Auxiliary Information Enhanced Span-Based Model for Nested Named Entity Recognition

Yiming Sun(✉), Chenyang Li, and Weihao Kong

School of Computer Science and Technology, Changchun University of Science and Technology, Changchun, China
{sunyiming,2021100903,2022200133}@mails.cust.edu.cn

Abstract. Span-based methods have unique advantages for solving nested named entity recognition (NER) problems. As primary information, boundaries play a crucial role in span representation. However, auxiliary information, which assists in identifying entities, still needs to be adequately investigated. In this work, We propose a simple yet effective method to enhance classification performance using boundaries and auxiliary information. Our model mainly consists of an adaptive convolution layer, an information-aware layer, and an information-agnostic layer. Adaptive convolution layers dynamically acquire words at different distances to enhance position-aware head and tail representations of spans. Information-aware and information-agnostic layers selectively incorporate boundaries and auxiliary information into the span representation and maintain boundary-oriented. Experiments show that our method outperforms the previous span-based methods and achieves state-of-the-art F_1 scores on four NER datasets named ACE2005, ACE2004, Weibo and Resume. Experiments also show comparable results on GENIA and CoNLL2003.

Keywords: Named entity recognition · span-based methods · auxiliary information enhanced

1 Introduction

Named entity recognition (NER) has been regarded as a fundamental task in natural language processing. Previously, flat NER was treated as a sequence labeling that requires assigning a label to each word in a sentence accordingly [12,25,34]. This requires an assumption that the entities should be short and that there should be no overlap between them. However, in real applications, as illustrated in Fig. 1(a), an organizational noun may be nested in a personal noun. The emergence of nested entities makes the assumption no longer applicable. Therefore, it is necessary to design a model that can identify flat and nested entities. Recent methods of nested NER can be divided into four categories: 1) *sequence labeling methods* has been improved for identifying nested entities. Some

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14302, pp. 209–221, 2023.
https://doi.org/10.1007/978-3-031-44693-1_17

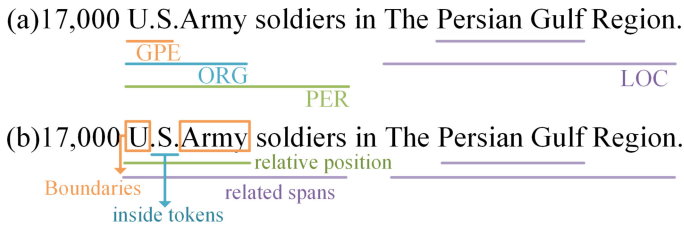

Fig. 1. (a) An example sentence with nested entities from ACE2005 (b) Information that can help determine the entity.

works overlay flat NER layers [9,23] to identify nested entities. However, such practice is prone to error propagation. 2) *hypergraph-based methods* represent all entity segments as graph nodes and combine them to represent hypergraph [17]. However, such methods suffer from structural errors and structural ambiguities during inference. 3) *sequence-to-sequence methods* generate entities directly [29], which leads to inefficiencies in the decoding process and common drawbacks of sequence-to-sequence(Seq2Seq) models, such as exposure bias. 4) *span-based methods* enumerate all spans in a sentence and classify them accordingly. The approach takes the boundaries as the key to constitute the span representation [19,35]. However, only the boundaries cannot effectively detect complex nested entities [32], so focusing only on the boundaries is not comprehensive.

As shown in Fig. 1, the information available for a span to be identified includes not only the boundaries but also the auxiliary information such as inside tokens, labels, related spans, and relative positions. The utilization of the above information is helpful to solve the entity recognition problem. Although there have been works to utilize them [6,27], some issues still need to be addressed. Firstly, enumerating all possible spans in a sentence using related spans is computationally expensive. Secondly, they can only leverage part of the aforementioned auxiliary information, and most overlook relative positions' importance. Lastly, the use of related spans involves the challenge of subjective selection, which can lead to error.

