

Mitigating Out-of-Entity Errors in Named Entity Recognition: A Sentence-Level Strategy

Guochao Jiang^{1†}, Ziqin Luo^{1†}, Chengwei Hu^{1‡}, Zepeng Ding^{1†}, Deqing Yang^{12‡✉}

¹School of Data Science, Fudan University, Shanghai, China

²Shanghai Key Laboratory of Data Science, Shanghai, China

[†]{gcjiang22, zqluo22, zpding22}@m.fudan.edu.cn

[‡]{cwhu20, yangdeqing}@fudan.edu.cn

Abstract

Many previous models of named entity recognition (NER) suffer from the problem of Out-of-Entity (OOE), i.e., the tokens in the entity mentions of the test samples have not appeared in the training samples, which hinders the achievement of satisfactory performance. To improve OOE-NER performance, in this paper, we propose a new framework, namely S+NER, which fully leverages sentence-level information. Our S+NER achieves better OOE-NER performance mainly due to the following two particular designs. 1) It first exploits the pre-trained language model’s capability of understanding the target entity’s sentence-level context with a template set. 2) Then, it refines the sentence-level representation based on the positive and negative templates, through a contrastive learning strategy and template pooling method, to obtain better NER results. Our extensive experiments on five benchmark datasets have demonstrated that, our S+NER outperforms some state-of-the-art OOE-NER models.

1 Introduction

Named entity recognition (NER) (Li et al., 2022) plays a vital role in many downstream tasks including knowledge graph construction (Xu et al., 2017), information retrieval (Banerjee et al., 2019), question answering (Aliod et al., 2006), etc. NER task aims to recognize the span(s) of entity mention(s) in one input sentence, and further type the entity(s) mentioned by the span(s). For example, given a sentence “Aurora Couture aims to illuminate the fashion industry with its unique designs and commitment to ethical practices”, the model should output ‘Aurora Couture’ as the entity mentioned and type the entity as ‘Brand’.

In recent years, many deep NER models (Ma and Hovy, 2016; Zhu et al., 2018; Wang et al., 2020, 2022; Zhu et al., 2023; Jiang et al., 2024b) have exhibited state-of-the-art (SOTA) performance. Nevertheless, these models’ performance declines

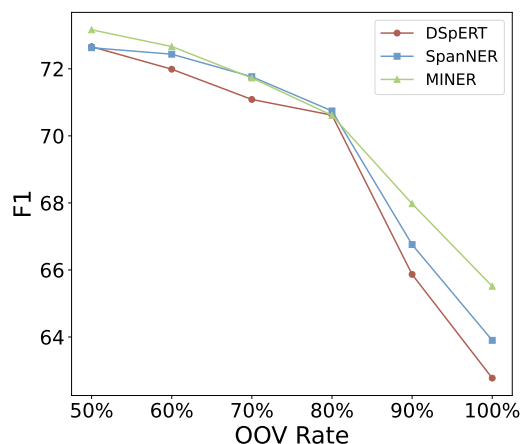


Figure 1: The span-based NER models’ F1 scores on TwitterNER dataset with different OOE rates. The OOE rate is defined as the ratio of the test entities whose mention spans have the words (tokens) not appearing in the training set, to all entities in the test set. The numbers here are the results of deduplication, and duplicate entities are not considered.

when many words (tokens) in the entity mentions have not appeared in the training set before, which is referred to as *Out-of-Entity* (OOE) problem (Mikheev et al., 1999). As shown in Figure 1, the F1 score of the representative models (SpanNER (Fu et al., 2021), MIENR (Wang et al., 2022) and DSpERT (Zhu et al., 2023)) drops apparently as the OOE rate increases. It is probably because the span representations of OOE words have not been fine-tuned during model training, while most of the previous models achieve NER mainly based on the span representations.

The previous NER models achieve the NER of OOE (OOE-NER) mainly through leveraging external knowledge, OOE word embedding, or contextualized embedding. The methods of external knowledge (Zhang and Yang, 2018; Li et al., 2018; Jiang et al., 2024a; Shi et al., 2024) enhance their recog-

dition capabilities by incorporating external information. However, external knowledge especially domain-specific knowledge, may not be obtained conveniently and cheaply. The methods of OOE word embedding (Fukuda et al., 2020; Peng et al., 2019) strive to construct more robust embeddings for OOE words, but they only leverage the semantics of the OOE words themselves without considering other useful information such as sentence-level context. Comparatively, the methods of contextualized embedding (Hu et al., 2019) incorporate the contextual information of the mentions besides the OOE word embeddings to achieve NER, through learning the appropriate embeddings of the tokens near the OOE words. Nonetheless, these models have not fully exploited the *sentence-level* contextual information of OOE words which is indeed helpful to improve OOE-NER performance further.

Intuitively, the semantic and syntactic information in the context of a new entity’s mention can prompt us to recognize the entity’s name and identify its type correctly. Recall the sentence illustrated in the first paragraph, when ‘Aurora Couture’ is replaced with another span, we still believe the new span is also the name of a brand or company. Another example is “XXX is a wonderful city.”, where the type of ‘XXX’ can be recognized as location easily through understanding the whole sentence. Inspired by it, we propose a new NER framework, namely *S+NER*, which fully leverages the mention’s contextual information to enhance OOE-NER performance. Since the OOE words in the target mention have not appeared in the training samples resulting in unsatisfactory OOE embeddings, our *S+NER* additionally incorporates the representation of the whole sentence, which is generated by a BERT-based encoder (Devlin et al., 2019). Furthermore, *S+NER* refines the sentence representation by adopting a contrastive learning strategy. As we mentioned before, contextual information is highly correlated with the correct understanding of the entity and its type. As a result, an optimal context representation in terms of accurate NER should be close to the representation of the positive template involving the entity span and its correct type, and simultaneously far away from the negative templates involving the entity span and its incorrect types. That is the basic principle of our proposed contrastive learning strategy.

