International Journal of Computational Intelligence Systems (2024) 17:144
https://doi.org/10.1007/s44196-024-00521-9

RESEARCH ARTICLE

LB-BMBC: MHBiaffine-CNN to Capture Span Scores with BERT Injected with Lexical Information for Chinese NER

Tao Guo[1] · Zhichao Zhang[1]

Received: 12 December 2023 / Accepted: 13 May 2024

Abstract
A substantial body of research has shown that introducing lexical information in Chinese Named Entity Recognition (NER) tasks can enhance the semantic and boundary information of Chinese words. However, in most methods, the introduction of lexical information occurs at the model architecture level, which cannot fully leverage the lexicon learning capability of pre-trained models. Therefore, we propose seamless integration of external Lexicon knowledge into the Transformer layer of BERT. Additionally, we have observed that in span-based recognition, adjacent spans have special spatial relationships. To capture this relationship, we extend the work after Biaffine and use Convolutional Neural Networks (CNN) to treat the score matrix as an image, allowing us to interact with the spatial relationships of spans. Our proposed LB-BMBC model was experimented on four publicly available Chinese NER datasets: Resume, Weibo, OntoNotes v4, and MSRA. In particular, during ablation experiments, we found that CNN can significantly improve performance.

Keywords Lexicon information · Chinese NER · MHBiaffine · CNN

1 Introduction

Information extraction [1] aims to meet the demand of individuals for rapidly and accurately accessing information and knowledge on the Internet. Specifically, information extraction refers to the process of identifying information from extensive unstructured data that cater to user needs. Among these processes, Named Entity Recognition (NER) [2, 3], operating as a text mining algorithm, assumes a crucial role in converting unstructured text into structured content. This algorithm offers robust support for the execution of information extraction tasks and enables the implementation of subsequent downstream applications. Compared to English NER, Chinese NER [4] faces greater challenges. This is primarily due to the complexities of Chinese language involving word segmentation, difficulty in determining boundaries of related entities, and intricate grammatical structures. In Chinese NER, entity boundaries align with character boundaries, and the segmentation process inevitably propagates errors, presenting a significant challenge in accurate recognition of entities in Chinese NER.

Introducing lexical information can effectively alleviate this issue, to fully leverage lexical information, a novel variant of the LSTM model called Lattice-LSTM [5] was introduced. This model utilizes words within a sentence that is segmented into individual characters and encodes these words into a directed acyclic graph. To address the non-parallelizability issue of Lattice-LSTM, WC-LSTM [6] introduced four word embedding strategies: shortest, longest, average, and self-attention. The FLAT [7] model incorporates a lattice structure and employs fully connected self-attention to capture long-distance dependency relationships within sequences. LEBERT [8] injects lexical information into the underlying BERT model by introducing a Lexicon Adapter. This innovative approach facilitates the integration of lexical knowledge into the BERT architecture. However, due to the presence of polysemy in the Chinese language, it is necessary to strike a balance in the value of injected lexical information.

In recent years, Span-based models have gained significant attention in NER research. This approach typically involves enumerating all candidate spans and categorizing them into entity types (including a "non-entity" type). Bi-LSTM is used to capture the contextual information of

✉ Zhichao Zhang
zhichao@ustb.edu.cn

Tao Guo
m202210695@xs.ustb.edu.cn

[1] School of Mathematics and Physics, University of Science and Technology Beijing, Beijing 100083, China

Published online: 10 June 2024

sentences, and then input into Biaffine attention [9] to score each segment. This method contributes to predicting entities within the text. In certain studies, the NER task has been transformed into a machine reading comprehension (MRC) [10, 11] task. They employ entity types as queries, asking the model whether a given segment belongs to a specific entity type. The W2NER model [12] converts the NER task into predicting the relationship categories between pairs of words. The Span-based methods are friendly to parallelism and the decoding is easy. Therefore, this formulation has been widely adopted [13, 14]. However, previous work has overlooked the spatial relationships between adjacent spans.

By dynamically extracting words corresponding to characters and feeding them into the Transformer layers of BERT, we ensured consistent dimensions for characters and words using a bilinear attention mechanism. We fine-tuned certain parameters of the BERT component to fully leverage word information. Furthermore, the spans surrounding a central span exhibit unique relationships, as shown in the Fig. 1, which can enhance the model's understanding of contextual information within the text. For further information, please refer to CNN-NER [14]. To leverage these correlations, we employed a biaffine decoder to generate a 3D feature matrix. Treating this feature matrix as an image, we utilized CNN to model the local interactions adjacent spans.

In summary, our main contributions are as follows:

- Proposing Char–Words pairs, injecting lexical information into the Transformer layers of BERT, and adaptively adjusting word representations.
- We observed interconnections between adjacent spans. After the BiLSTM layer, we employed a multi-head Biaffine to obtain a span feature matrix. Treating this matrix as an image, we utilized CNN to model the interactions between adjacent spans.
- We proposed the LB-BMBC (**L**exicon **B**ERT + **B**iLSTM + **MHB**iaffine + **C**NN) model and conducted experiments on four Chinese datasets: Resume [5], Weibo [15], Ontonotes [16], and MSRA [17]. Additionally, we performed ablation experiments to validate the effectiveness of the method.