In order to solve the problems mentioned above, we propose a simple but effective method to simultaneously utilize all the above-mentioned auxiliary information. The key of our model is to propose an **A**uxiliary **I**nformation **E**nhanced **S**pan-based NER (**AIESNER**) neural method. Specifically, our research follows two steps: entity extraction and entity classification. In the entity extraction stage, we design an adaptive convolution layer that contains a position-aware module, a dilated gated convolutions (DGConv) module, and a gate module. These three modules can not only dynamically acquire position-aware head and tail representations of spans by applying two single-layer fully connection layer, but also capture relationship between close and distant words. Through the acquisition of connections at different distances between words, the information-aware layer obtains auxiliary information, while the head and tail representations are used to acquire boundaries and then incorporate relatively

necessary parts into span representations. Because span representations have different association strengths under different labels in the entity classification stage, we design the information-agnostic layer to apply the multi-head self-attention mechanism to establish the corresponding span-level correlation for each label. To avoid excessive attention to auxiliary information, we emphasize the boundaries at this layer with the use of only head and tail representations.

To prove the effectiveness of proposed model, we conducted experiments on six NER datasets, three of them are nested datasets, and the other three are flat datasets. For the nested datasets, proposed model achieves F_1 scores of 87.73, 87.23, and 81.40 on ACE2004, ACE2005 and GENIA, respectively. For the flat datasets, our model achieves F_1 scores of 97.07, 73.81, and 93.07 on Resume, Weibo and CoNLL2003, respectively. Using BERT as an encoder, proposed model outperforms the state-of-the-art methods on ACE2005, ACE2004, Resume and Weibo. And we get comparable results on the GENIA and CoNLL03. Our contributions are summarized as:

- This is the first work of using boundary and complete auxiliary information (i.e., inside tokens, labels, related spans, relative position) that is more efficient and reduces subjective interference.
- This work has no artificially set rules. The research does not require any external knowledge resources to achieve promising results. Thus it can be easily adapted to most usage scenarios for domain-specific data.
- The experiments explore that our proposed method performs better than the existing span-based methods and achieves state-of-the-art performance on ACE2005, ACE2004, Resume, and Weibo.

2 Related Work

2.1 Nested NER

Here we mainly focus on four nested NER methods: sequence-tagging methods, hypergraph-based methods, sequence-to-sequence methods, and span-based methods since they are similar to our work.

By stacking flat NER layers, sequence labeling methods can obtain nested entities. However, this leads to the error propagation problem. By Using dynamic stacking of flat decoding layers, [9] construct revised representations of the entities identified in the lower layers. Then provide identified entities to the next layer. Some people have improved this method by designing a reverse pyramid structure to achieve the reverse flow of information [23]. Others divide NER into two steps: merging entities and sequence labeling [4].

Hypergraph-based method was first proposed by [14] as a solution to the problem of nested entities. It has been further consolidated and enhanced by subsequent work [16,22]. The methods requires complex structures to deal with nested entities. The method also leads to structural errors and structural ambiguities during inference.

Span-based methods enumerate all span representations in a sentence and predict their types. The span representation of an entity can be obtained in various ways [11,19,31]. Several works have proposed the use of external knowledge

resources. Such as the introduction of machine reading comprehension (MRC) [13] and dependency relations [10] for span prediction. The span-based methods can identify entities and their corresponding types directly [11], or they can split the process of identifying entities into two stages, including entity extraction and entity classification [19,26,27]. Compared with these previous methods, our method uses the auxiliary information that the span representation possesses.

Seq2Seq methods generate various entities directly. [5] first proposed a Seq2Seq model, where the input is the original sentence and the output is the entity start position, entity length, and entity type. [29] combines the Seq2Seq model with a BART-based pointer network. There are other methods using contrast learning [33], generative adversarial networks [8] and reinforcement learning [24] for entity recognition.

3 Approach

Figure 2 shows an overview of our approach, which consists of four main layers: the encoder layer, the adaptive convolution layer, the information-aware layer, and the information-agnostic layer.

