Dictionary-Assisted Chinese Nested Named Entity Recognition

Ye Wang, Tongtong Ding, and Lijie Li(✉)

College of Computer Science and Technology, Harbin Engineering University, Harbin, China
{wangye2020,dingtongtong,lilijie}@hrbeu.edu.cn

Abstract. Chinese Named Entity Recognition is a challenging task, made even more difficult by the presence of nested entity structures. Previous work on Nested Named Entity Recognition focused only on exploiting internal contextual information, while ignoring the use of external information. In this paper, we propose a dictionary-assisted Chinese Nested Named Entity Recognition model, called KBCNER. Our model uses the dictionary to obtain matching words, combines characters and phrases into character-phrase pairs, and integrates them into BERT. By doing so, we can extract richer semantic information from Chinese phrases than from a single character. The use of external information from the dictionary enhances the features of our model and obtains richer semantics. To avoid constraints from specific-length enumerations, we use bi-affine structures to obtain a global view of spans. We also model local interactions between spans using a Convolutional Neural Network (CNN), taking advantage of the spatial correlation between adjacent spans. Finally, we adopt the idea of contrastive learning based on R-drop to enhance the model's robustness. Experimental results demonstrate that our model achieves excellent performance on multiple datasets. By introducing external information, we improve the performance of the model, highlighting the significance of external information for Chinese Nested Named Entity Recognition.

Keywords: Chinese Nested Named Entity Recognition · Dictionary Assistance · Bi-affine Structure · Convolutional Neural Network

1 Introduction

Nested Named Entity Recognition (NNER) refers to the simultaneous recognition of multiple nested levels of named entities in text. For example, in Fig. 1 "哈尔滨 医科大学附属第一医院(First Affiliated Hospital of Harbin Medical University)", it contains three entities: "哈尔滨(Harbin)" belongs to LOC entity, "哈尔滨医科 大学(Harbin Medical University)" belongs to ORG entity, "哈尔滨医科大学附属第 一 医 院(First Affiliated Hospital of Harbin Medical University)" belongs to ORG entity. They overlap each other and are nested entities. NNER has a wide range of applications in information extraction, question answering systems, natural language understanding and other fields. However, most current NNER research focuses on English corpora, while relatively few

M. Zhang et al. (Eds.): CCF NCCA 2023, CCIS 1959, pp. 189–205, 2024.
https://doi.org/10.1007/978-981-99-8764-1_15

studies on Chinese corpora. There are some notable differences between Chinese and English Nested Named Entity Recognition. First of all, the vocabulary structure of Chinese and English is different from that of English. The fundamental unit of composition in Chinese is characters, while in English, it is letters. Therefore, identifying nested named entities in Chinese needs to take into account more complex language structures and features, such as polyphonic characters, ambiguous words, word order, etc. This also makes Chinese Nested Named Entity Recognition tasks more challenging than English. Secondly, Chinese Named Entity Recognition often needs to solve ambiguity problems, because words in Chinese often have many different meanings, and contextual information and context need to be considered to determine the correct entity type. Compared to English, the task of named entity recognition in Chinese is more challenging due to the presence of more ambiguities and context dependencies. Therefore, it is more crucial to incorporate external information for Chinese Nested Named Entity Recognition than for English Nested Named Entity Recognition.

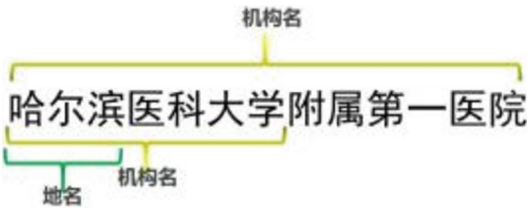

Fig. 1. Nested Entity Structure Example

Integrating external knowledge has been shown to be effective in various natural language processing tasks, such as text classification [1], semantic matching [2], text generation [3], and Named Entity Recognition [4].Among these, the dictionary- enhanced approach has demonstrated a notable improvement in Chinese Named Entity Recognition task. BERT is based on character-level granularity for Chinese and cannot capture the overall information of multi-word words, which hinders the identification of entity boundaries. Therefore, incorporating additional lexical knowledge is crucial for improving the accuracy of named entity recognition. Existing methods, such as Lattice LSTM [5] and FLAT [6], input extra vocabulary information alongside the sentence sequence into BERT and apply a specific attention mechanism to calculate them separately. However, this approach results in longer sequences, increasing computation time and memory consumption and introducing noise to the semantic representation. Recently, Liu et al. [7] proposed LEBERT, a model that integrates external dictionary information into the middle layer of BERT as an additional module, achieving promising results. In this study, we aim to apply the LEBERT dictionary-enhanced approach to Chinese Nested Named Entity Recognition to improve its performance.

In the field of Nested Named Entity Recognition, span enumeration is one of the prevailing approaches. Sohrab and Miwa et al. [8] have proposed a method of exhaustively enumerating all possible spans up to a specified length by connecting the output of start and end position LSTMs, which is then used to calculate the score for each span. To overcome the limitation of length in predicting entities, a bi-affine based structural model

is employed. By constructing the token-token table in parallel, the bi-affine decoder generates a global view of the sentence, including vector representations of all possible spans, thus improving efficiency. This approach has been demonstrated as effective in various works, such as Dozat and Manning (2017) [9] and Yu et al. (2020) [10]. In recent advances in Nested Named Entity Recognition research, Hang Yan et al. [11] treated the feature matrix as a view and utilized CNNs to model the spatial relationships between adjacent spans in the scoring matrix, which resulted in significant improvements in the task performance.