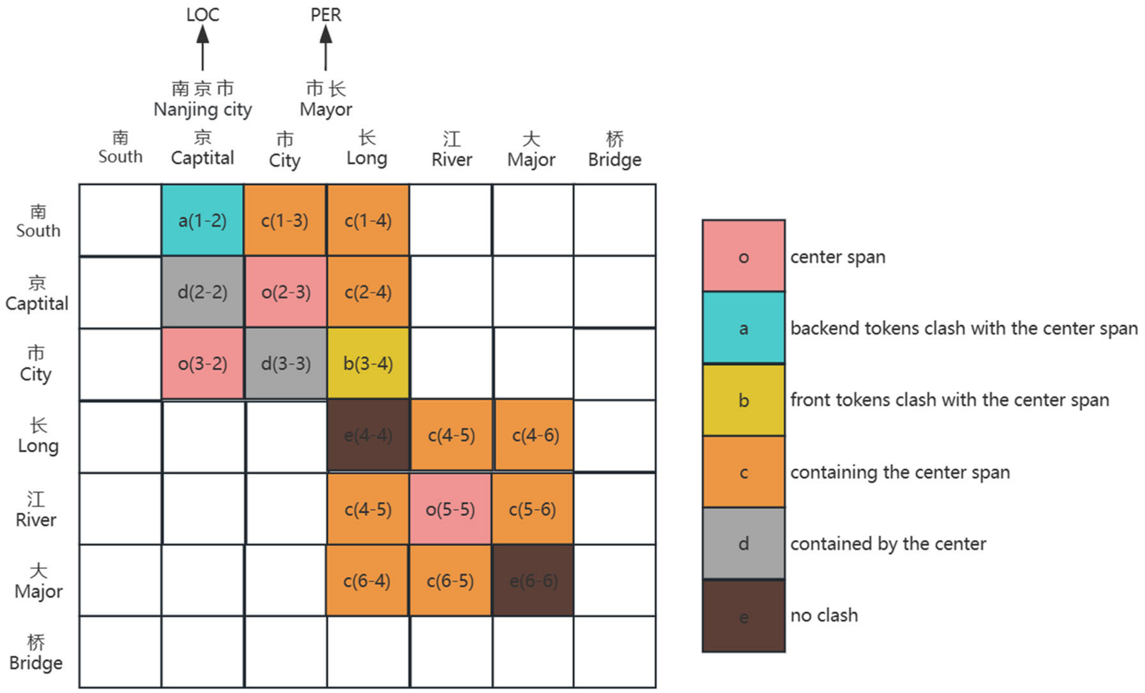
2 Related Work

2.1 Methods Based on Statistical Machine Learning

In previous work, supervised machine learning classification models were employed for NER, including models, such as HMM, MEM, SVM, and CRF. Zhang et al. [18] proposed an automatic Chinese person name recognition method based on role tagging using HMM. This method identifies and categorizes named entities by maximizing the matching of the best role sequence, addressing challenges such as the loss of names without distinctive features, internal word formation, and the difficulty in recalling person names within context-dependent word formations. Biekl et al. [19] used HMM to calculate the probability that a word is of an entity type based on features, such as case, number symbol, and the first word of a sentence. Zhou et al. [20] were among the first to apply MEM to the recognition of Chinese noun phrases, transforming the phrase recognition problem into a labeling problem. They extracted candidate features from pre-defined feature templates in the corpus and identified noun phrases based on these candidate features. Zhang et al. [21] proposed an MEM model that combines multiple features, integrating both local and global features. This model integrated rule-based and machine learning methods while incorporating heuristic knowledge to address efficiency and space issues. Takeuchi et al. [22] used SVM for NER in the MUC-6 evaluation corpus and in the field of molecular biology. They found that SVM performed well in the domain of biological NER. Li Lishuang et al. [23] proposed an automatic recognition method for Chinese place names based on SVM. They incorporated characteristic information of place names as vector features. Additionally, they employed active learning strategies to gradually increase the scale of classifier training samples, further improving the classifier's recognition. CRF models, which compute global probabilities and normalize not only locally but also globally, were widely applied in NER. McCallum et al. [24] proposed a CRF-based feature induction method, which automatically induced features to enhance accuracy while significantly reducing feature numbers. Feng et al. [25] proposed small-scale common suffix features into the CRF framework, improving model training speed while maintaining recognition accuracy. Yan Yang et al. [26] proposed a stacked CRF approach and addressed NER in Chinese electronic medical records. In the second layer, they used a feature set containing entities and lexicon information to recognize two types of named entities: disease names and clinical symptoms.

2.2 Methods Based on Deep Learning

Traditional supervised learning methods based on feature engineering have consumed a significant amount of human effort. Furthermore, during the feature extraction process, errors often propagate due to the inapplicability of prior experience. In contrast, deep learning methods, thanks to their end-to-end feature extraction capabilities, have effectively addressed this issue. Huang et al. [27] used BiLSTM to extract character-level features of words. Gregoirc et al. [28] adopted multiple independent BiLSTM units in the input and promoted diversity among LSTM units using inter-model regularization, reducing model parameters. Yang et al. [29] introduced a self-attention mechanism before BiL-

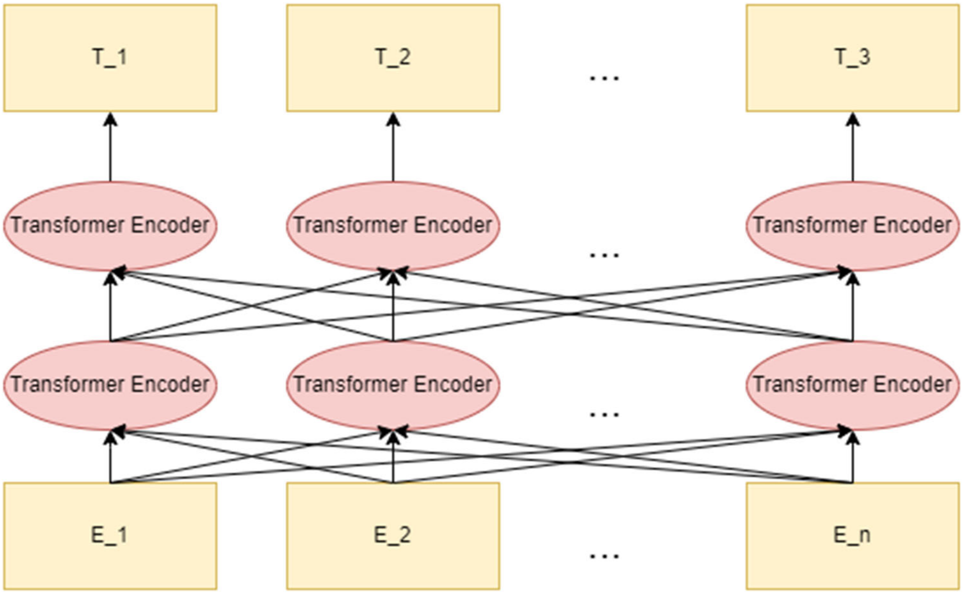


	南 South	京 Captital	市 City	长 Long	江 River	大 Major	桥 Bridge
南 South		a(1-2)	c(1-3)	c(1-4)			
京 Captital		d(2-2)	o(2-3)	c(2-4)			
市 City		o(3-2)	d(3-3)	b(3-4)			
长 Long				e(4-4)	c(4-5)	c(4-6)	
江 River				c(4-5)	o(5-5)	c(5-6)	
大 Major				c(6-4)	c(6-5)	e(6-6)	
桥 Bridge							

o	center span
a	backend tokens clash with the center span
b	front tokens clash with the center span
c	containing the center span
d	contained by the center
e	no clash

Fig. 1 Spatial relationships between adjacent spans. For instance, o(2-3) is represented as “京 (Capital) 市(City)”, and it is surrounded by “南京市 (Nanjing city)”, which is categorized as a location entity (LOC). It conflicts with “市长 (Mayor)” on the term “市 (City)”, as the mayor is categorized as a personal entity (PER). the center span can have the special relationship with its surrounding spans (different relations are depicted in different colors)

