)	1.02
#Entity / #Sent	1.53		2.07		2.15		1.55	

Table 4: Ablation study on 5-shot setting with entity-related statistics. $|\mathbf{C}|$ denotes the number of candidate spans during the testing phase. $|\mathbf{C}|$ / #Sen denotes the average number of candidate spans per sentence during the testing phase. #Entity / #Sent denotes the average number of named entities per sentence. $(\cdot)$ indicates the ratio to the number of candidate spans that produced by w/o seed & exp (e.g., the number of all unigrams and bigrams).

0.7, 0.7, respectively. λ is set to 5. The focusing parameter of focal loss γ is set to 2. β_1, β_2 and β_3 are set to 1, 1 and 1, respectively. The dropouts before the seeding, expanding and entailing are set with a rate of 0.5. The loss function is minimized using AdamW optimizer with a learning rate of 3e-05 and a linear warmup-decay learning rate schedule.

5 Experimental Results

5.1 Overall Results

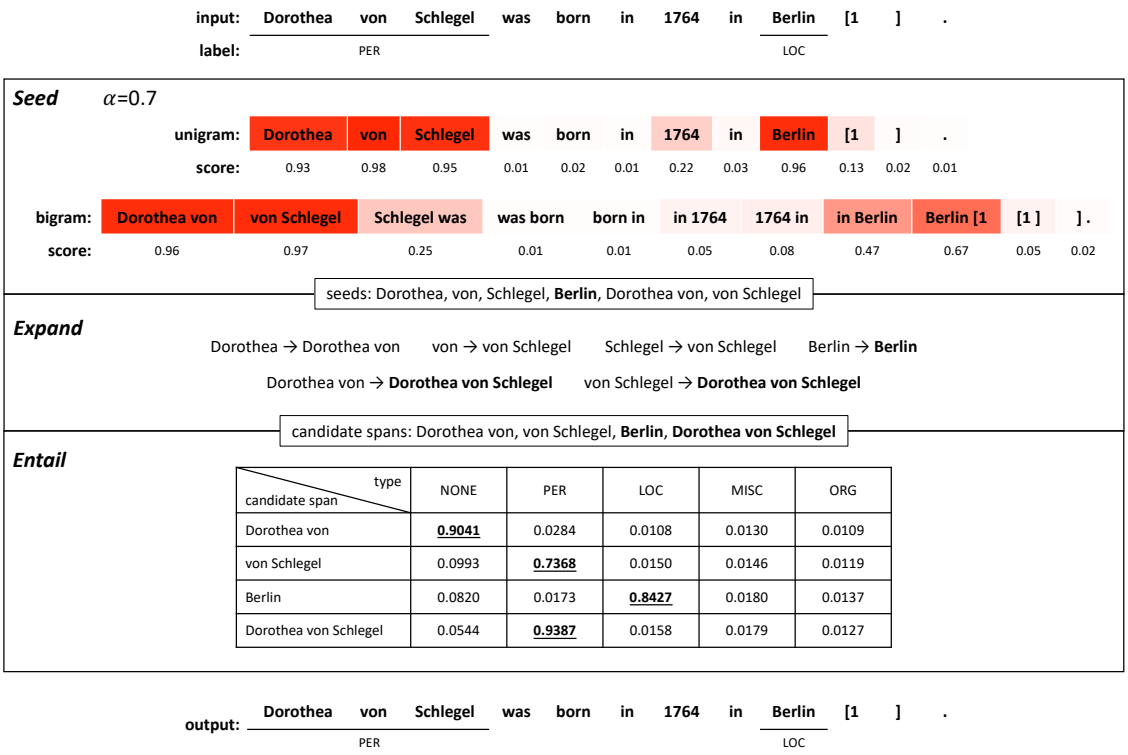
Table 1 shows the performances of the proposed method and the baselines under different K-shot settings. From the table, we can observe that: (1) The proposed method performs consistently better than all the baseline methods. Specifically, the F1-scores of our model advance previous models by +18.72%, +18.24%, +14.84%, +5.37% on Weibo, WikiGold, CoNLL03 and MIT-Restaurant respectively on 5-shot setting, which verifies the effectiveness of our approach in exploiting few-shot data. (2) Compared to baselines, our method can achieve comparable performance with less training data. Specifically, our approach achieves an F1-score of 55.21% on CoNLL03 dataset on 5-shot setting, which is better than the result of TemplateNER on 20-shot setting.