To improve the performance of Chinese Nested Named Entity Recognition, this paper proposes a dictionary-assisted method to capture richer semantics. The model constructs character-word pairs by using matching phrases obtained from a wiki dictionary and integrates them into the middle layer of BERT, fully utilizing its representational capacity. Chinese phrases contain richer semantic information than single characters, and introducing dictionary information enhances the feature richness. The model uses a bi-affine structure to obtain a global view of the span, avoiding the limitation of specific length enumeration. Additionally, the local interaction between spans is modeled using a Convolutional Neural Network (CNN) to capture spatial correlation between adjacent spans. Finally, the model's robustness is enhanced using the R-drop based contrastive learning approach. The model in this chapter aims to optimize the characteristics of the Chinese language, improve the accuracy and efficiency of Chinese Nested Named Entity Recognition.

The main contributions of this work are as follows:

1) Proposing a simple and effective model for Chinese nested named entity recognition, aimed at improving the accuracy and efficiency of the task.
2) Considering that phrases can provide richer semantics and better handle Chinese nested entity structures, integrating dictionary information into BERT to achieve deep lexical knowledge fusion. Applying the idea of contrastive learning based on R-drop to enhance the robustness and generalization ability of the model, while reducing overfitting.
3) Evaluating and validating the proposed method on both Chinese flat and nested datasets, and comparing it with baseline models, achieving the best results.

2 Related Work

Currently, methods for Nested Named Entity Recognition can be classified into four main categories: 1) Improved sequence labeling framework: through the design of a trade-off scheme, the sequence labeling task is capable of handling nested named entities; 2) Hypergraph-based methods: by utilizing a hypergraph structure, nested structures can be effectively addressed; 3) Parsing tree-based methods: similar to Constituency Parsing tree structures, they are used in nested named entity recognition; 4) Span-based methods: candidate spans are first exhaustively enumerated, and then assigned a corresponding category.

2.1 Improved Sequence Labeling Framework

Traditional sequence labeling methods, such as Hidden Markov Models and Conditional Random Fields, are usually inadequate for dealing with nested named entities. However, the improved sequence labeling method can handle nested named entities by introducing additional features and constraints. In 2018, Ju et al. [13] proposed a dynamic stacking plane NER method, which treats each plane NER as a single-layer sequence labeling, to extract entities from the inside to the outside. However, this approach is prone to error propagation. To model multiple named entity labels, Strakova et al. [14] proposed a linearized encoding scheme that combines all categories that may co-occur in pairs to generate new labels (e.g., combining B-Location with B-Organization to construct a new label B-Loc | Org). Shibuya et al. [15] provided a sub-optimal path solution that treats the label sequence of nested entities as the second-best path within the span of their parent entities, extracting entities from outside to inside. To identify nested named entities from bottom to top, Li et al. [16] proposed a Chinese NER model based on a self-attention aggregation mechanism, which connects a series of sub-models of multi-layer sequence labeling. Wang et al. [17] designed a pyramid framework to recognize nested entities. The improved sequence labeling method is straightforward and convenient to use, but it is not sufficiently accurate for modeling the nesting relationship.

2.2 Hypergraph-Based Approaches

A hypergraph is a graphical structure in which a node can be associated with multiple edges, and it can be used to model the nested structure of a sentence where each entity is a node and the nested relationship between entities is an edge. Hypergraph-based methods aim to better capture dependencies between entities by using hypergraphs. These methods typically transform the reasoning problem on hypergraphs into an integer linear programming problem. Lu et al. [18] proposed a joint entity extraction and classification model for nested NER that can effectively capture nested entities with infinite lengths. Katiyar et al. [19] extracted a hypergraph representation from an RNN and trained the model using greedy search. Wang et al. [20] proposed a piecewise hypergraph representation that avoids structural ambiguity. Luo et al. [21] proposed a bipartite planar graph structure that uses a planar NER module for the outermost entity and a graph module for all entities located in the inner layer to perform two-way information interaction between layers. Although hypergraph-based methods can explicitly capture nested entities, they require skillful hypergraph design to handle complex reasoning problems and may result in long running times.

2.3 Parsing Tree-Based Methods

Parsing tree-based methods use a tree-based algorithm to analyze the relationship between nested entities, similar to the constituency parsing tree structure used in syntactic analysis. A parsing tree is constructed in a bottom-up or top-down manner, and different features can be used for classification. In 2009, Finkel et al. [22] proposed converting a sentence into a constituency tree, with each entity corresponding to a phrase in the tree and a root node connecting the entire sentence. Fu et al. [23] proposed regarding

nested NER as the constituency resolution of the tree using local observations, with the entity spans of all markers as the nodes observed in the constituency tree, and other spans as potential nodes. Lou et al. [24] improved the method proposed by Fu by using a two-stage strategy and head-aware loss, which effectively utilized the effective information of entity heads. Yang et al. [25] proposed a new pointer network for the bottom-up analysis of nested NER and constituency resolution. Parsing tree-based methods can accurately capture nested relationships, but require more computing resources.