Our main contributions in this paper are summarized as follows.

1. We propose a novel framework *S+NER* based an effective contrastive learning method that fully leverages the sentence-level information of the mention(s) to achieve improved OOE-NER performance.
2. We utilize the large language model (LLM) to generate a template set for OOE-NER that is both high-quality and semantically rich. To enhance performance, we incorporate sentence-level information from various templates using a pooling method designed for multiple templates.
3. We conduct extensive experiments on some benchmark datasets to justify *S+NER*’s advantage on OOE-NER over some SOTA OOE-NER models.

The rest of our paper is organized as follows. We briefly introduce the research works related to our work in Section 2 and detail our proposed framework including its special designed components in Section 3. Then, we display our extensive experiment results and provide in-depth analysis in Section 4. At last, we conclude our work in Section 5.

2 Related Work

2.1 NER Solutions for OOE Problem

By now, in the research field of NER, the OOE problem has been addressed through three primary categories of approaches as follows.

The first category incorporates external knowledge to tackle the OOE issue. Zhang and Yang (2018) exemplify this by constructing extensive entity dictionaries, thereby enhancing the model’s look-up capability. However, this approach often compromises the model’s generalization ability. Li et al. (2018) attempted to mitigate this by introducing part-of-speech tags as external knowledge. However, this method is contingent on the availability of such tags and a high-quality external knowledge base, which is often challenging to procure.

The second category focuses on enhancing OOE word embeddings. For instance, Bojanowski et al. (2017) utilized each word’s character-level n-gram to represent the OOE word embedding, given the absence of OOE words at the character level. Pinter et al. (2017) employed character-level Recurrent Neural Networks (RNNs) to capture morphological features. Other methods involve using known

words during the training process to match OOE words, subsequently replacing the OOE word embeddings with the embeddings of the matched known words. Peng et al. (2019) trained a student network to predict the closest word embedding for OOE words in the representation space. Fukuda et al. (2020) leveraged the known words similar in character surface form to the OOE words. However, these methods typically yield a static OOE embedding, neglecting the exploitation of contextual information.

The third category leverages contextual information to enhance the representation of OOE words. Hu et al. (2019) reformulated the OOE problem as a K-shot regression problem, predicting OOE embedding by aggregating K context features. Recently, researchers have utilized pre-trained language models (PLMs) (Peters et al., 2018; Devlin et al., 2019; Liu, 2019; He et al., 2021), given their demonstrated proficiency in embedding contextual words with contextual information. However, Yan et al. (2021) found that BERT does not always outperform BiLSTM-CRF in capturing contextual information. Jiang et al. (2024b) introduced ToNER to solve NER by using an entity type matching model to identify potential entity types in the sentence and leverage contextual information. In contrast to these methods, our proposed S+NER model leverages sentence-level contextual information, as opposed to word-level and character-level embeddings.

2.2 NER Solutions with Contrastive Learning

Contrastive Learning is effective in representation learning practice, which is commonly aimed at improving the alignment and uniformity of the representation space to optimize the learned representations (Wang and Isola, 2020). Therefore, contrastive learning is widely applied in NER to improve entity representations thus further improving model performance. ERICA (Qin et al., 2021) introduces an entity-level pre-training task called entity discrimination, which optimizes entity representation by contrasting the semantics of different types of entities. CONTaiNER (Das et al., 2022) introduces contrastive learning at the token-level to reduce the distance between token embeddings of the same entity type and increase that between different types, alleviating the overfitting problem during downstream task transfer. BINDER (Zhang et al., 2023) contrasts the entity type description with all the tokens in the input text from both the

entity and token perspectives, to enhance nested and discontinuous entity extraction. Similar ideas are also found in ToNER (Jiang et al., 2024b). CLLMFS (Zhang et al., 2024) incorporates contrastive loss into the instruction fine-tuning phase to enhance the LLM’s understanding of entity mentions and entity types. In this work, we utilize contrastive learning to optimize the model’s comprehension of sentence-level contextual information, thereby improving OOE-NER performance.

3 Methodology

In this section, we first introduce the span classification task for NER (in Section 3.1). Then, we introduce the model design in S+NER (in Section 3.2), which are used to obtain optimal context representation for enhanced OOE-NER.

3.1 Backbone Model

In recent years, the model architecture for NER with pre-trained language models (PLMs) has changed from the initial sequence labeling task (Chiu and Nichols, 2016; Akbik et al., 2018; Yan et al., 2019) to the span classification task (Xue et al., 2020; Yamada et al., 2020; Fu et al., 2021; Zhu et al., 2023). We choose span-based tasks for NER as the backbone of our S+NER for the following two reasons.

1. Span-based model can extract the explicit information of entity mention spans, which can be easily leveraged by an advanced NER model (Zhu et al., 2023).
2. Compared with other sequence labeling models of NER, Span-based model has demonstrated better generalization capability on OOE scenarios (Fu et al., 2021).