5.2 Ablation Study

To validate the effectiveness of different components in our approach, we performed ablation experiments on 5-shot setting with a series of variants of SEE-Few. Table 3 shows the results with the metrics of precision, recall and F1-score, and Table 4 demonstrates how the number of candidate spans changes in ablations. The variants are as follows:

w/o seeding: removing the seeding module and directly enumerating all unigrams and bigrams as the seeds to expand. With the aid of expanding, this variant reduces 32.25% unigrams and bigrams on average. The recalls and F1-scores drop by 4.83% and 2.15% on average, respectively. The results show that the reduction of candidate span is mainly contributed by the seeding module and this variant suffers from the bias, the tendency to classify named entities as non-entities because of a large number of negative spans.

w/o expanding: removing the expanding module and directly using the seeds as candidate spans to entail. With the aid of seeding, this variant reduces 83.25% unigrams and bigrams on average. The recalls and F1-scores drop by 11.74% and 8.32% on average, respectively. Without expanding, this variant can not identify the entities whose lengths are greater than 2, achieving worse



candidate span \ type	NONE	PER	LOC	MISC	ORG
Dorothea von	**0.9041**	0.0284	0.0108	0.0130	0.0109
von Schlegel	0.0993	**0.7368**	0.0150	0.0146	0.0119
Berlin	0.0820	0.0173	**0.8427**	0.0180	0.0137
Dorothea von Schlegel	0.0544	**0.9387**	0.0158	0.0179	0.0127

Figure 2: Case study. The result was predicted by SEE-Few trained on WikiGold, 5-shot setting.

performance.

w/o entailing: replacing the entailing module with a multi-class classifier. With the aid of seeding and expanding, this variant reduces 83.5% unigrams and bigrams on average. The F1-scores drop significantly by 12.17% on average, while dropping more sharply on datasets with fine-grained entity types (i.e., MIT-Restaurant and Weibo) than on datasets with coarse-grained entity types (i.e., CoNLL03 and WikiGold). The results demonstrate that the improvement comes from exploiting of contextual clues and label knowledge, and the entailing module can better distinguish different entity types than a multi-class classifier.

w/o seed & exp: removing both the seeding module and the expanding module in the same way as the w/o seeding and w/o seeding variants. This variant is equivalent to the entailing module classifying all the unigrams and bigrams into the entity categories or the non-entity category. In addition to a significant drop in performance, this variant is time consuming. It is on average 28.50 times slower than the full model on Weibo dataset.

repl IoF with IoU: using IoU as the ground-truth seed score instead of IoF during seeding, and keeping the threshold αs on CoNLL03, MIT-Restaurant, WikiGold and Weibo as 0.5, 0.6, 0.7, 0.7, respectively. The results demonstrate that IoF is a better choice to evaluate the qualities of unigrams and bigrams than IoU.

All the above experiments show the effectiveness of each component in our approach. Seeding and expanding can significantly reduce the number of candidate spans and alleviate the impact of negative spans in the subsequent span classification stage. The entailing module leverages contextual clues and entity type information benefiting span classification.

5.3 Case Study

Figure 2 shows an example of model predictions. We visualize the seed scores and observe that the unigrams and bigrams contained in the ground-truth entities are assigned with higher scores. The threshold α is set to 0.7 in the experiment, so "Dorothea", "von", "Schlegel", "Berlin", "Dorothea von" and "von Schlegel", totally 6 spans, are selected as seeds to expand. Among them, "Berlin" already hits the entity exactly, "Dorothea von" and "von Schlegel" are both expanded to a ground-truth entity "Dorothea von Schlegel". Other seeds are not expanded to the ground-truth entities, but do not lead an error in the final output, attributed to the success of the entailing module in deter-

mining "Dorothea von" is not an entity, and assigning a higher score to "Dorothea von Schlegel" with PER type than another candidate span (i.e., "von Schlegel") overlapping with "Dorothea von Schlegel". Considering that in a data-scarce scenario where error propagation is inevitable, our approach can still mitigate the impact of error propagation to a certain extent, which demonstrates the superiority of the proposed paradigm.

6 Conclusion

In this work, we propose a novel multi-task (Seed, Expand and Entail) learning framework, SEE-Few, for Few-shot NER without using source domain data. The seeding and expanding modules are responsible for providing as accurate candidate spans as possible for the entailing module. The entailing module reformulates span classification as a textual entailment task, leveraging both the contextual clues and entity type information. All the three modules share the same text encoder and are jointly learned. To investigate the effectiveness of the proposed method, extensive experiments are conducted under the training-from-scratch few-shot setting. The proposed method outperforms other state-of-the-art few-shot NER methods by a large margin. For future work, we will combine the framework with contrastive learning to effectively make use of limited data and further enhance the performance of few-shot NER.