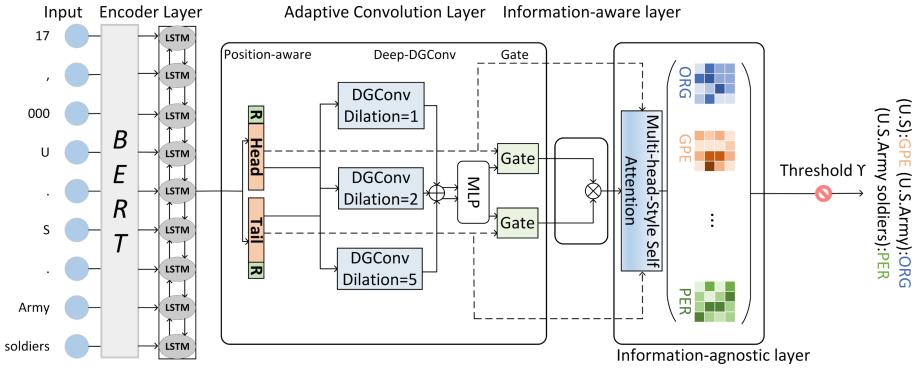

Fig. 2. The architecture of our method. MLP represents multi-layer perceptron. ⊕ and ⊗ represent concatenation and dot-product operations.

3.1 Encoder Layer

We follow [11] to encode the text. Given the input sentence $X = \{x_1, x_2, \ldots, x_N\}$ of N tokens, we first generate contextual embeddings of word pieces using BERT [3] and then combine them employing max-pooling to produce word representations. Then we adopt BiLSTM [7] to enhance the word representation. Finally, our sentence X can be represented as word representations H:

$$H = \{h_1, h_2, \ldots, h_N\} \in \mathbb{R}^{N \times d_w} \tag{1}$$

where d_w denotes the dimension of the word representation, and N is the length of the input sentence.

3.2 Adaptive Convolution Layer

Position-Aware Module. To represent the head and tail of a span, we use two single full connection layers to transform each h_i to the head and tail vector space. At this point, we obtain the head and tail representation. In addition, position plays an essential role in identifying entities [28], so we attach position embedding from [20] to the word representation:

$$h_i^s = (W_s h_i + b_1) \otimes R_i \tag{2}$$

$$h_i^t = (W_t h_i + b_2) \otimes R_i \tag{3}$$

$$H_\delta = \left\{h_1^\delta, h_2^\delta, \ldots, h_N^\delta\right\} \tag{4}$$

where $W_s, W_t \in \mathbb{R}^{d_w \times d_h}$ and $b_1, b_2 \in \mathbb{R}^{d_h}$ are trainable parameters and bias terms, respectively. R_i is the position embedding of the i-th word, $\otimes$ is the element-wise multiplication. $\delta \in \{s, t\}$. s and t are the head and tail, respectively.

DGConv Module. We feed the head and tail representation into the same convolution module, which allows the head and tail to learn each other's word representation without introducing additional parameters. For capturing the interactions between words at different distances, we use multiple DGConv with different dilation rates r(e.g.,$r \in [1, 2, 5]$). Gated convolution avoids gradient vanishing and controls information flow while these interactions form auxiliary information. The calculation of each dilated gated convolution can be expressed as:

$$\mathrm{DGConv}(H_\delta) = D_1 \otimes H_\delta + (\mathbf{1} - D_1) \otimes \phi\left(D_2\right) \tag{5}$$

$$C_\delta^r = \sigma\left(\mathrm{DGConv}\left(H_\delta\right)\right) \tag{6}$$

where D_1 and D_2 are parameter-independent 1-dimensional convolution with H_δ as input. σ and ϕ are relu and sigmoid activation functions, respectively. $\otimes$ is element-wise multiplication, and $\mathbf{1}$ is a 1-vector with its dimension matching D_1. After that, we combine the different dilatation rates of C_δ^r to get the final result $C_\delta = \left[C_\delta^1; C_\delta^2; C_\delta^5\right] \in \mathbb{R}^{N \times 3d_h}$ and feed it into the multi-layer perceptron (MLP) to reduce the dimension:

$$Q_\delta = MLP\left(C_\delta\right) \in \mathbb{R}^{N \times d_h} \tag{7}$$

Gate Module. Since the previous work [6,32] demonstrated that the boundaries are practical, we balance the word representation itself with the extracted word representation at different distances. Then we can filter the unnecessary information. The gate module is shown below:

$$r_\delta = W_1 H_\delta + W_2 Q_\delta + b \tag{8}$$

$$O_\delta = r_\delta \otimes H_\delta + (\mathbf{1} - r_\delta) \otimes Q_\delta \tag{9}$$

where H_δ and Q_δ are from Eqs. 4 and 7. $W_1, W_2 \in \mathbb{R}^{d_h \times d_h}$ and $b \in \mathbb{R}^{d_h}$ are trainable parameters and bias term, respectively. $\mathbf{1}$ is a 1-vector with its dimension matching H_δ. $\otimes$ is element-wise multiplication. Finally, we get head and tail representation:

$$S = Q_s = \{s_1, \ldots, s_N\} \in \mathbb{R}^{N \times d_h} \tag{10}$$

$$T = Q_t = \{t_1, \ldots, t_N\} \in \mathbb{R}^{N \times d_h} \tag{11}$$

3.3 Information-Aware Layer

To integrate boundaries and auxiliary information into the span representation. We obtain $\text{Span}(i, j)$ by dot product s_i^T and t_j, T is for transposition:

$$\text{Span}(i, j) = s_i^T t_j \tag{12}$$

$\text{Span}(i, j) \in \mathbb{R}^{1 \times 1}$ indicates the region of a candidate span from the i-th word to the j-th word in a sentence. Due to the filtering of the gate module [1] and the local attention of the convolution [2], the model can learn to discriminate the importance of words acquired at different distances. Thus, s_i and t_j itself will yield the boundary, close and distant words that are strongly associated with the current word pair (s_i, t_j) will be the inside tokens and related spans, respectively:

$$(A + B)^T (C + D) = A^T C + A^T D + B^T C + B^T D \tag{13}$$

Here we simplify the process and ignore the weights. As in Fig. 1, suppose the current entity $\text{Span}(i, j)$ is [*U.S.Army*], A represents U, B represents *Persian*, C represents *Army*, and D represents *Gulf*. $A + B$ represents the word representation of U that obtains *Persian* information from upper layer. $A^T D$ represents the boundary, and $B^T D$ represents the required related spans. Thus, instead of enumerating all spans, $\text{Span}(i, j)$ can obtain boundaries, inside tokens, and related spans, while the model can learn weights to adjust their importance. Additionally, the relative positions can be determined by using position embedding attached to the word representation. We take the boundary as an example:

$$\left(R_i h_i^s\right)^T \left(R_j h_j^t\right) = h_i^{sT} R_{j-i} h_j^t \tag{14}$$

where R_i and R_j are the position embeddings of the i-th and j-th words mentioned earlier (Eq. 2), related spans and inside tokens can also acquire their relative position.

3.4 Information-Agnostic Layer

Excess auxiliary information cluttering the span representation during entity classification may cause incorrect boundary predictions. So the boundaries become more significant in this layer. And in order to learn the correlation

intensities of span representation to different labels, motivated by the multi-head self-attention mechanism, we set the number of heads as the size of entity types, then apply attention operations. We denote $c_\alpha(i,j)$ as the correlation intensities of $\text{Span}(i,j)$ under the α tag and only use the boundaries to form span representation.

$$c_\alpha(i,j) = W_\alpha^T \left[h_i^s; h_j^t\right] \tag{15}$$

where $\alpha \in \{1,2,\ldots,|T|\}$, $|T|$ is the number of labels. $W_\alpha \in R^{(2\times d_h)}$ is the trainable parameters. [;] means concatenation operation. h_i^s and h_j^t are from Eq. 2 and Eq. 3. We combine the results of entity extraction and entity classification to get the final span score:

$$p_{i,j}^\alpha = \text{Span}(i,j) + c_\alpha(i,j) \tag{16}$$

3.5 Training and Inference Details

During training, we follow [21] which generalizes the softmax cross-entropy loss to multi-label classification. The method effectively solved the problem of positive and negative label imbalance. In addition, as in [31], we set a threshold γ to determine whether span belongs to label α. The loss function can be formulated as follows:

$$\mathcal{L}_\alpha = \log\left(e^\gamma + \sum_{(i,j)\in\Omega_\alpha} e^{-p_{i,j}^\alpha}\right) + \log\left(e^\gamma + \sum_{(i,j)\notin\Omega_\alpha} e^{p_{i,j}^\alpha}\right) \tag{17}$$

where Ω_α represents the set of entities span belonging to label α, γ is set to 0. Finally, we add up the loss on all labels to get the total loss:

$$\mathcal{L} = \sum_{\alpha\in\varepsilon} \mathcal{L}_\alpha \tag{18}$$

where $\varepsilon = \{1,2,...,|T|\}$, $[T]$ is the number of labels.