Fig. 2 BERT structure



STM, allowing the model to adjust its focus on different parts of the input sequence, capturing long-distance dependencies. Xu et al. [30] incorporated multi-head self-attention and dictionary information to adjust the weight relationships between Chinese characters and multi-level semantic features. While most BiLSTM architectures capture global features of sentences, they often lack local features. CNN were initially popular in computer vision for their ability to capture local features and have gradually been applied in NLP. Wu et al. [31] used CNN to represent the entire sentence as a global feature while extracting local features. After extracting both global and local features, they connected them to a fully connected neural network for sequence labeling and entity recognition. Kong et al. [32] proposed a Chinese clinical NER method that combines multi-level CNN and attention mechanisms, addressing the limitation of LSTM in capturing global information for long sentences. Strubell et al. [33] proposed an Iterated Dilated Convolutional Neural Network (ID-CNN) that offers better context and structured prediction capabilities compared to traditional CNNs while significantly reducing training time by fully utilizing CPU parallelism. Jiang et al. [34] proposed the Word Embedding-based BiLSTM-IDCNN-CRF model, utilizing different network architectures for obtaining global and local features. The advent of pre-trained models has ushered NLP into a new era. Research has shown that pre-trained models trained on very large corpora can learn language text representations suitable for various domains, benefiting various downstream NLP tasks, including NER. BERT is a transformer-based bidirectional encoder that leverages self-supervised learning tasks to mine contextual representations. Researchers have proposed improved pre-trained models based on BERT, such as RoBERTa [35], ALBERT [36], and BioBERT [37]. Chang et al. [38] concatenated CRF with pre-trained BERT models for NER tasks. Liu et al. [39] proposed the BERT-BiLSTM-CRF recognition method and applied it to research on citrus pests and diseases. Gan et al. [40] proposed the BERT-Transformer-BiLSTM-CRF model to handle the challenges posed by pronouns and polysemous words in Chinese NER. Li et al. [41] addressed the issue of excessive parameters and long training times in BERT by introducing the BERT-IDCNN-CRF model.

Sequence labeling-based [42] methods can encounter challenges when dealing with nested entities, as the Cartesian product of entity labels might lead to addressing the long-tail issue. Using hypergraphs [43] can effectively identify spans, but the decoding process becomes challenging. The Seq2Seq [44] framework can be employed to generate entity sequences, which can be either entity pointer sequences [45] or text sequences [46]. However, Seq2Seq suffers from the issue of high decoding time during the process.

2.3 Lexical Information for NER

Compared to English NER, Chinese NER faces more challenges primarily due to the difficulty in determining entity boundaries in Chinese text and the complex syntactical structure of the Chinese language. Previous research [4] compared character-based and word-based approaches, and character-based NER methods often fail to fully harness explicit word and word sequence information, despite its potential value. To leverage lexical features, Zhang et al. [5] proposed the Lattice-LSTM model, which encodes all words matched by individual characters in a sentence into a DAG. However, the DAG structure may sometimes struggle to select the correct path, potentially causing the model to degrade into a Character-based model. Liu et al. [6] proposed the WC-LSTM model to integrate word information into character-based models, employing four different word encoding strategies: shortest, longest, average, and self-attention. These strategies encode word information into fixed-size vectors, enabling batch training and adaptability to various application scenarios. To maximize the benefits of pre-trained models, Lai et al. [7] adopted a Lattice structure and employed fully connected self-attention to capture long-range dependencies in sequences. Liu et al. [8] introduced a Lexicon Adapter into the Transformer Encoder layer of BERT, allowing individual characters within sentences to interact with lexical information. This effectively enhances the model's ability to acquire the meaning of individual characters based on contextual semantics.

2.4 Biaffine for NER

Yu et al. [47] proposed a Biaffiner decoder from dependency parsing, which is used to convert the span classification into classifying the start and end token pairs. Text spans are treated as candidate entities and span tuples as candidate relation tuples allows for the sharing of span semantics [48]. Hanoi et al. [49] proposed in BiLSTM-Biaffine, using the context-rich word vector representation provided by BiLSTM to further accurately predict the entity category to which each word belongs and the start and end positions of the entity. Li et al. [50] introduced attention mechanism into Biaffine for the first time, achieving faster training speed under the same performance as BiLSTM. Gu et al. [51] combine Biaffine with Regularity-aware Module to effectively explore the internal composition information of entities, and use the special naming patterns or naming rules of entities to further enhance entity boundaries. Yan et al. [14] consider connecting CNN after Biaffine and using CNN to model adjacent spatial span relationships.

3 Method

3.1 BERT Pre-training Module

The internal architecture of BERT is primarily composed of multiple Transformer Encoder layers. BERT takes embedded vectors as input, denoted as $E = \{E_1, E_2, \ldots, E_n\}$. These vectors are then processed through multiple Transformer Encoders to yield the output layer, represented as $T = \{T_1, T_2, ..., T_n\}$, as shown in Fig. 2. In our approach, we introduce the Lexicon Adapter, which modifies one of these Transformer Encoders. Each encoder includes components, such as multi-head attention layers, feed-forward neural networks, and layer normalization, as depicted in Fig. 3. BERT is described by the following parameters: L, H, and T. Here, L corresponds to the number of layers in the Transformer,

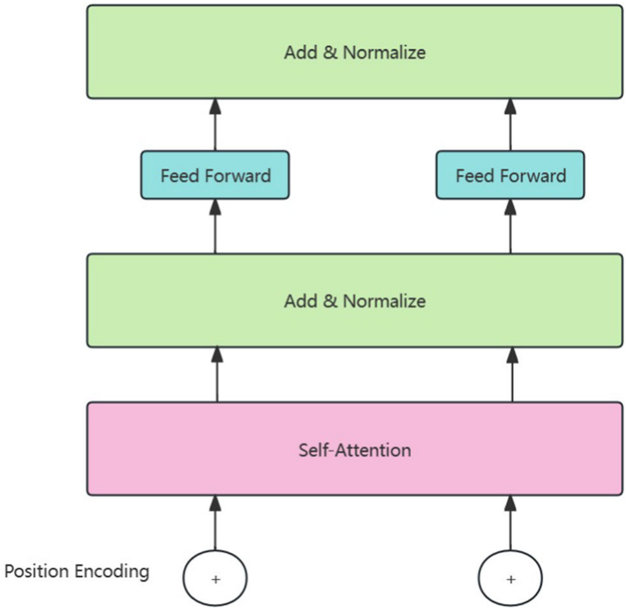


Fig. 3 Transformer encoder unit

denotes the output dimensionality, and T represents the total count of model parameters. In this study, we utilize bert-base-chinese, which comprises 12 Transformer Encoder layers.

To obtain input vectors E for a Chinese sequence $S = \{s_1, s_2, \ldots, s_n\}$

$$E = \text{Token}_{\text{Embeddings}}(S) + \text{Segment}_{\text{Embeddings}}(S) + \text{Position}_{\text{Embeddings}}(S). \quad (1)$$

Presently, we are directing the vector E into the Transformer Encoder layers, with H^0 initialized as E

$$G = \text{LN}\left(H^{l-1} + \text{MHAttn}\left(H^{l-1}\right)\right) \quad (2)$$

$$H^l = \text{LN}\left(G + \text{FFN}\left(G\right)\right). \quad (3)$$

MHAtten represents multi-head attention mechanism, LN stands for layer normalization and FFN refers to the feed-forward network. Specifically, FFN is a two-layer feed-forward network with ReLU as the hidden activation function. The ultimate layer of the Transformer Encoder functions as the output and is denoted as T.