2.4 Span-Based Methods

Span-based methods are among the most widely used approaches for nested NER. These methods involve enumerating all potential spans in a sentence and classifying each one. While some approaches exhaustively list all possible spans, such as Sohrab et al.'s method [25], this is computationally intensive. Other approaches, like Lin et al.'s [26], first locate an anchor word and then match the entire span for classification, but this approach only works for specific structures. Xia et al. [27] proposed a multi- granularity NER method that includes a detector for entity locations and a classifier for entity types. The boundary-aware model proposed by Zheng et al. [28] uses sequence labeling to determine span boundaries before classification. Yu et al. [29] applied the bi-affine model to nested NER, pinpointing spans and scoring each one using start and end markers. Xu et al. [30] proposed a supervised multi-head self- attention mechanism, where each head identifies a category and uses a boundary detection module as an auxiliary task. Finally, Shen et al. [31] developed a two-stage method that generates candidate spans by filtering and boundary regression of seed spans before marking the corresponding category.

3 Model

This paper proposes a KBCNNER model. As shown in Fig. 2, the model is divided into three parts: the first part is the dictionary information introduction module, which integrates the matched character-word pairs information into the BERT intermediate layer; the second part is the bi-affine decoder layer, which obtains the global view of the sentence; and the third part is the Convolutional Neural Network (CNN) layer, which models the relationship between adjacent spans using CNN.

3.1 Import Dictionary Information

Define the input as a Chinese sentence, $S = \{c_1, c_2, \ldots, c_n\}$, where n represents the number of characters in the sentence. Next, two parts of the operation are performed on the input sentence S at the same time, one part is to use the BERT embedding layer to extract the vector representation of each character, and get $E = \{e_1, e_2, \ldots, e_n\}$, and then input E into the Transformer encoder for the following calculation:

$$G = \mathrm{LN}(H^{l-1} + \mathrm{MHAttn}(H^{l-1})) \tag{1}$$

$$H^l = \mathrm{LN}(G + \mathrm{FFN}(G)) \tag{2}$$

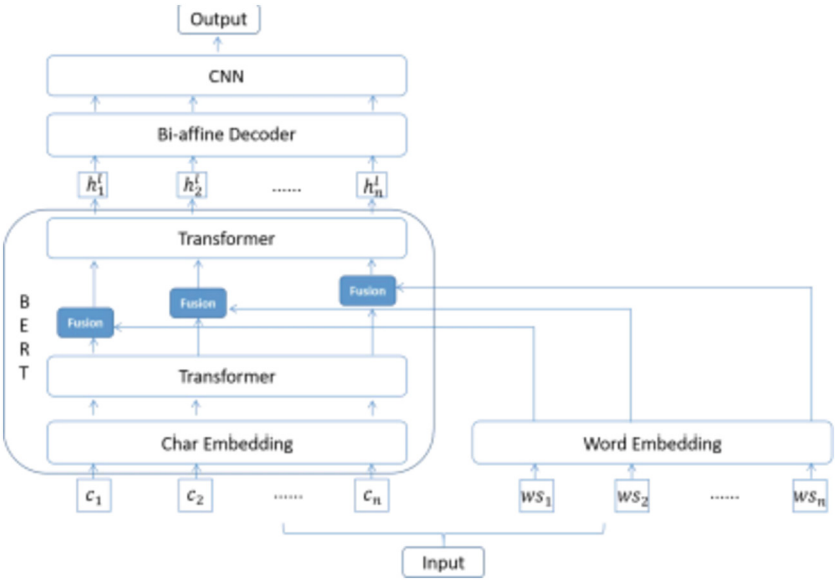

Fig. 2. KBCNNER model diagram

where $H^l = \{h_1^l, h_n^l, \ldots, h_n^l\}$, represents the output of the first layer of Transformer.

LN is a normalization operation, MHAttn is a multi-head attention mechanism, and FFN is a two-layer feedforward neural network using Relu as the activation function.

In the other part, the sentence S is matched with the dictionary D to construct character-word pairs, where the dictionary D is prepared in advance, as shown in Fig. 3. The specific method is as follows: First, build a dictionary tree Trie based on the dictionary D, then we iterate through all possible character subsequences in the input sentence and match them against the Trie tree, resulting in a list of potential words. For example, the sentence "中国人民" can be matched to "中国", " 中国人", "国人", "人民". For

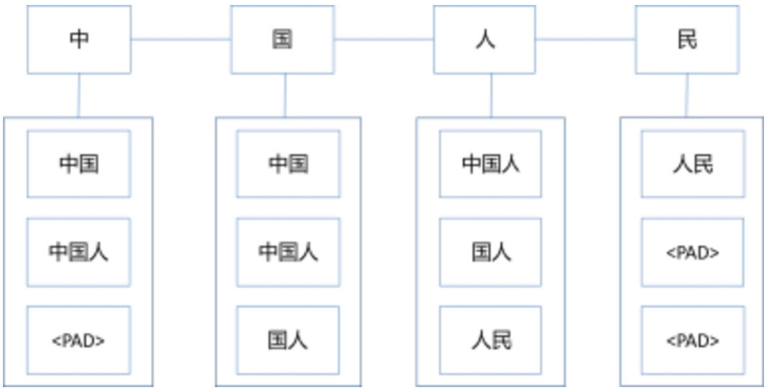

Fig. 3. Character-words pair

each matched word, assign the characters it contains. For example, the matched word "中国" is assigned to the characters "中" and "国". We pair each character with the matching word to form a character-word pair $s_{CW} = \{(c_1, ws_1), \ldots (c_i, ws_i), (c_n, ws_n)\}$ where c_i denotes the i−th character in the sentence and wsi denotes matched words assigned to ci.

