Chinese Named Entity Recognition Method Based on Lexicon and Convolution-Integrated Self-Attention

Wenjie Xu* LEOXUWJ@163.COM
College of Information Engineering, Shanghai Maritime University, Shanghai, 201306, China

Maoting Gao
College of Information Engineering, Shanghai Maritime University, Shanghai, 201306, China

Editors: Nianyin Zeng and Ram Bilas Pachori

Abstract

Chinese Named Entity Recognition methods primarily consist of span-based approaches and sequence-to-sequence methods. However, the former focuses solely on the recognition of entity boundaries, while the latter is susceptible to exposure bias. To address these issues, a Chinese NER method based on dictionary enhancement and self-attention fusion convolution is proposed. Initially, the text is encoded using the pre-trained model ERNIE 3.0 and lexical representations. Then, Bi-LSTM is utilized to further capture the contextual information of the sequence, resulting in the final character representations. Subsequently, a two-dimensional (2D) grid is constructed for modeling character pairs, and a feature integration layer is developed by merging self-attention mechanisms and convolution to refine and capture the interactions between characters. Finally, a joint predictor composed of a dual-affine classifier and multilayer perceptrons is used to predict entity categories. Experimental results demonstrate that this method can effectively recognize both flat and nested named entities. Compared to the current best-performing baseline models, the proposed method achieves an increase of 0.14% and 2.53% in F1 scores on the flat datasets Resume and Weibo, respectively, and an improvement of 0.52% in F1 score on the nested dataset ACE2005.

Keywords: Unified Named Entity Recognition, Lexicon, Self-Attention, Convolution

1. Introduction

With the surge in data volume, manually extracting valuable information from massive texts is undoubtedly a time-consuming and laborious task. Hence, the research on information extraction methods has gradually emerged. Among these, Named Entity Recognition (NER) as one of the core technologies, has been receiving high attention from both the academic and industrial communities for many years. Furthermore, NER also forms the basis for a variety of NLP applications, including but not limited to entity relationship extraction (Saxena et al., 2022), the construction of knowledge graphs (Finkel and Manning, 2009), and intelligent question answering systems (Ju et al., 2018).

Chinese Named Entity Recognition mainly focuses on two types of entities: flat named entities and nested named entities (Li et al., 2021). Most work on NER deals only with flat named entities, yet spans of entities often overlap (Zhang and Yang, 2018), causing the loss of potentially important information in recognition tasks and adversely affecting subsequent tasks (Gui et al., 2019a).

Traditionally, named entity recognition was studied separately for these two types. Recently, some research has focused on unified named entity recognition methods, aiming to use a single model to handle both types of NER tasks simultaneously (Gui et al., 2019b).

Therefore, a method based on dictionary enhancement and self-attention fusion convolution is proposed. It starts with ERNIE 3.0 concatenating lexical representations and uses Bi-LSTM to obtain the final character representations with contextual semantics. Then, a two-dimensional (2D) grid is constructed to model character pairs; subsequently, to capture the interactions between near and distant character pairs more finely, a feature integration layer that combines self-attention mechanisms and convolution is built to refine and capture the interactions between characters. Finally, a joint predictor consisting of a dual-affine classifier and multilayer perceptrons is used to predict entity categories.

2. Related Work

In Chinese NER tasks, it's common to use character-based rather than lexicon-based models because inaccurate word segmentation might degrade the model's performance. Integrating accurate lexicon information into character-based NER models helps to delineate entity boundaries more precisely. Zhang and others (Li et al., 2020) proposed the Lattice-LSTM model, which enhances the accuracy of Chinese NER by utilizing lexicon information through constructing a directed acyclic graph of characters and potential words from a dictionary to maximize the use of lexical boundary information. However, it introduces new noise and potential word conflicts. To address the potential word conflicts in the Lattice structure, Gui and others (Ma et al., 2020) introduced the LR-CNN model, which employs CNN for feature extraction and incorporates attention mechanisms at various layers to integrate equal-length word information. By designing new relative position encodings and improving the original Transformer's absolute position encoding, Li and others (Shen et al., 2021) introduced the FLAT model based on the Transformer architecture. Ma and others proposed the SoftLexicon model to implement the idea behind Lattice-LSTM in a simpler manner. By matching sentences with dictionaries to find all words containing each character, categorized into four types and mapped to corresponding category vectors.

3. Model Architecture

The model architecture is illustrated in Figure 1.

3.1. Embedding Layer

For character-based Chinese NER models, the input sentence s is viewed as a sequence of characters $S = \{c_1, c_2, \ldots, c_n\}$, where $c_i \in C$, and C is the set of all Chinese characters, with n being the length of the sentence s.