During inference, The span satisfying $p_{i,j}^\alpha > 0$ is the output of the entity belonging to the label α.

4 Experiments

4.1 Datasets

To evaluate the performance of our model on the two NER subtasks, we conduct experiments on six datasets.

Flat NER Datasets. We conduct experiments on the English dataset CoNLL2003 and the Chinese dataset Resume and Weibo. We employ the same experimental setting in previous work [29].

Nested NER Datasets We conducted experiments on the GENIA, ACE2005, and ACE2004. For ACE2005 and ACE2004, we used the same dataset split as [14]. For GENIA, we followed [11] using five types of entities, dividing the train/dev/test as 8.1:0.9:1.0.

Table 1. Results for flat NER datasets. † represents our re-implementation with their code.

	CoNLL2003			Weibo			Resume		
	P	R	F1	P	R	F1	P	R	F1
Span-based Methods									
Locate and Lable [19]	92.13	**93.70**	92.94	70.11	68.12	69.16	–	–	–
W2NER [11]	92.71	93.44	**93.07**	70.84	73.87	72.32	96.96	96.35	96.65
Biaffine [31] †	92.91	92.13	92.52	–	–	–	–	–	–
Baseline+BS [35]	–	–	–	70.16	**75.36**	72.66	96.63	**96.69**	96.66
Others									
TENER [28]	–	–	–	–	–	58.17	–	–	95
LSTM + Lexicon augment [15]	–	–	–	70.94	67.02	70.5	96.08	96.13	96.11
FLAT [12]	–	–	–	–	–	60.32	–	–	95.45
AESINER [18]	–	–	–	–	–	69.78	–	–	96.62
BartNER+BART [29] †	92.56	93.56	93.05	–	–	–	–	–	–
AIESNER(Ours)	**93.08**	93.06	**93.07**	**74.45**	73.19	**73.81**	**97.58**	96.56	**97.07**

Table 2. Results for nested NER datasets.

	ACE2004			ACE2005			GENIA		
	P	R	F1	P	R	F1	P	R	F1
Span-based Methods									
Briaffine [31]	87.30	86.00	86.70	85.20	85.60	85.40	81.80	79.30	80.50
A Span-based Model [10]	–	–	–	–	–	83.00	–	–	77.80
Locate and Lable [19]	87.44	87.38	87.41	86.09	87.27	86.67	80.19	80.89	80.54
Triaffine [32]	87.13	87.68	87.40	86.7	86.94	86.82	80.42	**82.06**	81.23
CNN-NER [30]	**87.82**	87.40	87.61	86.39	87.24	86.82	**83.18**	79.7	**81.40**
W2NER [11]	87.33	**87.71**	87.52	85.03	**88.62**	86.79	83.1	79.76	81.39
Others									
SH+LSTM [22]	78.00	72.40	75.10	76.80	72.30	74.50	77.00	73.30	75.10
Neural layered model [9]	–	–	–	74.20	70.30	72.20	78.50	71.30	74.70
BartNER+BART [29]	87.27	86.41	86.84	83.16	86.38	84.74	78.87	79.6	79.23
SMHSA [27]	86.90	85.80	86.30	85.70	85.20	85.40	80.30	78.90	79.60
AIESNER(ous)	**87.82**	87.64	**87.73**	**86.97**	87.49	**87.23**	81.75	81.06	**81.40**

4.2 Results for Flat NER

We evaluate our model on CoNLL03, Weibo, and Resume. As shown in Table 1, F_1 scores of our model were 93.07, 73.81 and 97.07, respectively, outperforming the representatives of other methods (+0.02 on CoNLL2003, +3.31 on Weibo, +0.45 on Resume). Compared to other span-based methods, our model achieves the best performance on the F_1 scores of Resume (+0.41 vs. baseline+BS) and Weibo (+1.14 vs. baseline+BS), reaching the state-of-the-art results and on CoNLL03 (+0.00 vs. W2NER) we also achieved competitive results. Further-

Table 3. Model ablation studies F_1. DGConv(r=1) denotes the convolution with the dilation rate 1. "-" means remove the module.