Next, use the Fusion module in Fig. 4 to inject vocabulary information into BERT, and the input of Fusion is a character−word pair (h_i^c, x_i^{ws}),where h_i^c is a character vector, the output of a certain transformer layer in BERT, and $x_i^{ws} = \left\{x_{i1}^w, x_{i2}^w, \ldots x_{im}^w\right\}$ is a set of word embeddings to the i-th character, where m is the number of words. The j-th word in x_i^{ws} is represented as following: $x_{ij}^w = e^w(w_{ij})$, where e^w is a pre-trained word embedding lookup table and w_{ij} is the j-th word in wsi.Align word representations and word representation dimensions using nonlinear changes:

$$v_{ij}^w = w_2(\tanh(w_1 x_{ij}^w + b_1)) + b_2 \tag{3}$$

where $w_1 \in R^{d_{c*}d_w}$, $w_1 \in R^{d_{c*}d_c}$, and b_1 and b_2 are scaler bias. d_c and d_w denote the dimension of word embedding and the hidden size of BERT respectively.

To pick out the most relevant words from all matched words, we introduce a character-to-word attention mechanism. We denote all v_{wij} assingned to i-th character as $v_i - (v_{i1}^w, v_{i2}^w, \ldots v_{im}^w)$. The relevance of each word can be calculated as:

$$a_i = \text{softmax}(h_i^c w_{attn} v_i^1) \tag{4}$$

where wattn is the weight matrix of billinear attention. Consequently, we can get the weighted sum of all words by:

$$z_i^w = \sigma_{j=1}^m a_{ij} v_{ij}^w \tag{5}$$

Finally, the weighted lexcion information is injected into the character vector by:

$$\tilde{h}_i = h_{ic} + z_i^w \tag{6}$$

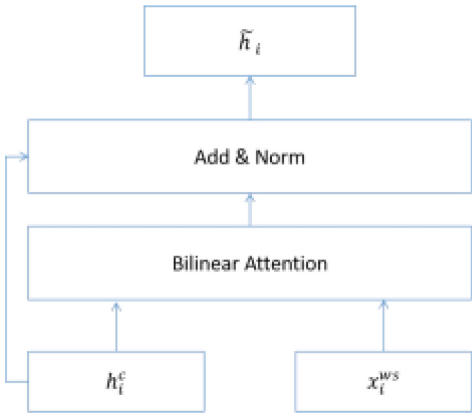

Fig. 4. Structure of Fusion. Enter a character vector and paired word features. By bilinear attention on characters and words, the lexical features are weighted into a vector. This vector is added to the character-level vectors, followed by layer normalization.

Input the fused vector into the remaining Transformer layer for calculation, and finally get $H^l = \{h_1^l, h_2^l, \ldots h_n^l\}$.

3.2 Bi-affine Decoder

The obtained vector representation of each character is input into the bi-affine decoder and mapped to a scoring matrix R of L × L × k, as shown in Fig. 5. L is the sentence length, k ∈ {1,…, |k|}, is the type of entity, |k| is the number of entity types. Specifically, each span (i, j) can be expressed as a tuple (i, j, k). i, j are the start, end index of entity. After BERT encoding, The embedding of the token at the position i, j are h_i, h_j, where h_i, $h_j \in R^d$, d is the hidden size of embedding. We compute the score for a span(i, j):

$$f(i, j) = h_i^T U h_j + w([h_i; h_j]) + b \tag{7}$$

where U is a d × k × d tensor, W is a 2d × k matrix and b is the bias.

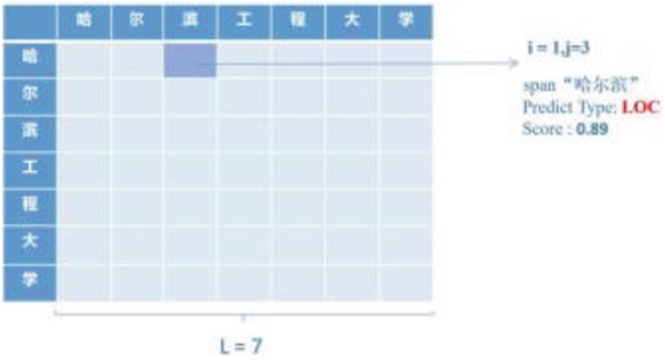

Fig. 5. Scoring matrix R, the element of each position of the square matrix is a k- dimensional vector, which is used to represent the distribution of named entity categories of the text segment corresponding to the position.