In the attention mechanism, each word corresponds to three different vectors, namely Query vector (Q), Key vector (K), and Value vector (V). These three vectors are obtained by multiplying the embedding vector by three different weight matrices w_q, w_k, w_v. Then, each word is scored by multiplying the Query vector and Key vector. Attention value is to use softmax to smooth the score item just obtained and then multiply the result with the Value vector

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \quad (4)$$

Furthermore, the Transformer encoder unit incorporates a residual network and layer normalization to address the issue of degradation and enhance model performance

$$\text{LN}(x_i) = \alpha \frac{x_i - u_L}{\sqrt{\sigma_L^2 + \epsilon}} + \beta. \quad (5)$$

$$\text{FFN} = \max(0, xW_1 + b_1)W_2 + b_2, \quad (6)$$

where α and β represent the parameters that need to be learned, and u and σ denote the mean and variance of the input, respectively.

3.2 Lexicon Adapter

The main architecture of the Lexicon Adapter is illustrated in Fig. 4, where Chinese sentences are converted into a sequence of Char–Words pairs. sequence. The Lexicon Adapter is placed between the Transformer layers of BERT, effectively integrating lexical knowledge into BERT. In this section, we describe the following aspects: (1) how the Char–Words pair sequence is generated, and (2) how the Adjust Lexicon Adapter functions within BERT.

3.2.1 Char–Words Pair Sequence

Chinese sentences are typically represented as sequences of characters, containing only character-level features. To fully leverage lexical information, we extend the character sequence into a sequence of Char–Words pairs.

Given a Chinese dictionary D[1] with associated embedding vectors, we traverse D to construct a Trie tree. For a Chinese sequence $S = \{s_1, s_2, \ldots, s_m\}$, by traversing all character subsequences in the sentence and matching them with the Trie tree, we can obtain all potential words. Taking “南(South) 京(Capital) 市(City) 长(Long) 江(River) 大(Major) 桥(Bridge)” as an example, we can obtain all different words: “南京(Nanjing), 南京市(Nanjing city), 市长(Mayor), 长江(Yangtze), 大桥(Major bridge)”. Subsequently, we allocate these words to individual characters in the Chinese sequence, as illustrated in Fig. 5. For instance, “南京(Nanjing)” is assigned to the characters “南(South)” and “京(Capital)” Characters without words are padded with PAD. This results in a Char–Words pair sequence. Exactlt,$S_w = \{(s_1, w_1), (s_2, w_2), \ldots, (s_m, w_m)\}$, where w_i corresponds to the words assigned to s_i.

[1] The dictionary D consists of 2 million pre-trained word vectors with a dimension of 200. https://ai.tencent.com/ailab/nlp/en/download.html

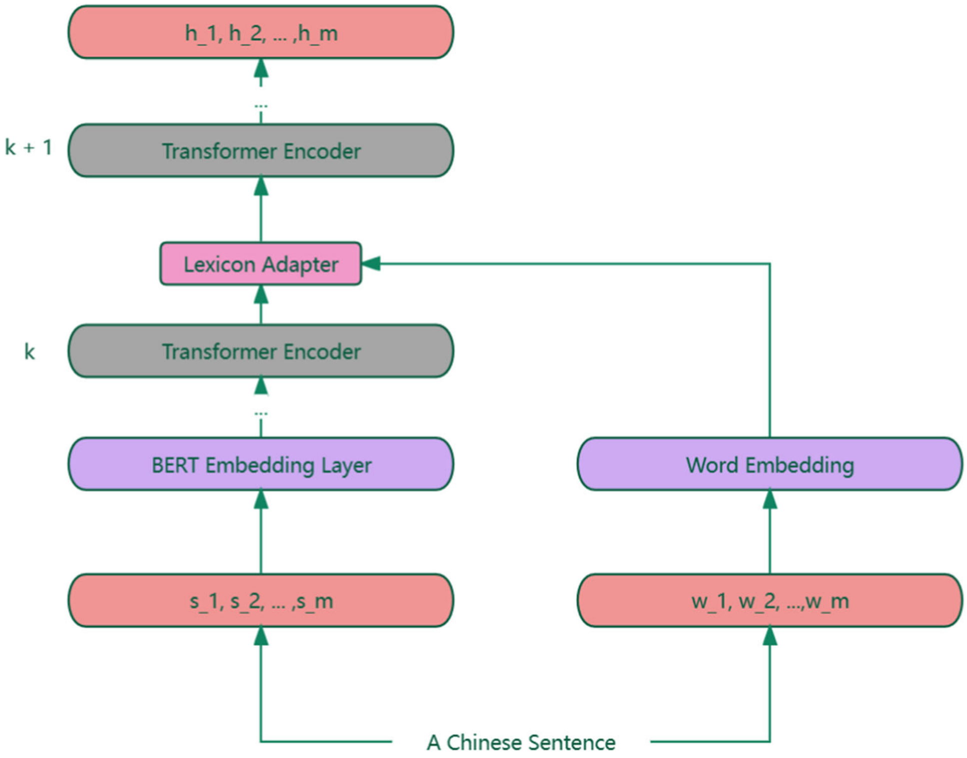


Fig. 4 The architecture of Lexicon Enhanced BERT, in which lexicon features are integrated between kth and $(k+1)$th Transformer Layer using Lexicon Adapter. Where s_i denote the ith Chinese character in the sentence, and w_i denotes matched words assigned to character s_i

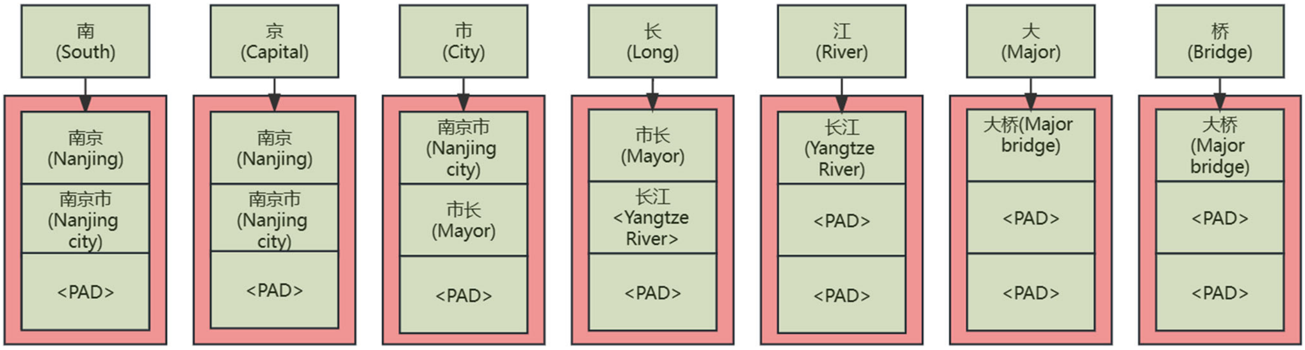


Fig. 5 Character–words pair sequence of a truncated Chinese "南(South) 京(Capital) 市(City) 长(Long) 江(River) 大(Major) 桥(Bridge)", The words that match with "市 (City)" are "南京市 (Nanjing city)" and "市长 (mayor)". PAD denotes padding value and each word is assigned to the characters it contains.