Character Vector. ERNIE improves upon BERT's pre-training tasks by masking complete words, phrases, and named entities, allowing the language model to better capture global information. By inputting each character c_i into the ERNIE model, we obtain the character vector S^e corresponding to the input sequence S.

$$S^e = ERNIE\,(S) = \{c_1^e, c_2^e, \ldots, c_n^e\} \tag{1}$$

Here, c_i^e represents the character vector of the i^{th} character after being encoded by ERNIE.

Word Vector. In Chinese NER, models based on the character level are commonly used. To preserve the segmentation information that the lexicon provides to each character, the words matched to each character c_i are divided into 4 categories based on the character's position within the word. For each character c_i in the input sequence $S = \{c_1, c_2, \ldots, c_n\}$, the definitions of the 4 category

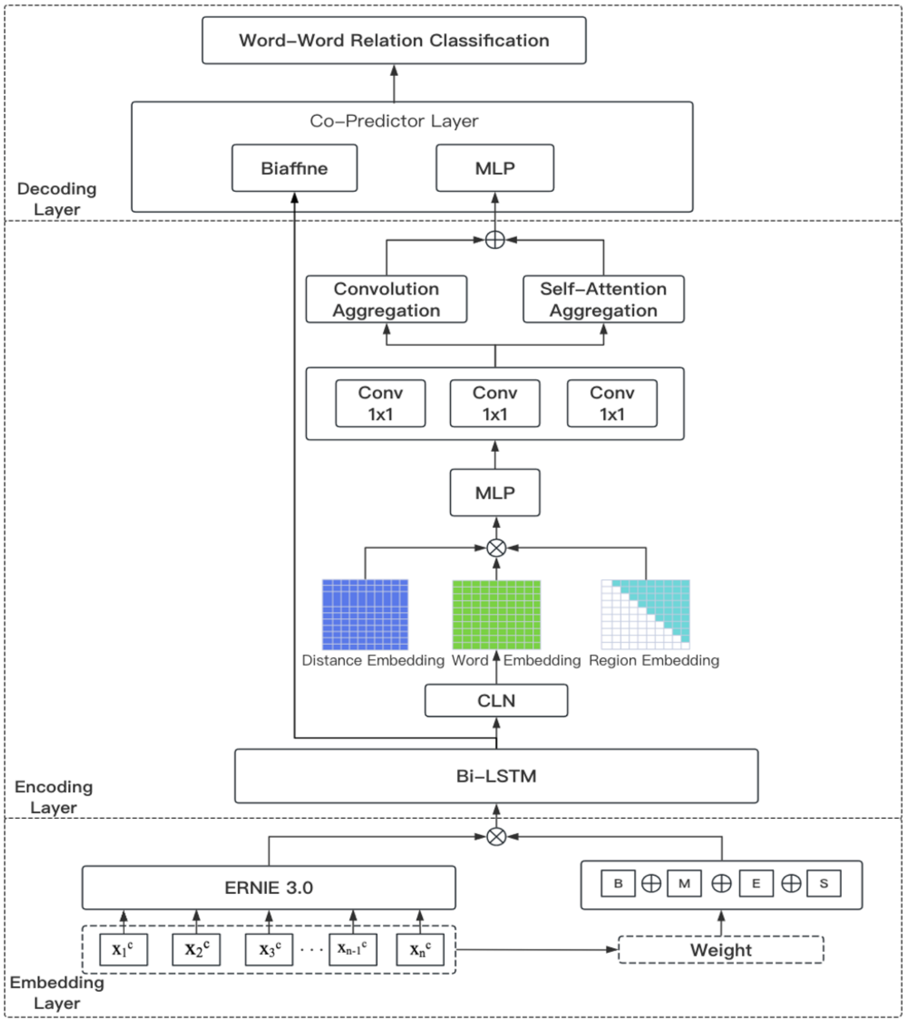

Figure 1: Model framework.

labels are shown in Equation 2.

$$\begin{gathered}
B\left(c_i\right)=\left\{w_{i,m}, \forall w_{i,m} \in D, i<m \leq n\right\} \\
M\left(c_i\right)=\left\{w_{k,m}, \forall w_{k,m} \in D, 1 \leq k<i<m \leq n\right\} \\
E\left(c_i\right)=\left\{w_{k,i}, \forall w_{k,i} \in D, 1 \leq k<i\right\} \\
S\left(c_i\right)=\left\{c_i, \exists c_i \in D\right\}
\end{gathered} \tag{2}$$

Here, w refers to the words matched in the sequence S, and D represents the dictionary used by the model.