3.3 CNN on Score Matrix

Understand the scoring matrix as a picture with k channels and L × L length and width, and further use the Convolutional Neural Network (CNN) commonly used in the field of computer vision to model this spatial connection:

$$R' = \text{Conv2d}(R) \tag{8}$$

$$R'' = \text{Gelu}(\text{LayerNorm}(R' + R)) \tag{9}$$

where Conv2d is 2DCNN, the convolution kernel performs a sliding window operation in two-dimensional space. LayerNorm is layer normalization, which performs normalization operations in the feature layer, and Gelu is the activation function. Since the

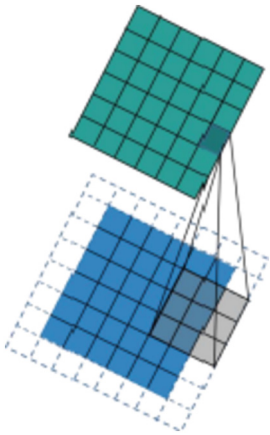

Fig. 6. CNN

number of tokens in the sentence is different, their R have different shapes. In order to ensure the same result when processing *R* in batches, 2DCNN has no bias and fills R with 0, as shown in Fig. 6.

We use a perceptron to get the prediction logits as follows:

$$P = \text{Sigmoid}(w_0(R' + R'') + b) \tag{10}$$

where $w_0 \in R^{|k| \times d}$, $b \in R^{|k|}$, $P \in R^{L \times L \times |k|}$.

3.4 Loss Function

(1) (1) The loss function of the model itself we use the binary cross entropy to calculate the loss as:

$$\mathcal{L}_{BCE} = -\sigma_{0 \le i,j < L} Y_{ij} \log(P_{ij}) \tag{11}$$

where Yij is ground truth entity, Pij is the predicted probability.

The tag for the score matrix is symmetric, namely, the tag in the (i,j)-th entry is the same as in the (j-i)-th. When inference, we calculate scores in the upper triangle part as:

$$\widehat{P}_{ij} = (p_{ij} + p_{ji})/2 \tag{12}$$

where $i \le j$. Then we only use this upper triangle score to get the final prediction.

(2) Contrastive learning based on R-drop

In order to enhance the robustness and generalization ability of the model and reduce the occurrence of overfitting, this paper adopts the idea of contrastive learning based on R-drop. During the training process, because dropout randomly discards some hidden units, the same sentence is input into the model twice to get two different vector representations, but they have the same label. This data augmentation method does not require any

modifications to the neural network structure, but only needs to add a KL bifurcation loss function, so no noise is introduced.

For the construction of positive examples, use the dropout data enhancement method to input a sample sentence into the model twice, and obtain two probability distributions p(i, j), p^{+} (i, j) through the Bert, Bi-affine and CNN modules. In order to construct negative examples, this paper uses Gaussian distribution to initialize M distribution of K × L × L and calculates the loss with the label and then selects the N with the largest loss ass, the negative example $p_{\overline{n}}(i, j)$. The purpose of this is to introduce noise, increase the robustness of the model, and avoid too much negative impact on the training of the model. The loss of contrastive learning is expressed as

$$\mathcal{L}_{\mathrm{KL}} = \begin{matrix} \mathrm{K_L p(i, j)} p^{+}(i, j) \\ \sigma_{\mathrm{n}=0} \mathrm{KL}, \mathrm{p(i, j)}, p_{\overline{n}}(\mathrm{i, j}) \end{matrix} \tag{13}$$

The purpose is to minimize the kl divergence of positive examples and maximize the kl divergence of negative examples to optimize the training effect of the model.

(3) Final Loss Function

The final loss function is expressed as

$$\mathcal{L} = \mathcal{L}_{\mathrm{BCE}} + \mathcal{L}_{\mathrm{KL}} \tag{14}$$

3.5 Entity Decoding

First discard all fragments with a predicted probability lower than 0.5, then sort the spans from high to low according to the predicted probability, and then select the fragment with the highest current predicted probability in turn, if it does not conflict with the previously decoded named entity, then the The fragment is decoded into a new named entity, otherwise it is discarded. By doing this iteratively, all non- conflicting named entities of the input sequence predicted by the model are obtained.

4 Experimental Analysis

4.1 Dataset

We conduct experiments on both the Chinese nested NER dataset and the flat NER dataset. Among them, the Chinese nested NER dataset selects "人民日报" and the Chinese medical dataset CMeEE, and the Chinese flat NER dataset selects Weibo and Resume.

The "人民日报" dataset belongs to the news field and contains three entity types, namely, person names, place names, and organization names. The number of nested entities accounts for about 12.81% of the total number of entities. The CMeEE dataset is called Chinese Medical Entity Extraction dataset. It contains nine types of medical entities such as common pediatric diseases, body parts, clinical manifestations, and medical procedures. Nesting is allowed in the "clinical manifestations" entity category, and other eight types of entities are allowed within this entity. The Weibo dataset is

generated by filtering and filtering the historical data of Sina Weibo from November 2013 to December 2014, including 1890 Weibo messages. The entity category of this dataset is divided into four categories: people, organizations, addresses and geopolitical entities. The Resume data set is generated by screening and manual labeling based on the summary data of resumes of senior managers of listed companies on Sina Finance and Economics. The data set contains 1027 resume summaries, and the entity annotations are divided into 8 categories including name, nationality, place of origin, race, major, degree, institution, and job title. The statistics of the above datasets are shown in Table 1.