The classical span-based model for NER mainly consists of three parts: token representation layer, span representation layer, and span classification layer, which are introduced in turn as follows.

3.1.1 Token Representation Layer

Formally, let $X = \{x_1, x_2, \dots, x_n\}$ represent the input sentence sequence, \mathbf{h}_i denote the representation or hidden state of the i -th token x_i respectively. In most cases, they are as follows:

$$\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n = \text{Encoder}(x_1, x_2, \dots, x_n), \quad (1)$$

where $\text{Encoder}(\cdot)$ can be implemented with any network structure with context encoding function,

e.g., LSTM (Hochreiter and Schmidhuber, 1997), Transformer (Vaswani et al., 2017) and so on. In most cases, $\mathbf{h}_i \in \mathbb{R}^d$, $1 \leq i \leq n$, which means that the dimension of token embedding vector is equal to the dimension of token representation vector for PLMs. d is the dimension of the PLM, e.g. $d = 768$ for BERT-base and $d = 1024$ for BERT-large.

3.1.2 Span Representation Layer

Denote $S = \{s_1, s_2, \dots, s_m\}$ is the set of all potential spans in sequence X where b_i and e_i are the start position index and the end position index of span $s_i = (b_i, e_i)$, $1 \leq i \leq m$, respectively. For each span s_i , a label y_i is assigned to indicate whether s_i is the entity span of a certain type or non-entity span (denoted by O in most NER tasks). For example, given a sentence $X = \text{“Milan is wonderful.”}$, its potential span set is $S = \{(1, 1), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3)\}$, and the corresponding label set is $\{\text{LOC}, \text{O}, \text{O}, \text{O}, \text{O}, \text{O}\}$ where LOC indicates the span ‘Milan’ is a location entity.

For the span $s_i = (b_i, e_i)$, its representation consists of two parts in the classical span-based NER model: **boundary embedding** and **span length embedding**.

- **Boundary Embedding:** For the NER model based on span classification, the text dimension information of the span itself must be the key to downstream tasks. One method is to use the representation of each token in a span and concatenate them together as the information representation of this part. However, this method will produce representations of different dimensions for spans of different lengths, and for spans that are too long, the representation dimension will be too large. The classic method is to use the representation of two boundary tokens of span. Formally, denote \mathbf{z}_i^b as the boundary embedding. It is the concatenation of the start token’s representation \mathbf{h}_{b_i} and the end token’s representation \mathbf{h}_{e_i} .

$$\mathbf{z}_i^b = [\mathbf{h}_{b_i}; \mathbf{h}_{e_i}] \in \mathbb{R}^{2d}. \quad (2)$$

- **Span Length Embedding:** If only the boundary embedding discussed above is used as a span representation for subsequent entity classification, this method will lose the explicit positional information, even if the Transformer architecture has its own positional encoding

in most cases (Gehring et al., 2017; Su et al., 2024). At the same time, due to the use of boundary embedding, the length of the span itself cannot be used. Therefore, in general methods, an additional learnable span length embedding is added as part of the span representation. Formally, denote $\mathbf{z}_i^l \in \mathbb{R}^{d'}$ as the span length embedding, which is initialized randomly and learned through model training.

Then, denote \mathbf{z}_i as the span s_i ’s representation, which is the concatenation of the boundary embedding and span length embedding:

$$\mathbf{z}_i = [\mathbf{z}_i^b; \mathbf{z}_i^l] \in \mathbb{R}^{2d+d'}. \quad (3)$$

3.1.3 Span Classification Layer

Upon acquiring the span representation \mathbf{z}_i , it is fed into a fully connected neural network. The purpose of this operation is to classify the span in accordance with the entity type label that the model suggests for the span s_i . Formally, we denote the span classifier as a function F . This function F maps a span representation of dimension $2d + d'$ to a vector of dimension $|\mathcal{Y}|$, where \mathcal{Y} represents the entity type set. The resultant vector $F(\mathbf{z}_i)$ embodies scores that correspond to each unique label for the span s_i . The model’s predicted label \hat{y} for the span s_i is determined by identifying the label that corresponds to the highest score in the vector.

For the overlapped spans, only the span with the largest score is reserved at last. Using this heuristic decoding method to achieve flat NER (Li et al., 2022) can avoid predicting overlapped spans.

3.1.4 Learning Objective

Suppose y_i is the true label of s_i , the loss function with respect to a training sample (sentence) in span-based NER model is

$$\mathcal{L}_1 = - \sum_{s_i \in S} \text{CE}(F(\mathbf{z}_i), y_i), \quad (4)$$

where S is the sentence’s span set from which the overlapped spans have been removed, and $\text{CE}(\cdot, \cdot)$ is the cross entropy function for score vector $F(\mathbf{z}_i)$ and corresponding true label y_i .