3.2.2 Adjust Lexicon Influence BERT Adapter

To inject lexical information into BERT, we utilize Char-Words pairs as our input features. Now, let us review the workings of the Transformer Encoder

$$H^0 = E \tag{7}$$

$$G = \mathrm{LN}\left(H^{l-1} + \mathrm{MHAttn}\left(H^{l-1}\right)\right) \tag{8}$$

$$H^l = \mathrm{LN}\left(G + \mathrm{FFN}\left(G\right)\right). \tag{9}$$

Here, E corresponds to the outputs obtained after transforming $s_1, s_2, \ldots, s_m$ from $S_w = \{(s_1, w_1), (s_2, w_2), \ldots, (s_m, w_m)\}$ into tokens and incorporating segment and position embeddings. Subsequently, E is fed into Transformer encoders and each Transformer layers acts as follows. MHAtten is multi-head attention mechanism, LN is layer normalization, FFN is the multi-head attention mechanism, and FFN is a two-layer feed-forward network with ReLU as hidden activation function.

In the typical BERT layer, we have $H^l = \{h_1^l, h_2^l, \ldots, h_m^l\}$, without the inclusion of lexical information. However, in the Adjust Lexicon Layer, we possess $\widetilde{H^l} = \widetilde{h_1^l}, \widetilde{h_2^l}, \ldots, \widetilde{h_m^l}$, where the computation formula for $\widetilde{h_i^l}$ is defined as follows:

$$\widetilde{h_i^l} = h_1^l + \gamma z_i. \tag{10}$$

Here, γ is the weight coefficient that governs the influence of lexical information, while z_i signifies the lexical information associated with The ith character.

For the Adjust Lexicon Layer, considering the char–word pairs $S_w = \{(s_1, w_1), (s_2, w_2), \ldots, (s_m, w_m)\}$, we obtain the embedding for w_m and each character can match a maximum of p words

$$x_{ij} = e^w (w_{ij}) \\ i = 1 \ldots, m \quad j = 1 \ldots, p. \tag{11}$$

Here, $x_{ij} \in R^{d_w}$ represents the embedding value of the jth matching word for the ith character. e^w is the pre-trained vocabulary embedding matrix, with an embedding dimension $d_w = 200$. After obtaining the vocabulary embedding values, we apply a nonlinear transformation

$$v_{ij} = W_2 \left(\tanh \left(W_1 x_{ij} + b_1\right)\right) + b_2. \tag{12}$$

Here, $W_1 \in R^{d_c \times d_w}$, $W_2 \in R^{d_c \times d_c}$, b_1 and b_2 are scalar biases. $d_c = 768$ represents the hidden size of BERT.

Specifically, we denote all v_{ij} assigned to ii-th character as $V_i = \{v_{i1}, v_{i2}, \ldots, v_{ip}\}$, The relevance of each word can be calculated as

$$\alpha_i = \mathrm{softmax}(h_i^l W_{attn} V_i^T), \tag{13}$$

where $W_{\mathrm{attn}} \in R^{d_c \times d_c}$ is the weight matrix of bilinear attention. Consequently, we can get the weighted sum of all words by

$$z_i = \sum_{j=1}^{p} a_{ij} v_{ij}. \tag{14}$$

3.3 CNN-Span

The main architecture of CNN-Span is illustrated in Fig. 6, where we introduce lexical information into BERT, pass it through Bi-LSTM to capture contextual information, and then feed it to a multi-head Biaffine layer to obtain probability scores for entities. Finally, we utilize a residual-connected CNN to model the spatial correlation between adjacent entities.

3.3.1 BiLSTM and MHBiaffine

We approach this NER task as a span classification task, where our model designates an entity label for each valid span. Initially, we input the Chinese sentence $S_w = \{(s_1, w_1), (s_2, w_2), \ldots, (s_m, w_m)\}$ into the Encoder enriched with lexical information

$$H = \mathrm{Encoder}_{\mathrm{Lexicon}} (S_w). \tag{15}$$

Here, $H \in R^{m \times d_c}$. In the next step, we convey the output H to a BiLSTM to obtain the Head and Tail of each span

$$H_H = \mathrm{BiLSTM}(H) \tag{16}$$

$$H_T = \mathrm{BiLSTM}(H). \tag{17}$$

$H_h \in R^{m \times d_h}$, $H_T \in R^{m \times d_h}$, d_h represents the size of the output layer in the BiLSTM, which can obtain the context information of each element by traversing in both forward and reverse directions, which is crucial for understanding the context environment of the entity.

Then, we convey both H_H and H_T through a multi-head Biaffine decoder to get the score matrix Q

$$Q_{ij} = H_h(i)^T U H_T(j) + W(H_H(i) \oplus H_T(j)) + b; \tag{18}$$

$Q \in R^{m \times m \times |T|}$, where each cell(i,j) within Q can be seen as the feature vector in Q corresponding to the span. Specifically, and for the lower triangle of Q (where $i > j$), the span contains character from jth to the ith positions.

3.3.2 CNN

Now, we apply a CNN to model the interaction adjacent spans. We repeat the following steps in the model:

$$Q' = \mathrm{Conv2d}(Q) \tag{19}$$

$$Q'' = \mathrm{GeLU}\left(\mathrm{LayerNorm}\left(Q' + Q\right)\right), \tag{20}$$

where Conv2d, LayerNorm, and GeLU represent the 2D CNN, layer normalization, and GeLU activation function. Layer normalization is performed across the feature dimension, allowing the network to use a higher Learning Rate without causing delivery problems during the training process, accelerating model convergence. It is important to note that due to the varying number of tokens m in sentences, their Q matrices have different shapes. To ensure consistent results in batch processing, the 2D CNN has no bias term, and all the paddings within Q are filled with 0. Following traversal through multiple CNN blocks, the Q'' will be further processed by another 2D CNN module.