Table 1. The statistics of the datasets

Dataset	Train	Dev	Test
《人民日报》	15.3K	1.9K	2. 1K
CMeEE	15K	5K	5K
Weibo	1.4K	0.27K	0.27K
Resume	3.8K	0.46K	0.48K

4.2 Experimental Settings

The BERT-base-chinese pre-training model with 12 hidden layers, outputting 768-dimensional tensors, 12 self-attention heads, and a total of 110M parameters is used in this study. The model is pre-trained on Simplified and Traditional Chinese texts. The 200-dimensional pretrained word embeddings of Song et al. [32] are used, which are trained on news and webpage texts using a directional tab model. Dictionary D is trained on texts such as Wikipedia and Baidu Baike. The Adam optimizer with a learning rate of 2e−5 is used for model optimization during training. The maximum epoch on all datasets is 30, and the maximum input length is 150. The character-word pair information between the 1st and 2nd Transformers in BERT is fused, and BERT and pretrained word embeddings are fine-tuned during training. The CNN convolution kernel is set to 3. An entity is considered correct when both the predicted class and the predicted span are exactly correct. Evaluation metrics used in this study include Precision (P), Recall (R), and F-score (F1). The hyperparameter settings are summarized in Table 2.

4.3 Analysis of Results

Baselines:

LSTM-Crf [33]: The LSTM-CRF model is a traditional sequence labeling model composed of two parts. First, the LSTM (Long Short-Term Memory) neural network maps each input element to a high-dimensional vector space by learning the contextual information in the input sequence. The LSTM network can handle variable-length sequences and update the parameters.

Table 2. The hyper-parameter in this paper

Parameter	Value
Optimiser	Adam
BERT leaning rate	2e−5
Epoch	30
Max seqence length	150
Lexcion add layer	1
CNN kernel size	3

BERT-Crf [34]: The BERT-CRF model is a sequence labeling model based on a pre-trained Transformer model. It first uses the pre-trained BERT model to encode the input sequence to obtain context-aware word embeddings. These embeddings capture rich semantic information of the input sequence and are fed to the CRF layer for label prediction. Compared with traditional models such as LSTM-CRF, BERT-CRF can better capture rich semantic information and dependency relationships between labels.
LEBERT-Crf [8]: LEBERT-CRF model is a sequence labeling model that integrates external lexical knowledge directly into the BERT layer. Specifically, it integrates external lexical knowledge into the BERT layer through the vocabulary adapter module, and uses a linear transformation layer to fuse external knowledge with internal embeddings. Then, the CRF layer is used for label decoding. The advantage of the LEBERT-CRF model is that it can directly integrate external knowledge into the model, thereby improving its performance.

To assess the efficacy of the model proposed in this study, we compared its experimental outcomes against those of the baseline model across four datasets. The comparative findings are presented in Tables 3 and 4.

Table 3. Nested dataset comparison experiment results

Models	《人民日报》			CMeEE		
	P	R	F1	P	R	F1
LSTM-Crf	86.02	81.37	83.63	50.20	44.20	47.00
BERT-Crf	92.26	90.58	91.41	56.90	56.00	56.45
LEBERT-Crf	92.51	93.49	93.00	58.26	56.47	57.35
Ours	96.08	96.13	96. 11	64.98	65.46	65.22

According to the results in Table 3, on the Chinese nested dataset “人民日报”, both the BERT-CRF and LEBERT-CRF models outperformed the LSTM-CRF in terms of precision, recall, and F1-score. The proposed method in this paper achieved higher precision,

recall, and F1-score (96.08%, 96. 13%, and 96. 11%, respectively) than the corresponding values of the other three models. Specifically, compared with the LEBERT-CRF model, our proposed method showed improvements of 3.57%, 2.64%, and 3. 11%, indicating better performance in the task on this dataset. On the nested dataset CMeEE, our proposed method achieved better performance in precision, recall, and F1-score than the other three models. Compared with the F1- score values of the LSTM-CRF, BERT-CRF, and LEBERT-CRF models (47.00%, 56.45%, and 57.35%, respectively), the proposed method achieved an F1-score of 65.22%, representing relative improvements of 18.22%, 8.77%, and 7.87%, respectively. These results demonstrate that our proposed method achieves better performance and usability than the other three models on this dataset.

Table 4. Flat dataset comparison experiment results

Models	Weibo			Resume		
	P	R	F1	P	R	F1
LSTM-Crf	53.04	62.25	58.79	94.81	94.11	94.46
BERT-Crf	57. 14	66.67	59.92	95.37	94.84	95. 11
LEBERT-Crf	70.94	67.02	70.50	95.75	95.10	95.42
Ours	70.84	73.87	72.32	96.96	96.35	96.65

According to the results in Table 4, the proposed method in this paper achieves significant improvement in precision, recall, and F1 score on the flat data set Weibo compared to the LSTM-CRF model and BERT-CRF model. Specifically, compared to the LSTM-CRF model, the proposed method improves precision, recall, and F1 score by 17.80, 11.62, and 13.53% points, respectively. Compared to the BERT- CRF model, the proposed method improves precision, recall, and F1 score by 13.70, 7.20, and 12.40% points, respectively. Compared to the LEBERT-CRF model, the proposed method has similar precision but significantly higher recall and F1 score, improving by 6.85 and 1.82% points, respectively. These results demonstrate that the proposed method outperforms the other three models in terms of performance, indicating its superior classification ability and generalization performance on this data set.