3.2 S+NER

Besides the span’s token-level features, we further incorporate the sentence-level information of the span, i.e., the sentence representation, into our

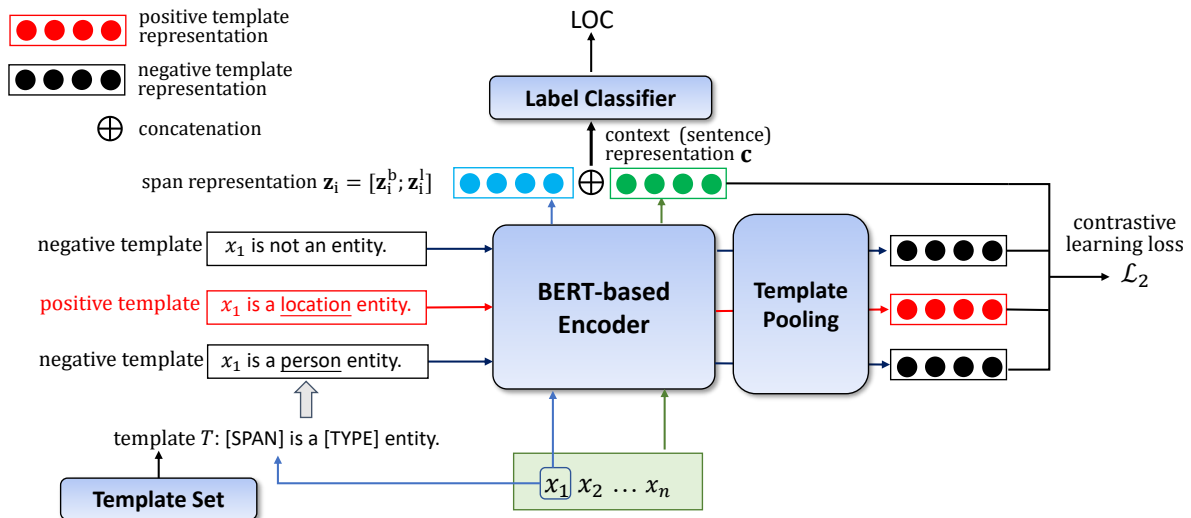


Figure 2: The overall architecture of our proposed S+NER (better viewed in color), which has two major parts: the encoding layer and the label classifier. In the encoding layer fed with the input sentence X , the target span’s representation \mathbf{z}_i is obtained. In addition, the sentence representation of X , denoted as \mathbf{c} , is originally generated by the BERT-based encoder, and then refined through the contrastive learning with the positive and negative template representations. Then, \mathbf{z}_i is concatenated with \mathbf{c} and then fed into the classifier to predict the span’s label.

framework to achieve NER. Specifically, the sentence representation is further refined by our proposed contrastive learning strategy. In this subsection, we introduce the procedure of generating and refining the sentence representation in our S+NER, of which the overall architecture is shown in Figure 2.

3.2.1 Generating Original Context Representation

As we emphasized in Section 1, the contextual information of the mention (span) is significant for a NER model to recognize the entity. An intuitive way to incorporate contextual information is to directly leverage the sentence’s representation into which the semantic and syntactic information of the sentence are encoded. Given the power of pre-trained language models (PLMs) such as BERT (Devlin et al., 2019) on understanding sentences, we employ the BERT-based encoder to encode the input sentence into an embedding at first.

Formally, given an input sentence sequence $X = \{x_1, x_2, \dots, x_n\}$, its original sentence representation is obtained by

$$\mathbf{c} = \text{SentenceEmbedding}(X) = f(X) \in \mathbb{R}^d, \quad (5)$$

where $\text{SentenceEmbedding}(\cdot)$ and $f(\cdot)$ represents the sentence encoder’s operations. In this paper, we use the average pooling of all token representations in a sentence to obtain \mathbf{c} .

Then, \mathbf{c} can be used as an auxiliary feature to help the model recognize the target span s_i more accurately. Thus s_i ’s representation \mathbf{z}_i is expanded with three parts: **boundary embedding** \mathbf{z}_i^b , **span length embedding** \mathbf{z}_i^l and **sentence representation** \mathbf{c} , that is

$$\mathbf{z}_i = [\mathbf{z}_i^b; \mathbf{z}_i^l; \mathbf{c}] \in \mathbb{R}^{3d+d'}. \quad (6)$$

At last, we feed expanded \mathbf{z}_i into the label classifier which is the same as the span-based NER model’s classification layer, to predict the label of s_i .

3.2.2 Refining Context Representation with Templates

Specifically, we formally define the *positive/negative template* as follows, which is used in our contrastive learning as the positive/negative golden standard of the context representation. Take an empty template $T = \text{“[SPAN] is a [TYPE] entity.”}$ as an example, where the position of [SPAN] will be filled with the span s_i , and the position of [TYPE] will be filled with s_i ’s correct or incorrect type (label). Thus, we get the s_i ’s positive template and negative template by filling T respectively with its positive and negative labels. To make the encoder understand the template more effectively, we further translate the label into a term of natural language before filling T , e.g., ‘LOC’ is translated into ‘location’.

Dataset	Training sen. #	Training ent. #	Test sen. #	Test ent. #	OOE rate
WNUT2017	3,394	1,592	1,286	947	100%
JNLPBA	18,545	18,899	3,856	4,344	67%
TwitterNER	4,000	4,899	3,257	4,106	79%
CoNLL2003-Typos	14,041	8,082	2,676	4,130	86%
CoNLL2003-OOE	14,041	8,082	3,453	2,689	96%

Table 1: The five datasets’ statistics of sentence number and unique entity number in the training and test sets, along with the rate of OOE entities in the test sets.

Then, the filled template is also fed into the BERT-based encoder as shown in Figure 2, to obtain the positive/negative template representation. As we mentioned in Section 1, the positive template explicitly indicates the correct type of the entity corresponding to s_i , e.g., “Milan is a location entity.”. Thus, s_i ’s context representation should be aligned with the representation of this positive golden standard, so as to help the model recognize the entity more accurately. In addition, the negative template also explicitly indicates the misunderstanding of the entity’s type, e.g., “Milan is not an entity.” or “Milan is a person entity.”. Thus, s_i ’s context representation should be simultaneously pushed away from the negative template representation as far as possible.