To effectively utilize the lexical information of each position and to accelerate the computation of lexical weights through dictionaries available offline, the frequency of the lexicon's occurrence in the dictionary is used as the weight. The specific calculations are shown in Equations 3 and 4.

$$F=\sum_{w \in B \cup M \cup E \cup S} f(w) \tag{3}$$

$$h\left(G\right)=\frac{4}{F}\sum_{w\in G}f\left(w\right)e^{w}\left(w\right) \tag{4}$$

Here, $f(w)$ is the frequency of the lexicon word w matched from the dictionary across the entire dictionary.

To preserve as much lexical information as possible, the word vectors from the four sets are merged into a single word vector with a fixed dimension. This combined vector, which integrates the B, M, E, S information, is added to each character vector. Consequently, the representation of each character vector is as shown in Equation 5.

$$\begin{gathered} e^{s}\left(B,M,E,S\right)=\left[h\left(B\right);h\left(M\right);h\left(E\right);h\left(S\right)\right] \\ c^{e}\leftarrow\left[c^{e};e^{s}\left(B,M,E,S\right)\right] \end{gathered} \tag{5}$$

Here, h is the weighting function from Equation 5.

3.2. Encoding Layer

Grid Representation Construction. The character pair grid representation is denoted by a three-dimensional matrix $E^{w}\in R^{N\times N\times d_{h}}$, where V_{ij} represents the representation of characters c_i and c_j in the character pair (c_i,c_j).

To further enrich the representation of character pairs, a grid representation modeling approach similar to the BERT philosophy is utilized. The tensor $E^{w}\in R^{N\times N\times d_{h}}$ represents the information of all characters in a sentence. The tensor $E^{d}\in R^{N\times N\times d_{E_d}}$ represents the relative positional information. The tensor $E^{r}\in R^{N\times N\times d_{E_r}}$ represents the region information used to distinguish between the upper and lower triangular areas of the grid. The final grid representation is $G\in R^{N\times N\times d_{G}}$. The overall process of grid representation construction is illustrated in Equation 6.

$$G=\mathrm{MLP}_{1}\left(\left[E^{w};E^{d};E^{r}\right]\right) \tag{6}$$

Integration of Self-Attention and Convolution. To predict the relationships between character pairs within a two-dimensional grid and capture the interactions across varying distances, the ACmix module is employed. This module benefits from the inductive biases provided by CNNs while gaining the flexibility of the self-attention mechanism. The implementation details are as follows.

First, decompose the standard convolution operation into two stages for remodeling. The first stage involves projecting the input feature map linearly using the kernel weights at a specific position (m,n), similar to a standard 1×1 convolution.

$$\tilde{t}_{ij}^{(m,n)}=K_{m,n}s_{ij} \tag{7}$$

In the second stage, the projected feature map is shifted according to the kernel position and then aggregated together.

$$t_{ij}^{(m,n)}=\mathrm{Shift}\left(\tilde{t}_{ij}^{(m,n)},m-k/2,n-k/2\right) \tag{8}$$

$$t_{ij}=\sum_{m,n}t_{ij}^{(m,n)} \tag{9}$$

The self-attention mechanism is decomposed into two stages too. The first stage involves performing a 1×1 convolution to project the input features into queries, keys, and values.

$$q_{ij} = W_q s_{ij}, k_{ij} = W_k s_{ij}, v_{ij} = W_v s_{ij} \tag{10}$$

The second stage involves calculating the attention weights and aggregating the value matrices.

$$t_{ij} = \overset{N}{\underset{l=1}{\|}} \left(\sum_{a,b \in P_k(i,j)} A\left(q_{ij}, k_{ab}\right) v_{ab} \right) \tag{11}$$

Finally, the outputs of the two paths are added together, controlled by two learnable scalars.

$$F_{\text{out}} = \alpha F_{\text{att}} + \beta F_{\text{conv}} \tag{12}$$

3.3. Decoding Layer

The word-pair grid representation F_ij based on the output of the self-attention and convolution fusion module is then used with an MLP to compute the relationship score $y_{ij}^{''} \in \mathbb{R}^{|\mathcal{R}|}$ for the word pair (x_i, x_j).

$$y_{ij}^{''} = \text{MLP}_2\left(F_{ij}\right) \tag{13}$$

Finally, the final relationship probability y_{ij} for the word pair (x_i, x_j) is calculated by combining the scores from the Biaffine and MLP predictors.

$$y_{ij} = \text{Softmax}\left(y_{ij}^{'} + y_{ij}^{''}\right) \tag{14}$$

4. Experiments and Analysis

4.1. Experimental Settings

Dataset. The experimental datasets consist of public Chinese flat datasets Resume and Weibo, as well as the public Chinese nested dataset ACE2005. The Resume dataset is curated from summaries of resumes of senior management personnel from listed companies on Sina Finance website. The ACE2005 dataset is released by the Linguistic Data Consortium (LDC) and consists of various types of data annotated with entities, relations, and events.