	ACE2005	Weibo	Genia
Ours	**87.23**	**73.81**	**81.40**
-Gate Module	86.74 (−0.49)	71.32 (−2.49)	80.98 **(−0.42)**
Gate replaced with Add	86.44 (**−0.79**)	70.56 (**−3.25**)	81.08 (−0.32)
DGConv replaced with DConv	86.71 (−0.52)	73.06 (−0.75)	81.05 (−0.35)
-Position Emb	86.64 (−0.59)	72.46 (−1.35)	80.33 **(−1.07)**
-DGConv	86.48 **(−0.75)**	71.38 **(−2.43)**	80.68 (−0.72)

more, our model achieves the best precision performance, demonstrating the effectiveness of the auxiliary information we incorporated.

4.3 Results for Nested NER

Table 2 shows the performance of our model on ACE2004, ACE2005, and GENIA. F_1 scores of our model were 87.73, 87.23, and 81.40, respectively, which substantially outperforms the representatives in other methods (+0.89 on ACE2004, +1.83 on ACE2005, +0.80 on GENIA), proving the advantage of span-based methods in solving nested NER. Compared with other span-based methods, our model outperforms previous state-of-the-art methods in terms of F_1 scores for ACE2004 (+0.12 vs. CNN-NER) and ACE2005 (+0.41 vs. Tri-affine). Our model also achieved competitive performance for GENIA (+0.00 vs. CNN-NER).

4.4 Ablation Study

As shown in Table 3, we ablate or replace each part of the model on ACE2005, Weibo, and GENIA. First, we remove the gate module, and the performance drop proves the importance of the boundaries. In contrast, changing the gates to direct addition would make the model unable to use the information obtained selectively. The overall performance drop is more pronounced than the weakening of the boundaries information. The model's performance drops after replacing the DGconv with the Dconv. After removing the adaptive convolution layer or position embedding, the performance of the model decreases significantly.

4.5 Case Study

To analyze the effectiveness of auxiliary information, we show two examples from the ACE2005 and GENIA datasets in Table 4. We remove the position embedding, DGConv module, and Gate module from the dynamic convolution layer to eliminate the effect of auxiliary information. In the first example, the model misclassifies "*Sukhoi Su-27*" as "None" and "*Su-27*" as "VEH" in the absence

Table 4. Case study on ACE2005 and GENIA dataset. The colored brackets indicate the boundary and label of the entity. "AUX infor" is the abbreviation for auxiliary information.

		AIESNER w/o AUX infor		AIESNER	
Span	Gold label	label	$p^{\alpha}_{i,j}$	label	$p^{\alpha}_{i,j}$
... [several squadrons of [Sukhoi Su-27]$_{\text{VEH}}$ interceptors, considered [[the world]$_{\text{LOC}}$'s premier dogfighters]$_{\text{VEH}}$]$_{\text{VEH}}$.					
several squadrons of Sukhoi Su-27 ... dogfighters	VEH	VEH	2.00	VEH	7.08
Sukhoi Su-27	VEH	None	−0.29	VEH	4.98
the world	LOC	LOC	5.95	LOC	8.12
the world's premier dogfighters	VEH	None	−1.56	VEH	2.41
Su-27	None	VEH	3.79	None	−3.89
... [octamer element]$_{\text{DNA}}$ 5-ATGCAAAG-3, located in the [upstream region]$_{\text{DNA}}$ of this [promoter]$_{\text{DNA}}$ and in the [promoters]$_{\text{DNA}}$ of ...					
octamer element	DNA	DNA	0.94	DNA	2.79
upstream region	DNA	DNA	1.21	DNA	2.19
promoter	DNA	None	−0.17	DNA	1.92
promoters	DNA	DNA	0.29	DNA	2.49

of auxiliary information. However, with the help of auxiliary information, the model corrects them to "VEH" and "None". In the second example, the model successfully corrects "*promoter*" from the "None" to the "DNA". In addition, with the help of the auxiliary information, the confidence level $p^{\alpha}_{i,j}$ of the model for the correct label can be significantly improved.