On the flat data set Resume, the proposed method also exhibits excellent performance with higher precision (96.96%), recall (96.35%), and F1 score (96.65%) than the other three models. Specifically, compared to the LSTM-CRF model, the proposed method improves precision, recall, and F1 score by 1.15, 2.24, and 1.19% points, respectively. Compared to the BERT-CRF model, the proposed method improves precision, recall, and F1 score by 1.59, 1.51, and 1.54% points, respectively. Compared to the LEBERT-CRF model, the proposed method improves precision, recall, and F1 score by 1.21, 1.25, and 1.23% points, respectively. Therefore, the proposed method exhibits significantly superior performance on the flat data set Resume compared to the other three models.

5 Ablation Study

To verify the effectiveness of our proposed method in Chinese nested named entity recognition, we chose to conduct ablation experiments on the Chinese nested datasets "人民日报" and CMeEE. We further chose to delete some components and conducted three experiments: 1) Our complete model, using dictionary assistance, incorporates character-word pair information into BERT, and uses bi-affine structure encoding to obtain a 3D feature matrix. At the same time, the feature matrix is regarded as an image, and the local interaction between spans is modeled by using the convolutional neural network (CNN), and the spatial correlation between adjacent spans is fully utilized, and finally all non-conflicting features of the predicted input sequence are obtained. Named entity. 2) Remove the CNN module, skip formulas (6)–(7), and directly obtain the predicted entity after bi-affine structure decoding. 3) Remove the dictionary auxiliary module, skip formulas (3)–(5), and refer to the use of character information. 4) Remove the contrastive learning module, skip formulas (13) The comparison between our full model(14), and only use binary cross-entropy to calculate the loss function.The experimental results are shown in Table 5.

Table 5. The comparison between our full model and ablated models

Models	《人民日报》			CMeEE		
	P	R	F1	P	R	F1
Ours	96.08	96.13	96.11	64.98	65.46	65.22
Ours(w/o CNN)	94.71	94.08	95.09	63.25	71.43	62.50
Ours(w/o KB)	92.35	91.87	92.56	63.25	54.20	58.38
Ours(w/o contrast)	95.75	95.10	95.42	65.58	62.73	64. 12

On the nested data set "人民日报", the performance of the model decreased slightly after removing the CNN module, with precision, recall, and F1 score decreasing by 1.37%, 2.05%, and 1.02%, respectively. Removing the dictionary- assisted module had a greater impact on the model's performance, with precision, recall, and F1 score decreasing by 3.73%, 4.26%, and 3.55%, respectively. The removal of the contrastive learning module had a relatively small impact on the model's performance, with precision, recall, and F1 score decreasing by 0.33%, 0.03%, and 0.69%, respectively. On the nested data set CMeEE, the F1 score of this approach (65.22%) was higher than that of the other three experiments, with the experiment that removed the dictionary-assisted module achieving the lowest F1 score. When the CNN module was removed, the F1 score decreased by 2.72, indicating that the CNN module has certain advantages in modeling local interactions between spans on this dataset. Additionally, the dictionary-assisted module in this approach had a significant effect on Chinese nested named entity recognition, with a significant decrease in F1 score after its removal. The effect of the contrastive learning module was relatively stable, and its removal also led to a slight decrease in F1 score,

which indicates that the module can enhance the model's robustness and generalization ability while reducing overfitting. Overall, the various components in this approach contributed to the model's performance to varying degrees.

6 Conclusion

In this paper, we propose KBCNER, a dictionary-assisted Chinese Nested Named entity Recognition model. The matching words are obtained through the dictionary, and the character-phrase pairs are formed and integrated into BERT. The semantic information contained in Chinese phrases is richer than that of a single character, and the dictionary information enhancement feature is introduced to obtain richer semantics. Using the bi-affine structure, get a global view of the span. At the same time, the feature matrix is regarded as an image, and the local interaction between spans is modeled by using a Convolutional Neural Network (CNN), which improves the recognition accuracy of nested entities. Finally, the idea of contrastive learning based on R-drop is adopted to enhance the robustness of the model. In the experimental part, the model is compared with the Chinese nested NER dataset ("人民日报", CMeEE) and the flat NER dataset (Weibo, Resume). At the same time, we conduct ablation experiments to analyze in detail the influence of the main components of the model on its performance. The model has achieved better performance than the baseline model on all data sets, indicating that the model has strong adaptability and versatility in different fields and different data sets.

Acknowledgment. This work was supported by the National Key R&D Program of China under Grant No. 2020YFB1710200.