Acknowledgements

The authors would like to thank the anonymous reviewers for the insightful comments. This work was funded by the National Natural Science Foundation of China (62176053).

References

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R Curran. 2009. Named entity recognition in wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web)*, pages 10–18.

Chun Chen and Fang Kong. 2021. Enhancing Entity Boundary Detection for Better Chinese 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 2: Short Papers)*, pages 20–25, Online. Association for Computational Linguistics.

Yanping Chen, Lefei Wu, Liyuan Deng, Yongbin Qing, Ruizhang Huang, Qinghua Zheng, and Ping Chen. 2020. A boundary regression model for nested named entity recognition. *arXiv preprint arXiv:2011.14330*.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using bart.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

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.

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

Hangfeng He and Xu Sun. 2016. F-score driven max margin neural network for named entity recognition in chinese social media. *CoRR*, abs/1611.04234.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. *arXiv preprint arXiv:2006.05702*.

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: A comprehensive study. *EMNLP*.

Hyunjae Kim, Jaehyo Yoo, Seunghyun Yoon, Jinhyuk Lee, and Jaewoo Kang. 2021. Simple questions generate named entity recognition datasets.

Alexander Kolesnikov and Christoph H Lampert. 2016. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European conference on computer vision*, pages 695–711. Springer.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Fei Li, Zheng Wang, Siu Cheung Hui, Lejian Liao, Dandan Song, Jing Xu, Guoxiu He, and Meihuizi Jia. 2021a. Modularized Interaction Network for 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 200–209, Online. Association for Computational Linguistics.

Jiacheng Li, Haibo Ding, Jingbo Shang, Julian McAuley, and Zhe Feng. 2021b. Weakly supervised named entity tagging with learnable logical rules.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A Unified MRC Framework for Named Entity Recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. 2013. Asgard: A portable architecture for multilingual dialogue systems. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8386–8390. IEEE.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Qi Zhang, and Xuanjing Huang. 2021. Template-free prompt tuning for few-shot ner.

Xue Mengge, Bowen Yu, Zhenyu Zhang, Tingwen Liu, Yue Zhang, and Bin Wang. 2020. Coarse-to-Fine Pre-training for Named Entity Recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6345–6354.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

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.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Fei Yang, Li Zhe, Hujun Bao, and Xipeng Qiu. 2021. Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation. *arXiv preprint arXiv:2109.05729*.

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*.

Jake Snell, Kevin Swersky, and Richard S Zemel. 2017. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*.

Meihan Tong, Shuai Wang, Bin Xu, Yixin Cao, Minghui Liu, Lei Hou, and Juanzi Li. 2021. Learning from miscellaneous other-class words for few-shot named entity recognition.

Rui Wang, Deyu Zhou, and Yulan He. 2019. Open event extraction from online text using a generative adversarial network. 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 282–291, Hong Kong, China. Association for Computational Linguistics.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named Entity Recognition as Dependency Parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

Deyu Zhou, Kai Sun, Mingqi Hu, and Yulan He. 2021. Image generation from text with entity information fusion. *Knowledge-Based Systems*, 227:107200.

A Entity Types and Templates

In the span classification stage, the entailing module reformulates span classification as a textual entailment task to leverage contextual clues and entity type information. Table 5 shows the entity types in each dataset and corresponding natural language templates we use in our experiments. Besides, for English datasets, we use "is not an entity." as the template of the non-entity type. For Chinese datasets, we use "不是命名实体。" as the template of the non-entity type.

Dataset	Entity Type	Template
CoNLL03	PER	is a person.
	LOC	is a location.
	MISC	is a miscellaneous entity.
	ORG	is an organization.
MIT Restaurant	Hours	is a time.
	Rating	is the rating.
	Amenity	is an amenity.
	Price	is the price.
	Dish	is a dish.
	Location	is a location.
	Cuisine	is is a cuisine.
	Restaurant_Name	is a restaurant name.
WikiGold	PER	is a person.
	LOC	is a location.
	MISC	is a miscellaneous entity.
	ORG	is an organization.
Weibo	GPE.NAM	是城市、国家的特指。
	GPE.NOM	是城市、国家的泛指。
	LOC.NAM	是地名的特指。
	LOC.NOM	是地名的泛指。
	ORG.NAM	是组织名的特指。
	ORG.NOM	是组织名的泛指。
	PER.NAM	是人名的特指。
	PER.NOM	是人名的泛指。

Table 5: Entity types and their corresponding natural language templates.