5 Conclusion

In this paper, we propose a span-based method for nested and flat NER. We argue that boundaries and auxiliary information including relative position, inside tokens, labels, and related spans, should be used reasonably to enhance span representation and classification. To this end, we design a model that automatically learns the correlation between boundaries and auxiliary information, avoiding the error and tedium of human-defined rules. Experiments show that our method outperforms all span-based methods and achieves state-of-the-art performance on four datasets.

Acknowledgement. This work was supported by the Jilin Provincial Department of Education Science and Technology Research Planning Project, Grant number jjkh20220779kj. Jilin Provincial Science and Technology Development Plan Project, Grant number 20220201149gx.

References

1. Cao, H., et al.: OneEE: a one-stage framework for fast overlapping and nested event extraction. In: Proceedings of COLING, pp. 1953–1964. International Committee on Computational Linguistics, Gyeongju, Republic of Korea (2022)
2. Cordonnier, J.B., Loukas, A., Jaggi, M.: On the relationship between self-attention and convolutional layers. arXiv preprint arXiv:1911.03584 (2019)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019)
4. Fisher, J., Vlachos, A.: Merge and label: a novel neural network architecture for nested NER. In: Proceedings of ACL, pp. 5840–5850. Association for Computational Linguistics, Florence, Italy (2019)
5. Gillick, D., Brunk, C., Vinyals, O., Subramanya, A.: Multilingual language processing from bytes. In: Proceedings of NAACL, pp. 1296–1306. Association for Computational Linguistics, San Diego, California (2016)
6. Gu, Y., Qu, X., Wang, Z., Zheng, Y., Huai, B., Yuan, N.J.: Delving deep into regularity: A simple but effective method for Chinese named entity recognition. In: Findings of NAACL, pp. 1863–1873. Association for Computational Linguistics, Seattle, United States (2022)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
8. Huang, P., Zhao, X., Hu, M., Fang, Y., Li, X., Xiao, W.: Extract-select: a span selection framework for nested named entity recognition with generative adversarial training. In: Findings of ACL, pp. 85–96. Association for Computational Linguistics, Dublin, Ireland (2022)
9. Ju, M., Miwa, M., Ananiadou, S.: A neural layered model for nested named entity recognition. In: Proceedings of NAACL, pp. 1446–1459. Association for Computational Linguistics, New Orleans, Louisiana (2018)
10. Li, F., Lin, Z., Zhang, M., Ji, D.: A span-based model for joint overlapped and discontinuous named entity recognition. In: Proceedings of ACL-IJCNLP, pp. 4814–4828. Association for Computational Linguistics, Online (2021)
11. Li, J., et al.: Unified named entity recognition as word-word relation classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 10965–10973 (2022)
12. Li, X., Yan, H., Qiu, X., Huang, X.: FLAT: Chinese NER using flat-lattice transformer. In: Proceedings of ACL, pp. 6836–6842. Association for Computational Linguistics, Online (2020)
13. Li, X., Feng, J., Meng, Y., Han, Q., Wu, F., Li, J.: A unified MRC framework for named entity recognition. In: Proceedings of ACL, pp. 5849–5859. Association for Computational Linguistics, Online (2020)
14. Lu, W., Roth, D.: Joint mention extraction and classification with mention hypergraphs. In: Proceedings of EMNLP, pp. 857–867. Association for Computational Linguistics, Lisbon, Portugal (2015)
15. Ma, R., Peng, M., Zhang, Q., Wei, Z., Huang, X.: Simplify the usage of lexicon in Chinese NER. In: Proceedings of ACL, pp. 5951–5960. Association for Computational Linguistics, Online (2020)
16. Muis, A.O., Lu, W.: Learning to recognize discontiguous entities. In: Proceedings of EMNLP, pp. 75–84. Association for Computational Linguistics, Austin, Texas (2016)