To leverage the rich semantic information from multiple templates, we employ the state-of-the-art LLM to generate these templates. Specifically, we utilized GPT-4 (Achiam et al., 2023) to produce 100 templates, which were then filled in as previously described. From these, we manually selected 10 representative templates. Formally, denote $\mathcal{T} = \{T_1, T_2, \dots, T_k\}$ is the template set. Let \mathcal{T}_i^+ and \mathcal{T}_i^- be the positive and negative template sets corresponding to template T_i respectively. After pooling each entity type set along the template dimension, we obtain the template representation for the entire template set. We define the loss function of contrastive learning, inspired by InfoNCE loss (Oord et al., 2018), as follows:

$$\mathcal{L}_2 = \text{InfoNCE}(\text{c}, \text{Pooling}(\mathcal{T}_i^+), \text{Pooling}(\mathcal{T}_i^-)). \quad (7)$$

Then, the overall loss of training S+NER is

$$\mathcal{L} = \mathcal{L}_1 + \lambda \mathcal{L}_2, \quad (8)$$

where λ is the weight for contrastive learning loss. A larger λ value will make the training proportion of the comparative learning part larger, and a

smaller λ value will make the training proportion of the comparative learning part smaller. A suitable λ value will make a good tradeoff between \mathcal{L}_1 and \mathcal{L}_2 , which allows the trained model to make full use of the extracted contextual information to assist in the OOE-NER task.

It is important to highlight that the contrastive learning module is not incorporated during the inference phase of S+NER. This is primarily due to the fact that the true labels of the spans in the test set are not available. Despite the lack of refinement in the context representation of a test sample during model inference, the fine-tuned parameters in S+NER ensure the context representation is adequately optimized. This optimization is crucial as it facilitates accurate prediction, thereby enhancing the overall performance and reliability of the model.

4 Experiment

In this section, we display and analyze the results of evaluating our S+NER’s OOE-NER performance upon five benchmark OOE datasets, compared with the S+NER NER models. At the same time, we have also conducted ablation experiments on S+NER to verify that each component in the S+NER is reasonable and effective.

4.1 Datasets and Evaluation Metric

We conducted our experiments on five datasets including **WNUT2017** (Derczynski et al., 2017), **JNLPBA** (Collier and Kim, 2004), **TwitterNER** (Zhang et al., 2018), **CoNLL2003-Typos** (Wang et al., 2021) and **CoNLL2003-OOE** (Wang et al., 2021).

As evaluating previous NER models (Fu et al., 2021; Wang et al., 2022), we report the entity-level micro F1 scores of all compared models in our experiments. Table 1 gives some basic information on the training set and the test set of these datasets, including the number of sentences, the number of

Model	WNUT2017	JNLPBA	TwitterNER	CoNLL2003-Typos	CoNLL2003-OOE	Avg.
BERT-Tagger	44.69	71.69	72.18	81.17	64.53	66.85
BERT-CRF	43.97	72.62	73.22	81.60	65.65	67.41
DataAug	52.29	75.85	73.69	81.73	69.60	70.63
InferNER	50.52	72.33	74.17	81.11	67.78	69.18
CoFEE	39.10	72.55	69.50	83.13	65.44	65.94
SpanNER	51.83	73.78	71.57	81.83	64.43	68.69
MINER	54.52	77.03	<u>75.26</u>	<u>87.09</u>	<u>78.03</u>	74.39
DSPERT	<u>55.32</u>	<u>81.46</u>	74.11	85.44	77.56	<u>74.78</u>
S+NER	58.27	82.70	78.01	90.96	83.52	78.69

Table 2: Overall performance comparisons between our S+NER and the baselines on the five datasets.

Para.	Value	Comment
d	1,024	token and sentence representation size
d'	50	span length embedding size
τ	1	the temperature hyperparameter in \mathcal{L}_2
λ	0.1	\mathcal{L}_2 's weight in the overall loss

Table 3: Some important hyperparameter settings for S+NER implementation.

entities, and the proportion of OOE entities that do not appear in the training set but appear in the test set.

4.2 Baselines and Implementation Details

We compared our framework with the baselines including BERT-Tagger (Devlin et al., 2019), BERT-CRF, SpanNER (Fu et al., 2021), DataAug (Dai and Adel, 2020), InferNER (Shahzad et al., 2021), CoFEE (Fukuda et al., 2020), MINER (Wang et al., 2022) in the experiments.

The implementation details of our experiments are as follows. We choose the BERT-large (Devlin et al., 2019) as S+NER’s span and sentence encoder in the following comparison experiments. The learning rate for the span classification layer is set to $5e-5$, the learning rate for BERT is set to $1e-5$, and the dropout rate for the span classification layer is set to 0.2. In order to make a trade-off between the effectiveness and performance of the model, for each input sentence exceeding 128 tokens, we only reserved its first 128 tokens. To limit the number of all extracted spans on the affordable level, we set the maximum length of a span as 4. We conducted our experiments on 1 NVIDIA GeForce RTX 4090 GPU. The checkpoint with the best performance on the validation set will be evaluated on the test set to report the final result. The F1 scores of all experimental results are obtained by averaging the results of five random experiments. Some impor-

tant hyperparameter settings are listed in Table 3, most of which were decided based on our tuning studies.