References

1. Yao, H., Wu, Y., Al-Shedivat, M., et al.: Knowledge-aware meta-learning for low- resource text classification. arXiv preprint arXiv:2109.04707 (2021)
2. Chen, M.Y., Jiang, H., Yang, Y.: Context enhanced short text matching using clickthrough data. arXiv preprint arXiv:2203.01849 (2022)
3. Yu, W., Zhu, C., Li, Z., et al.: A survey of knowledge-enhanced text generation. ACM Comput. Surv. **54**(11s), 1–38 (2022)
4. Wang, X., Shen, Y., Cai, J., et al.: Damo-NLP at semeval-2022 task 11: A knowledge-based system for multilingual named entity recognition. arXiv preprint arXiv:2203.00545 (2022)
5. Zhang, Y., Yang, J.: Chinese NER using lattice LSTM. arXiv preprint arXiv:1805.02023 (2018)
6. Li, X., Yan, H., Qiu, X., et al.: FLAT: Chinese NER using flat-lattice transformer. arXiv preprint arXiv:2004.11795 (2020)
7. Liu, W., Fu, X., Zhang, Y., et al.: Lexicon enhanced Chinese sequence labeling using BERT adapter. arXiv preprint arXiv:2105.07148 (2021)
8. Sohrab, M.G., Miwa, M.: Deep exhaustive model for nested named entity recognition. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp. 2843–2849. Association for Computational Linguistics (2018)

9. Timothy Dozat and Christopher Manning. 2017. Deep biaffifine attention for neural dependency parsing. In: Proceedings of 5th International Conference on Learning Representations (ICLR)
10. Yu, J., Bohnet, B., Poesio, M.: Named entity recognition as dependency parsing. In: Proceedings of the 58th Annual Meeting of the as Sociation for Computational Linguistics, ACL 2020, Online, July 5–10, 2020, pp. 6470–6476. Association for Computational Linguistics (2020)
11. Yan, H., Sun, Y., Li, X., et al.: An embarrassingly easy but strong baseline for nested named entity recognition. arXiv preprint arXiv:2208.04534 (2022)
12. Wu, L., Li, J., Wang, Y., et al.: R-drop: regularized dropout for neural networks. Adv. Neural. Inf. Process. Syst. **34**, 10890–10905 (2021)
13. Ju, M., Miwa, M., Ananiadou, S.: A neural layered model for nested named entity recognition. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long Papers), pp. 1446–1459 (2018)
14. Straková, J., Straka, M., Hajič, J.: Neural architectures for nested NER through linearization. arXiv preprint arXiv:1908.06926 (2019)
15. Shibuya, T., Hovy, E.: Nested named entity recognition via second-best sequence learning and decoding. Trans. Assoc. Comput. Linguist. **8**, 605–620 (2020)
16. Li, H., Xu, H., Qian, L., et al.: Multi-layer joint learning of Chinese nested named entity recognition based on self-attention mechanism. In: CCF International Conference on Natural Language Processing and Chinese Computing. Springer, Cham (2020) 144–155
17. Wang, J., Shou, L., Chen, K., et al.: Pyramid: a layered model for nested named entity recognition. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5918–5928 (2020)
18. Lu, W., Roth, D.: Joint mention extraction and classification with mention hypergraphs. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 857–867 (2015)
19. Katiyar, A., Cardie, C.: Nested named entity recognition revisited. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1 (2018)
20. Wang, B., Lu, W.: Neural segmental hypergraphs for overlapping mention recognition. arXiv preprint arXiv:1810.01817 (2018)
21. Luo, Y., Zhao, H.: Bipartite flat-graph network for nested named entity recognition. arXiv preprint arXiv:2005.00436 (2020)
22. Finkel, J.R., Manning, C.D.: Nested named entity recognition. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pp. 141–150 (2009)
23. Fu, Y., Tan, C., Chen, M., et al.: Nested named entity recognition with partially-observed treecrfs. In: Proceedings of the AAAI Conference on Artificial Intelligence **35**(14), 12839–12847 (2021)
24. Lou, C., Yang, S., Tu, K.: Nested named entity recognition as latent lexicalized constituency parsing[J]. arXiv preprint arXiv:2203.04665 (2022)
25. Yang, S., Tu, K.: Bottom-up constituency parsing and nested named entity recognition with pointer networks. arXiv preprint arXiv:2110.05419 (2021)
26. Lin, H., Lu, Y., Han, X., et al.: Sequence-to-nuggets: nested entity mention detection via anchor-region networks. arXiv preprint arXiv:1906.03783 (2019)
27. Xia, C., Zhang, C., Yang, T., et al.: Multi-grained named entity recognition. arXiv preprint arXiv:1906.08449 (2019)

28. Zheng, C., Cai, Y., Xu, J., et al.: A boundary-aware neural model for nested named entity recognition. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics (2019)
29. Yu, J., Bohnet, B., Poesio, M.: Named entity recognition as dependency parsing. arXiv preprint arXiv:2005.07150 (2020)
30. Xu, Y., Huang, H., Feng, C., et al.: A supervised multi-head self-attention network for nested named entity recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 16, pp. 14185–14193 (2021)
31. Shen, Y., Ma, X., Tan, Z., et al.: Locate and label: a two-stage identifier for nested named entity recognition. arXiv preprint arXiv:2105.06804 (2021)
32. Song, Y., Shi, S., Li, J., Zhang, H.: Directional skip-gram: Explicitly distinguishing left and right context for word embeddings. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 2 (Short Papers), pp. 175–180, New Orleans, Louisiana. Association for Computational Linguistics (2018)
33. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
34. Li, X., Zhang, H., Zhou, X.H.: Chinese clinical named entity recognition with variant neural structures based on Bert methods. J. Biomed. Inform. **107**, 103422 (2020)