17. Muis, A.O., Lu, W.: Labeling gaps between words: recognizing overlapping mentions with mention separators. In: Proceedings of EMNLP, pp. 2608–2618. Association for Computational Linguistics, Copenhagen, Denmark (2017)
18. Nie, Y., Tian, Y., Song, Y., Ao, X., Wan, X.: Improving named entity recognition with attentive ensemble of syntactic information. In: Findings of EMNLP, pp. 4231–4245. Association for Computational Linguistics, Online (2020)
19. Shen, Y., Ma, X., Tan, Z., Zhang, S., Wang, W., Lu, W.: Locate and label: a two-stage identifier for nested named entity recognition. In: Proceedings of ACL-IJCNLP, pp. 2782–2794. Association for Computational Linguistics, Online (2021)
20. Su, J., Lu, Y., Pan, S., Wen, B., Liu, Y.: Roformer: enhanced transformer with rotary position embedding. arXiv preprint arXiv:2104.09864 (2021)
21. Sun, Y., et al.: Circle loss: a unified perspective of pair similarity optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6398–6407 (2020)
22. Wang, B., Lu, W.: Neural segmental hypergraphs for overlapping mention recognition. In: Proceedings of EMNLP, pp. 204–214. Association for Computational Linguistics, Brussels, Belgium (2018)
23. Wang, J., Shou, L., Chen, K., Chen, G.: Pyramid: a layered model for nested named entity recognition. In: Proceedings of ACL, pp. 5918–5928. Association for Computational Linguistics, Online (2020)
24. Wang, X., et al.: Automated concatenation of embeddings for structured prediction. In: Proceedings of ACL-IJCNLP, pp. 2643–2660. Association for Computational Linguistics, Online (2021)
25. Wang, X., et al.: Improving named entity recognition by external context retrieving and cooperative learning. In: Proceedings of ACL, pp. 1800–1812. Association for Computational Linguistics, Online (2021)
26. Wang, Y., Yu, B., Zhu, H., Liu, T., Yu, N., Sun, L.: Discontinuous named entity recognition as maximal clique discovery. In: Proceedings ACL-IJCNLP, pp. 764–774. Association for Computational Linguistics, Online (2021)
27. Xu, Y., Huang, H., Feng, C., Hu, Y.: A supervised multi-head self-attention network for nested named entity recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 14185–14193 (2021)
28. Yan, H., Deng, B., Li, X., Qiu, X.: Tener: adapting transformer encoder for named entity recognition. arXiv preprint arXiv:1911.04474 (2019)
29. Yan, H., Gui, T., Dai, J., Guo, Q., Zhang, Z., Qiu, X.: A unified generative framework for various NER subtasks. In: Proceedings of ACL-IJCNLP, pp. 5808–5822. Association for Computational Linguistics, Online (2021)
30. Yan, H., Sun, Y., Li, X., Qiu, X.: An embarrassingly easy but strong baseline for nested named entity recognition. arXiv preprint arXiv:2208.04534 (2022), https://arxiv.53yu.com/pdf/2208.04534
31. Yu, J., Bohnet, B., Poesio, M.: Named entity recognition as dependency parsing. In: Proceedings of ACL, pp. 6470–6476. Association for Computational Linguistics, Online (2020)
32. Yuan, Z., Tan, C., Huang, S., Huang, F.: Fusing heterogeneous factors with triaffine mechanism for nested named entity recognition. In: Findings of ACL, pp. 3174–3186. Association for Computational Linguistics, Dublin, Ireland (2022)
33. Zhang, S., Cheng, H., Gao, J., Poon, H.: Optimizing bi-encoder for named entity recognition via contrastive learning. arXiv preprint arXiv:2208.14565 (2022)

34. Zhang, Y., Yang, J.: Chinese NER using lattice LSTM. In: Proceedings of ACL, pp. 1554–1564. Association for Computational Linguistics, Melbourne, Australia (2018)
35. Zhu, E., Li, J.: Boundary smoothing for named entity recognition. In: Proceedings of ACL, pp. 7096–7108. Association for Computational Linguistics, Dublin, Ireland (2022)