4.3 Overall Performance Comparisons

Table 2 shows the NER performance of our S+NER and all baselines on the five datasets, where the best and second best score in each dataset are bolded and underlined, respectively. In addition, we directly report the scores of InferNER, CoFEE, and MAML provided in the paper of MINER since their source codes have not been published. Except for the JNLPBA dataset, our proposed S+NER outperforms all other baselines on the WNUT2017, TwitterNER, CoNLL2003-Typos, and CoNLL2003-OOE datasets. Compared to the sub-optimal model, S+NER shows a noticeable performance improvement, especially on the WNUT2017 dataset with a 100% OOE ratio. This demonstrates that S+NER has better robustness in more severe OOE scenarios. Based on the average performance scores across the five datasets, our proposed S+NER achieves the best overall results in OOE-NER. SpanNER serves as the foundation of the span-based NER model, outperforming the two basic sequence tagging models, BERT-Tagger and BERT-CRF. DataAug enhances SpanNER with data augmentation, further improving its OOE-NER capabilities. MINER incorporates the information bottleneck theory into SpanNER to better handle OOE-NER, achieving overall sub-optimal results. By fully leveraging sentence-level information, S+NER surpasses all baselines in OOE-NER, highlighting the crucial role of context information in OOE-NER.

4.4 Ablation Study

To validate the efficacy of S+NER, we conducted an ablation study, the results of which are presented in Table 4. In this study, we compared the per-

Datasets	Models	F1
WNUT2017	SpanNER	51.83
	SpanNER + context representation	53.21
	SpanNER + template pooling & contrastive learning	58.27
JNLPBA	SpanNER	73.78
	SpanNER + context representation	77.51
	SpanNER + template pooling & contrastive learning	82.70
TwitterNER	SpanNER	71.57
	SpanNER + context representation	74.48
	SpanNER + template pooling & contrastive learning	78.01

Table 4: Ablation study results on the three datasets.

Entity Template	None-Entity Template	F1
[SPAN] is a [TYPE] entity.	[SPAN] is not an entity.	56.27
[SPAN] belongs to [TYPE] category.	[SPAN] belongs to none category.	56.92
[SPAN] should be tagged as [TYPE].	[SPAN] should be tagged as none entity.	55.75
[SPAN] can be viewed as [TYPE] entity.	[SPAN] can be viewed as none entity.	54.29
Template Pooling	Template Pooling	58.27

Table 5: Micro F1 scores of adopting different templates in WNUT2017.

formance of three methods on the WNUT2017, JNLPBA, and TwitterNER datasets. The first method is the original SpanNER, the second is SpanNER enhanced with context representation, and the third is an extension of the second method, incorporating template pooling and contrastive learning to optimize context representation, which we refer to as S+NER. The ablation study conducted on these three datasets consistently demonstrates that both context representation and contrastive learning, with the templates for context representation, positively impact the performance of the model.

4.5 Effects of Different Templates

As previously discussed, S+NER utilizes a template-based contrastive learning and template pooling method. To evaluate the performance of single template on S+NER for OOE-NER, we designed different templates inspired by Cui et al. (2021) and obtained the results in WNUT2017 shown in Table 5. From the experimental results, it is evident that S+NER is sensitive to template selection, with different templates directly affecting the final OOE-NER performance. The template pooling method, based on multiple templates, concurrently achieved optimal results, which underscores the efficacy of template pooling.

4.6 Effects of Different Encoders

We have argued that the special designs in S+NER, i.e., incorporating the context representation and refining it by contrastive learning, are model-agnostic. They can be plugged into the models with a sentence encoder. To verify this characteristic, we compared the variants of SpanNER and S+NER with different pre-trained language models as the encoder, including BERT-large (Devlin et al., 2019), RoBERTa-large (Liu, 2019) and DeBERTa-large (He et al., 2021). The comparison results shown in Table 6 imply that different pre-trained language models have different capabilities of natural language understanding, resulting in different NER performances. In addition, no matter which pre-trained language model is adopted as the encoder, S+NER consistently outperforms SpanNER, justifying the effectiveness of our special designs.

4.7 Effects on Different OOE Rates

We focus on the OOE-NER task in this paper, so we further investigated the compared models' performance in the scenarios with different OOE rates. To this end, we re-partitioned the training and test set to achieve different OOE rates in the dataset, meanwhile keeping the sentence and entity number as close as possible to the original dataset.

We display the results of our S+NER on TwitterNER compared with SpanNER, DSpERT and the previous competitive model MINER in Figure

Model	WNUT2017	JNLPBA	TwitterNER
SpanNER(BERT-large)	51.83	73.78	71.57
S+NER(BERT-large)	58.27	82.70	78.01
SpanNER(RoBERTa-large)	52.47	74.67	73.50
S+NER(RoBERTa-large)	59.03	79.13	77.60
SpanNER(DeBERTa-large)	52.75	74.37	73.87
S+NER(DeBERTa-large)	58.97	78.59	79.37

Table 6: Micro F1 scores of adopting different encoders in SpanNER and S+NER.