We use a Sigmoid Function to get the prediction score as follows:

$$P = \mathrm{Sigmoid}\left(W_o \left(Q + Q''\right) + b\right). \tag{21}$$

Here, $P \in R^{m \times m \times |T|}$. And then, we use the binary cross entropy to calculate the loss as

$$\mathrm{LBCE} = -\sum_{0 \le i,j < n} y_{ij} \log\left(P_{ij}\right). \tag{22}$$

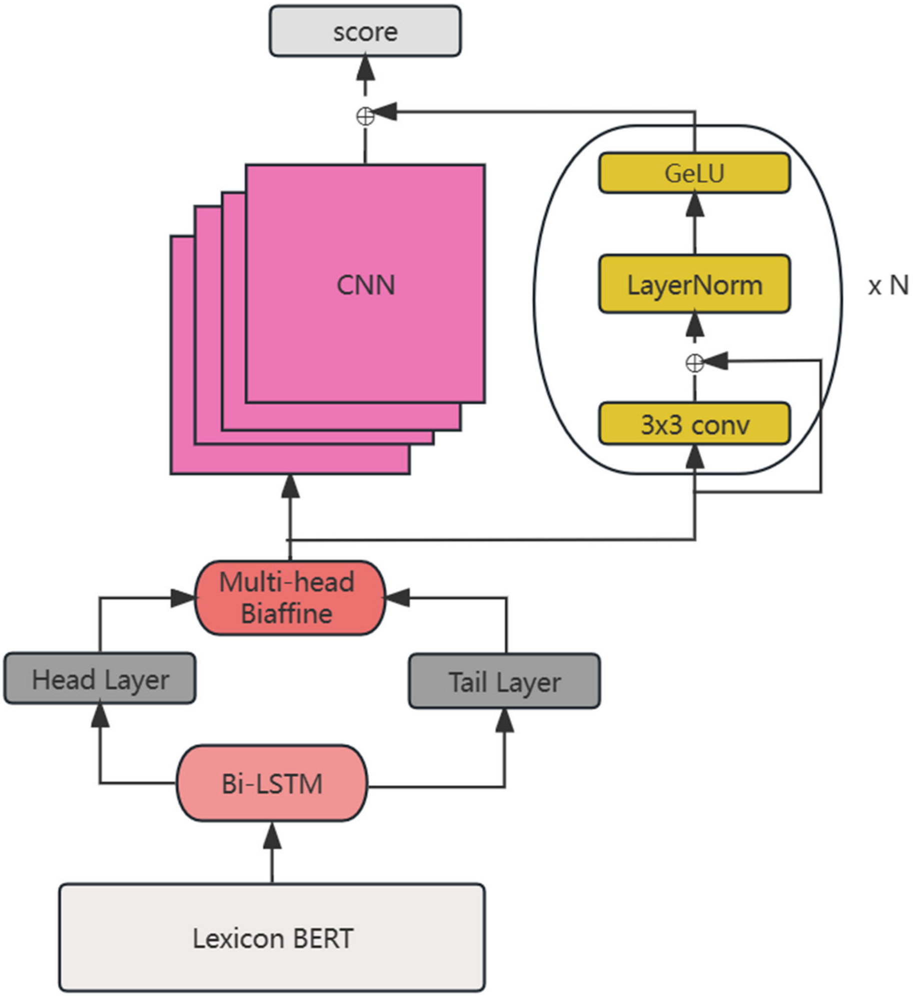


Fig. 6 The main architecture of LB-BMBC model

To facilitate batch processing, we cannot just compute the upper triangular part. We consider both the upper and lower triangular parts and output them. Since the labels of the score matrix are symmetric, the label for (i, j) is the same as the label for (j, i). During inference, we compute scores within the upper triangle segments as follows:

$$\hat{P_{ij}} = \left(P_{ij} + P_{ji}\right)/2. \tag{23}$$

We filter out non-entity spans (score < 0.5) and subsequently arrange the remaining spans in descending order based on their maximum entity scores. We then prioritize the selection of spans based on this order.

Algorithm 1 LB-BMBC

```
Input: $S$, $D$. $S$ is a Chinese sentence and $D$ is word vectors.
Output: The probabilities of each Span belongs to each entity.
Tranverse $D$ to construct a Trie tree;
Match $S$ to get the potential words from the Trie tree to get the Character–Words pair $S_W$;
$v_{ij} = W_2\left(tanh\left(W_1 x_{ij} + b_1\right)\right) + b_2$, Convert the word dimension to the character dimension;
$z_i = \sum_{j=1}^{p} a_{ij} v_{ij}, \alpha_i = softmax(h_i^l W_{attn} V_i^T)$, Assign weights to matching words;
$\widetilde{h_i^l} = h_1^l + \gamma z_i$, Introduce lexicon to the Transformer layer;
$H = Encoder_{Lexicon}\left(S_w\right)$;
$H_H, H_T = BiLSTM\left(H\right)$;
$Q = MHBiaffine(H_H, H_T)$;
$Q' = Conv2d\left(Q\right)$;
$Q'' = GeLU\left(LayerNorm\left(Q' + Q\right)\right)$;
$P = Sigmoid\left(W_o\left(Q + Q''\right) + b\right)$;
Loss fun: $LBCE = -\sum_{0 \leq i,j < n} y_{ij} log\left(P_{ij}\right)$;
Predict: $\hat{P_{ij}} = \left(P_{ij} + P_{ji}\right)/2$;
```

4 Experiments

4.1 Metrics

We adopt strict metric, and only when the entity boundary and entity type match are correct.

- TP: Correctly recognize entity boundaries and types.
- FP: The entity can be recognized but the category or boundary judgment is wrong.

- FN: No entity recognized.

$$\text{Precall}(P) = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{24}$$

$$\text{Recall}(R) = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{25}$$

$F1$ is used to balance P and R

$$F1 = \frac{2 \times P \times R}{P + R}. \tag{26}$$

4.2 Dataset

We have employed four prominent Chinese NER benchmark datasets: Resume [5], Weibo [15], OntoNotes 4.0 [16], and MSRA [17]. The statistics of the dataset is shown in Table 1.

- Resume: The Resume dataset is derived from filtering and manually annotating executive summary data from Sina Finance. This dataset encompasses 1027 executive summaries, with entity annotations distributed across eight distinct categories: CONT, EDU, LOC, PER, ORG, PRO, RACE, and TITLE.
- Weibo: The Weibo dataset is curated from historical data of Sina Weibo, covering the period from November 2013 to December 2014. This dataset comprises 1890 microblog messages, with entity annotations spanning four categories: PER, ORG, LOC, and GPE.
- OntoNotes 4.0: OntoNotes 4.0 is a Chinese dataset primarily sourced from the news domain. It encompasses entity annotations for four categories: GPE, LOC, ORG, and PER.
- MSRA: MSRA is a news-domain entity recognition dataset meticulously annotated by Microsoft Research Asia. It also serves as one of the datasets utilized in the SIGNAN backoff 2006 entity recognition task. This dataset encompasses over 50,000 Chinese entity recognition annotations, classified into three fundamental categories: ORG, PER, and LOC.

4.3 Hardware Environment and Experimental Parameters

Our model is implemented with Python interpreter version 3.8, and Pytorch 1.10, the GPU is an n NVIDIA GeForce RTX 3090.

Table 2 shows the hyper-parameter values of our model

4.4 Baselines

We conduct previous SoTA methods as baselines.