Evaluation Metrics. In the experiments, precision, recall and F1-score are used as performance evaluation metrics.

$$P = \frac{TP}{(TP\ +\ FP)} \times 100\% \tag{15}$$

$$R = \frac{TP}{(TP\ +\ FN)} \times 100\% \tag{16}$$

$$F1 = \frac{2 \times P \times R}{P + R} \times 100\% \tag{17}$$

Table 1: Different Algorithms' Performance Metrics on Flat and Nested Named Entity Datasets.

	Resume			Weibo			ACE2005		
	P	R	F1	P	R	F1	P	R	F1
Lattice (Li et al., 2020)	94.81	94.11	94.46	53.04	62.25	58.79	-	-	-
LGN (Sun et al., 2019)	95.28	95.46	95.37	-	-	68.55	-	-	-
FLAT (Shen et al., 2021)	-	-	95.45	78.92	62.09	69.50	-	-	-
SoftLexicon (Gui et al., 2019b)	96.08	96.13	96.11	70.94	67.02	70.50	-	-	-
De-Bias (Zhang and Yang, 2018)	-	-	-	-	-	-	82.92	87.05	84.93
Span-graph (Gui et al., 2019a)	-	-	-	-	-	-	84.37	85.87	85.11
Locate-label (Finkel and Manning, 2009)	-	-	-	-	-	-	86.09	87.27	86.67
W2NER (Li et al., 2021)	96.85	96.13	96.49	76.20	64.98	70.14	85.91	88.35	87.11
Ours	**96.89**	**96.76**	**96.63**	**76.24**	**69.42**	**72.67**	**86.67**	**88.40**	**87.63**

4.2. Experimental Results

From Table 1, it can be seen that W2NER achieves the highest F1 scores of 96.49% on the flat named entity Resume dataset and the second highest of 70.14% on the Weibo dataset. It also scores the highest on the nested named entity ACE2005 dataset with an F1 score of 87.11%. Compared to the W2NER, the model discussed in this paper leverages ERNIE to enhance the integration of Chinese entity-level information and further strengthens entity boundary information by incorporating a Lexicon. Additionally, it utilizes ACmix to capture interactions between characters within both non-local and local scopes. This leads to a 0.04% increase in precision on both the Resume and Weibo datasets for flat named entities, a recall increase of 0.63% and 4.44% respectively, and an F1 score improvement of 0.14% and 2.53% respectively. For the nested named entity ACE2005 dataset, precision improved by 0.76%, recall by 0.05%, and F1 score by 0.52%.

5. Conclusion

This paper introduces a unified model that can simultaneously address both flat and nested NER challenges. Initially, by integrating the knowledge-enhanced, multi-paradigm unified Chinese pretraining framework ERNIE 3.0 into the encoding layer, the model obtains higher-quality dynamic Chinese character embeddings. Subsequently, the introduction of Lexicon enhances the entity boundary information within the character embeddings. In the encoding layer, ACmix leverages the complementary properties of convolution and self-attention mechanisms to further capture the relationships between characters at various distances in the character grid. This enhances the model's performance in recognizing both flat and nested named entities. Experimental results show that this model outperforms current advanced models in terms of recognition accuracy, recall rate, and F1 score. However, the model's increased structural complexity and parameter count lead to slower inference speeds and higher resource requirements for deployment. The next step will involve attempting to transfer knowledge to a lighter model through knowledge distillation techniques, enabling the model to run quickly on limited computational resources while maintaining high accuracy.

References

Jenny Rose Finkel and Christopher D. Manning. Nested named entity recognition. In Philipp Koehn and Rada Mihalcea, editors, *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 141–150, Singapore, August 2009. Association for Computational Linguistics.

Tao Gui, Ruotian Ma, Lujun Zhao, Yu-Gang Jiang, and Xuanjing Huang. Cnn-based chinese ner with lexicon rethinking. pages 4982–4988, 08 2019a. doi: 10.24963/ijcai.2019/692.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuanjing Huang. A lexicon-based graph neural network for Chinese NER. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 1040–1050, Hong Kong, China, November 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-1096.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. A neural layered model for nested named entity recognition. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1131.

Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. Unified named entity recognition as word-word relation classification, 2021.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. FLAT: Chinese NER using flat-lattice transformer. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6836–6842, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.611.

Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuanjing Huang. Simplify the usage of lexicon in Chinese NER. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5951–5960, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.528.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. Sequence-to-sequence knowledge graph completion and question answering, 2022.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. Locate and label: A two-stage identifier for nested named entity recognition. In Chengqing Zong, Fei Xia, Wenjie

Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2782–2794, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.216.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration, 2019.

Yue Zhang and Jie Yang. Chinese NER using lattice LSTM. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1144.