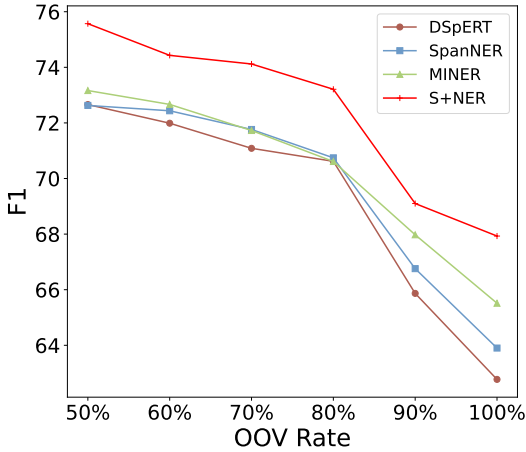


Figure 3: The performance of SpanNER, DSPErT, MINER and S+NER on different OOE rates of TwitterNER. It is worth noting that the TwitterNER dataset here is re-partitioned the training and test set to achieve different OOE rates.

3, where their performance on six different OOE rates are displayed. It shows that, although all models’ performance degrades as the OOE rate increases, S+NER outperforms the three baselines on all OOE rates consistently. This result highlights the benefits of our proposed S+NER in OOE-NER compared to previous models. As the OOE rate increases (>80%), all models exhibit a significant drop in performance, underscoring the severity of the OOE issue.

5 Conclusion

In this paper, we propose a novel framework S+NER to handle Out-of-Entity Named Entity Recognition, in which we have two major designs: incorporating the sentence representation of the input sequence and then refining it with a contrastive learning and template pooling strategy. These two designs can help our framework better understand the contextual information of the target entity, and thus alleviate the OOE problem for achieving bet-

ter OOE-NER. Our extensive experiments demonstrate our S+NER’s advantage over the SOTA OOE-NER models. In addition, we also conduct experiments to examine the impact of various templates, pre-trained language model encoders, and OOE rates on the performance of S+NER. Through case studies, we intuitively demonstrate that S+NER outperforms other models.

Limitations

As we analyzed in the experiment section, our proposed framework relies on the PLM’s capability of understanding the span’s context. For the data from a professional field or a special domain, a PLM may not well understand the contexts since it is generally pre-trained with the corpus of open domains. In such scenario, re-pre-training the PLM or fine-tuning with the data from the special domain is expected for achieving satisfactory NER performance.

Acknowledgements

This work was supported by the Chinese NSF Major Research Plan (No.92270121).

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. [Contextual string embeddings for sequence labeling](#). In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1638–1649. Association for Computational Linguistics.
- Diego Mollá Aliod, Menno van Zaanen, and Daniel Smith. 2006. [Named entity recognition for question answering](#). In *Proceedings of the Australasian Language Technology Workshop, ALTA 2006, Sydney,*

- Australia, November 30-December 1, 2006, pages 51–58. Australasian Language Technology Association.
- Partha Sarathy Banerjee, Baisakhi Chakraborty, Deepak Tripathi, Hardik Gupta, and Sourabh S. Kumar. 2019. [A information retrieval based on question and answering and NER for unstructured information without using SQL](#). *Wirel. Pers. Commun.*, 108(3):1909–1931.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomáš Mikolov. 2017. [Enriching word vectors with subword information](#). *Trans. Assoc. Comput. Linguistics*, 5:135–146.
- Jason P. C. Chiu and Eric Nichols. 2016. [Named entity recognition with bidirectional lstm-cnns](#). *Trans. Assoc. Comput. Linguistics*, 4:357–370.
- Nigel Collier and Jin-Dong Kim. 2004. [Introduction to the bio-entity recognition task at JNLPBA](#). In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, NLPBA/BioNLP 2004, Geneva, Switzerland, August 28-29, 2004*.
- 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, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1835–1845. Association for Computational Linguistics.
- Xiang Dai and Heike Adel. 2020. [An analysis of simple data augmentation for named entity recognition](#). In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 3861–3867. International Committee on Computational Linguistics.
- Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca J. Passonneau, and Rui Zhang. 2022. [Container: Few-shot named entity recognition via contrastive learning](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6338–6353. Association for Computational Linguistics.
- Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. [Results of the WNUT2017 shared task on novel and emerging entity recognition](#). In *Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 140–147. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.
- Jinlan Fu, Xuanjing Huang, and Pengfei Liu. 2021. [Spanner: Named entity re-/recognition as span prediction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7183–7195. Association for Computational Linguistics.
- Nobukazu Fukuda, Naoki Yoshinaga, and Masaru Kit-suregawa. 2020. [Robust backed-off estimation of out-of-vocabulary embeddings](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4827–4838. Association for Computational Linguistics.
- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. [Convolutional sequence learning](#). In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252. PMLR.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. [Deberta: decoding-enhanced bert with disentangled attention](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long short-term memory](#). *Neural Comput.*, 9(8):1735–1780.
- Ziniu Hu, Ting Chen, Kai-Wei Chang, and Yizhou Sun. 2019. [Few-shot representation learning for out-of-vocabulary words](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4102–4112. Association for Computational Linguistics.
- Guochao Jiang, Zepeng Ding, Yuchen Shi, and Deqing Yang. 2024a. [P-icl: Point in-context learning for named entity recognition with large language models](#). *arXiv preprint arXiv:2405.04960*.
- Guochao Jiang, Ziqin Luo, Yuchen Shi, Dixuan Wang, Jiaqing Liang, and Deqing Yang. 2024b. [Toner: Type-oriented named entity recognition with generative language model](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 16251–16262. ELRA and ICCL.
- Changliang Li, Liang Li, and Ji Qi. 2018. [A self-attentive model with gate mechanism for spoken language understanding](#). In *Proceedings of the 2018*

- Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3824–3833. Association for Computational Linguistics.
- Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2022. [A survey on deep learning for named entity recognition](#). *IEEE Trans. Knowl. Data Eng.*, 34(1):50–70.
- Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Xuezhe Ma and Eduard H. Hovy. 2016. [End-to-end sequence labeling via bi-directional lstm-cnns-crf](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.
- Andrei Mikheev, Marc Moens, and Claire Grover. 1999. [Named entity recognition without gazetteers](#). In *EACL 1999, 9th Conference of the European Chapter of the Association for Computational Linguistics, June 8-12, 1999, University of Bergen, Bergen, Norway*, pages 1–8. The Association for Computer Linguistics.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- Minlong Peng, Qi Zhang, Xiaoyu Xing, Tao Gui, Jinlan Fu, and Xuanjing Huang. 2019. [Learning task-specific representation for novel words in sequence labeling](#). In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5146–5152. ijcai.org.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.
- Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. 2017. [Mimicking word embeddings using subword rnns](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 102–112. Association for Computational Linguistics.
- Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2021. [ERICA: improving entity and relation understanding for pre-trained language models via contrastive learning](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3350–3363. Association for Computational Linguistics.
- Moemmur Shahzad, Ayesha Amin, Diego Esteves, and Axel-Cyrille Ngonga Ngomo. 2021. [Inferner: an attentive model leveraging the sentence-level information for named entity recognition in microblogs](#). In *Proceedings of the Thirty-Fourth International Florida Artificial Intelligence Research Society Conference, North Miami Beach, Florida, USA, May 17-19, 2021*.
- Yuchen Shi, Guochao Jiang, Tian Qiu, and Deqing Yang. 2024. [Agentre: An agent-based framework for navigating complex information landscapes in relation extraction](#). In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, pages 2045–2055. ACM.
- Jianlin Su, Murtadha H. M. Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. [Roformer: Enhanced transformer with rotary position embedding](#). *Neurocomputing*, 568:127063.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.
- Jiuniu Wang, Wenjia Xu, Xingyu Fu, Guangluan Xu, and Yirong Wu. 2020. [ASTRAL: adversarial trained LSTM-CNN for named entity recognition](#). *Knowl. Based Syst.*, 197:105842.
- Tongzhou Wang and Phillip Isola. 2020. [Understanding contrastive representation learning through alignment and uniformity on the hypersphere](#). In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of Proceedings of Machine Learning Research*, pages 9929–9939. PMLR.
- Xiao Wang, Shihan Dou, Limao Xiong, Yicheng Zou, Qi Zhang, Tao Gui, Liang Qiao, Zhanzhan Cheng, and Xuanjing Huang. 2022. [MINER: improving out-of-vocabulary named entity recognition from an information theoretic perspective](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 5590–5600. Association for Computational Linguistics.
- Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li, Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, Yiding Tan, Yuan Hu,

- Qiyuan Bian, Zhihua Liu, Shan Qin, Bolin Zhu, Xiaoyu Xing, Jinlan Fu, Yue Zhang, Minlong Peng, Xiqing Zheng, Yaqian Zhou, Zhongyu Wei, Xipeng Qiu, and Xuanjing Huang. 2021. [Textflint: Unified multilingual robustness evaluation toolkit for natural language processing](#). In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL 2021 - System Demonstrations, Online, August 1-6, 2021*, pages 347–355. Association for Computational Linguistics.
- Bo Xu, Yong Xu, Jiaqing Liang, Chenhao Xie, Bin Liang, Wanyun Cui, and Yanghua Xiao. 2017. [Cndbpedia: A never-ending chinese knowledge extraction system](#). In *Advances in Artificial Intelligence: From Theory to Practice - 30th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2017, Arras, France, June 27-30, 2017, Proceedings, Part II*, volume 10351 of *Lecture Notes in Computer Science*, pages 428–438. Springer.
- Mengge Xue, 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 2020, Online, November 16-20, 2020*, pages 6345–6354. Association for Computational Linguistics.
- Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. [LUKE: deep contextualized entity representations with entity-aware self-attention](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6442–6454. Association for Computational Linguistics.
- Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. [Tener: adapting transformer encoder for named entity recognition](#). *arXiv preprint arXiv:1911.04474*.
- Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. [A unified generative framework for various NER subtasks](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5808–5822. Association for Computational Linguistics.
- Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang. 2018. [Adaptive co-attention network for named entity recognition in tweets](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5674–5681. AAAI Press.
- Sheng Zhang, Hao Cheng, Jianfeng Gao, and Hoifung Poon. 2023. [Optimizing bi-encoder for named entity recognition via contrastive learning](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Yafeng Zhang, Zilan Yu, Yuang Huang, and Jing Tang. 2024. [Cllmfs: A contrastive learning enhanced large language model framework for few-shot named entity recognition](#). *arXiv preprint arXiv:2408.12834*.
- Yue Zhang and Jie Yang. 2018. [Chinese NER using lattice LSTM](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1554–1564. Association for Computational Linguistics.
- Enwei Zhu, Yiyang Liu, and Jinpeng Li. 2023. [Deep span representations for named entity recognition](#). In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10565–10582. Association for Computational Linguistics.
- Qile Zhu, Xiaolin Li, Ana Conesa, and Cécile Pereira. 2018. [GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text](#). *Bioinform.*, 34(9):1547–1554.