Table 1 Statistics of the Chinese datasets

Dataset	Type	Train	Dev	Test
Resume	Sent	3.8 k	0.46 k	0.48 k
	Char	124.1 k	13.9 k	15.1 k
Weibo	Sent	1.4 k	0.27 k	0.27 k
	Char	73.8 k	14.5 k	14.8 k
Ontonotes	Sent	15.7 k	4.3 k	4.3 k
	Char	491.9 k	200.5 k	208.1 k
MSRA	Sent	46.4 k	–	4.4 k
	Char	2169.9 k	–	172.6 k

Table 2 Model hyper-parameter settings

Hyper-parameter	Resume	Weibo	Ontonotes	MSRA
Batch size	16		8	
Seed	42			
Max seq len	256			
Optimizer	Adam			
Learning rate	1e-5			
Weight decay	0.01			
Warmup factor	0.1			
Max grad norm	1			
Word gamma	[0.5, 1.5]			
Max word num	[2, 4]			
Lexicon layer	[1, 3]			
Biaffine size	400			
Biaffine head	4			
CNN block	[2, 4]			
CNN dim	[120, 200]			

- Lattice-LSTM [5]: For the Chinese NER task, an LSTM model using a lattice structure, Lattice-LSTM, is proposed to encode both the character features of the input sequence and all potential words matched with the lexicon for NER after fusing the information of words and word sequences.
- CAN-NER [52]: Extract local character information through CNN, and then capture adjacent character or context information using a global self-attention layer composed of GRU.
- LR-CNN [53]: Used CNN to encode sentences, and proposes the rethinking mechanism to resolve lexical conflicts.
- LGN [54]: Introduce a dictionary-based graph neural network, modeling the Chinese NER problem as a node classification task in Solve the issue of ambiguous word boundaries in Chinese by employing an iterative aggregation mechanism.
- PLT [55]: Enhance self-attention through positional relationship representation, while also introducing a porous

mechanism to enhance local modeling and preserve the ability to capture extensive long-term dependencies effectively.

- FLAT [7]: A position encoding scheme has been designed to incorporate lattice structures for introducing lexicon information, and cross-domain relative position encoding has been proposed to make the Transformer suitable for NER tasks.
- softLexicon(LSTM) [56]: Proposing a simple yet effective method to incorporate the word lexicon into character representations by fine-tuning the character representation layer to introduce lexical information.
- MECT [57]: A two-stream Transformer coding model incorporating Chinese character structure features is proposed.

To analyze the contribution of each component in our model, we ablate the full model and demonstrate the effectiveness of each component:

- B-MB: The composition of the model is **B**ERT + **MHB**iaffine, and it is the most basic model that we will compare.
- B-BMB [47]: The composition of the model is **B**ERT + **B**iLSTM + **MHB**iaffine, Using BiLSTM to obtain the Head and Tail of the sentence and then feeding it into a Biaffine classifier.
- B-MBC: The model is composed of **B**ERT + **MHB**iaffine + **C**NN, which uses CNN to capture the spatial relations adjacent span.
- B-BMBC: The model is composed of **B**ERT + **B**iLSTM + **MHB**iaffine + **C**NN. We use BiLSTM to obtain the Head and Tail relationship of the sentence, and then send it to the Biaffine decoder to score the sentence. Finally, CNN is used to capture the spatial relationship adjacent span.
- LB-BMBC: Introduce **L**exical information in **B**ERT's Transformer layer and then concatenate **B**iLSTM + **MHB**iaffine + **C**NN.

4.5 Results and Discussion

The results on public datasets are shown in Tables 3 and 4, and we can observe that compared with other models, LB-BMBC model has the best performance in Resume (Precall 96.15%, Recall 94.44%, F1 96.29%), Weibo (Precall 65.01%, Recall 72.71%, F1 68.64%), Ontonotes (Precall 80.49%, Recall 82.22%, F1 81.35%), and MSRA (Precall 95.54%, Recall 95.45%, F1 95.50%).

Ablation Study All the components of our model play an important role in improving performance. If any component is missing, then the performance will decrease. We also conducted additional experiments on LB-BMBC with ablation consideration

- B-MB: Compared with the LB-BMBC model, the F1 score of the LB-BMBC model has decreased in different degrees (3.42% on Resume, 13.14%, 11.35% on Ontonotes, 3.77% on MSRA). Experiments on four datasets found that lexicon information and CNN Block play a key role in improving the performance of NER system.
- B-BMB: In this study, we add BiBLSTM to the B-MB model. Compared with the B-MB model, the F1 score have improved in different degrees(2.95% on Resume, 6.94% on Ontonotes, and 2.21% on MSRA). BiLSTM can capture the Head–Tail relationship of the sentence quite well.
- B-MBC: In this study, we add CNN Block to the B-MB model. Compared B-MB model, the F1 score has improved in different degrees(2.71% on Resume, 8.44% on Weibo, 8.49% on Ontonotes, and 3.33% on MSRA). After using the CNN Block to capture spatial information of adjacent spans, there was a significant improvement in the model's performance.
- B-BMBC: In this study, we add BiLSTM and CNN Block to the B-MB model. Compared with the B-MB model, the F1 score have improved in different degrees(3.02% on Resume, 9.97% on Weibo, 10.62% on Ontonotes, and 3.68% on MSRA). After applying BiLSTM to capture the Head–Tail relationships of the sentence, the model's performance further improved.
- LB-BMBC: In this study, the F1 score of the LB-BMBC model we proposed is the best (96.29%, 68.64%, 81.35%, 95.50%). Intecting lexicon information into the BERT alleviates the problem of word have multiple meanings and improves the recognition of entity boundaries.

4.6 The Impact of Different CNN Block Numbers

To investigate the effectiveness of CNN in modeling adjacent spans, furthermore, we conducted experiments with different numbers of CNN Blocks, as shown in Tables 5, 6, as illustrated in Fig. 7. By introducing CNN Blocks, more True Positives (TP) were correctly predicted, thereby improving Recall (R). When the number of CNN Blocks was set to 1, due to the insufficient capacity for modeling the span space, it resulted in some False-Positive (FP) predictions, thus reducing Precision (P). However, this can be rectified by increasing the number of CNN Blocks. In summary, introducing more than one CNN Block effectively enhances the model's recognition capability. When using fewer CNN Block numbers, there may be a problem of layer disappearance, which will make the network difficult to train. As the number of CNN Block numbers increases, the model's capacity increases,

Table 3 Experiment results (%) on Resume and Weibo

Models/datasets	Resume			Weibo		
	P	R	F1	P	R	F1
Lattice-LSTM	94.81	94.11	94.46	53.04	62.25	58.79
CAN-NER	95.05	94.82	94.94	55.38	62.98	59.31
LR-CNN	95.37	94.84	95.11	57.14	66.67	59.92
LGN	95.28	95.46	95.37	55.34	64.98	60.21
PLT	95.34	95.46	95.40	55.34	64.90	59.76
FLAT	–	–	95.45	–	–	60.32
softLexicon (LSTM)	95.34	95.64	95.59	56.99	61.41	61.24
MECT	96.40	95.39	95.89	61.91	62.51	63.30
B-MB	89.34	**96.69**	92.87	47.66	66.43	55.5
B-BMB	95.22	96.44	95.82	59.17	41.30	48.65
B-MBC	95.47	95.77	95.62	54.41	74.54	63.94
B-BMBC	95.17	96.63	95.89	58.16	**74.88**	65.47
LB-BMBC (ours)	**96.15**	94.44	**96.29**	**65.01**	72.71	**68.64**

Across the four datasets (Resume, Weibo, MSRA, and Ontonotes) and under evaluation by three metrics (precision P, recall R, and F1 score F1), the standout model selections, highlighted in bold, are as follows: Resume Dataset: In Table 3, LB-BMBC emerges as the top performer for both P and F1, while B-MB is leading in R. Weibo Dataset: Also in Table 3, LB-BMBC takes the lead for P and F1, with B-BMBC excelling in R

Table 4 Experiment results(%) on Ontonotes and MSRA

Models/datasets	Ontonotes			MSRA		
	P	R	F1	P	R	F1
Lattice-LSTM	76.35	71.56	73.88	93.57	92.79	93.18
CAN-NER	75.05	72.29	73.64	93.53	92.42	92.97
LR-CNN	76.40	72.60	74.45	94.50	92.93	93.71
LGN	76.40	72.60	74.45	94.19	92.73	93.46
PLT	76.78	73.68	74.89	94.25	92.30	93.26
FLAT	–	–	76.45	–	–	94.12
softLexicon (LSTM)	77.31	73.85	75.54	93.56	93.44	93.50
MECT	77.57	76.27	76.92	94.55	94.09	94.32
B-MB	61.39	81.62	70.00	88.76	94.91	91.73
B-BMB	76.19	77.70	76.94	94.64	93.25	93.94
B-MBC	73.97	**83.59**	78.49	94.34	**95.79**	95.06
B-BMBC	78.71	82.64	80.62	95.32	95.51	95.41
LB-BMBC (ours)	**80.49**	82.22	**81.35**	**95.54**	95.45	**95.50**

Ontonotes Dataset: Turning to Table 4, LB-BMBC is the premier choice for P and F1, and B-MBC tops for R. MSRA Dataset: In Table 4 again, LB-BMBC reigns supreme in P and F1, with B-BMBC leading in R

which means that the model has more parameters to learn the features of the data. Therefore, it can provide the model's expressive ability to some extent, thereby improving performance.

5 Conclusion

Named Entity recognition is one of the important tasks in information extraction within the field of natural language processing, playing a crucial role in downstream tasks such as knowledge graphs and question-answering systems. Chinese NER faces even greater challenges compared to English NER due to complexities like word segmentation and intricate grammatical structures in the Chinese language. In this paper, we propose the LB-BMBC model, which incorporates lexical information into the Transformer layers of BERT, allowing for substantial interaction between lexicon information and individual characters. Additionally, we model the spatial relationships between adjacent spans by introducing a CNN after Biaffine. Using our proposed method, we validate its effectiveness on four Chinese NER datasets, out-

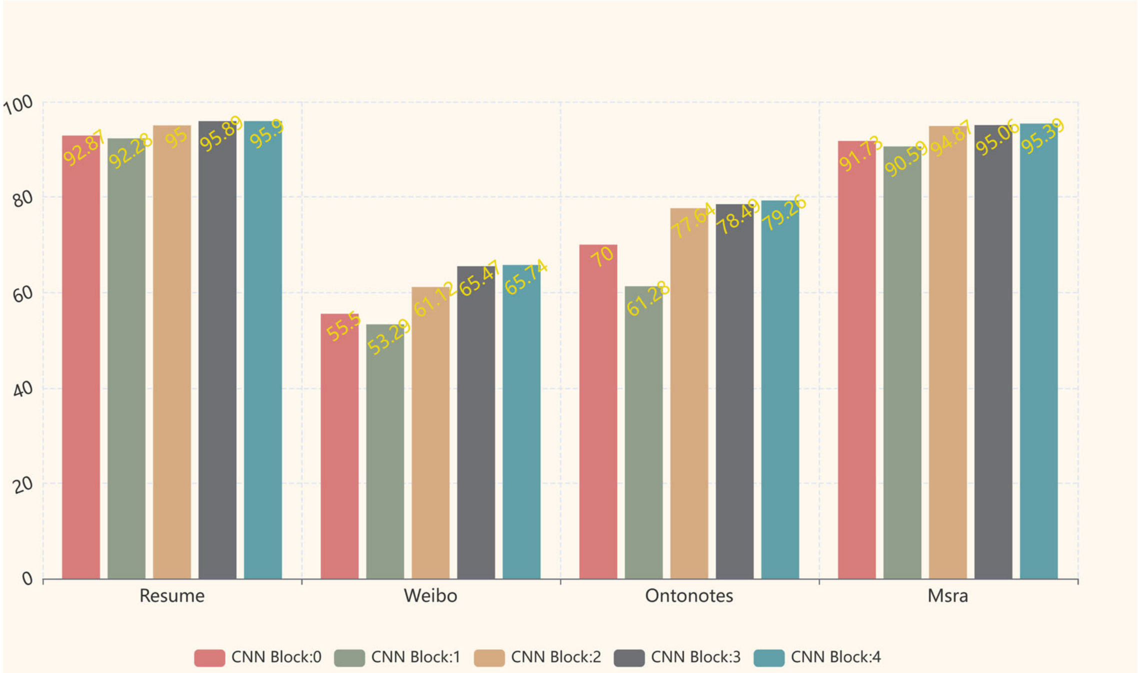


Fig. 7 F1 (%) for different numbers of CNN Blocks in Resume, Weibo, Ontonotes, and MSRA

Table 5 Different CNN Block numbers experiments results (%) on Resume and Weibo

Models/datasets	Resume			Weibo		
	P	R	F1	P	R	F1
CNN Block:0	89.34	96.69	92.87	47.66	66.43	55.50
CNN Block:1	88.16	**96.81**	92.28	41.82	73.43	53.29
CNN Block:2	93.36	96.69	95.00	51.40	**75.36**	61.12
CNN Block:3	95.17	96.63	95.89	58.16	74.88	65.47
CNN Block:4	**95.57**	96.75	**95.90**	**59.04**	74.15	**65.74**

Bold value indicates the optimal CNN Blocks for maximum P, R, and F1 are 4, 1, and 4, respectively. The Weibo dataset's peak performances correlate with CNN Block counts of 4 for P, 2 for R, and 4 for F1.

Table 6 Different CNN Block numbers experiments results (%) on Ontonote and MSRA

Models/datasets	Ontonote			MSRA		
	P	R	F1	P	R	F1
CNN Block:0	61.39	81.62	70	88.76	94.91	91.73
CNN Block:1	46.72	**88.99**	61.28	85.85	**95.88**	90.59
CNN Block:2	70.99	85.66	77.64	94.09	95.67	94.87
CNN Block:3	73.97	83.59	78.49	94.34	95.79	95.06
CNN Block:4	**75.32**	83.63	**79.26**	**94.97**	95.82	**95.39**

Both the Ontonotes and MSRA datasets, see their best scores across P, R, and F1 achieved with consistent CNN Block configurations of 4, 1, and 4 respectively, as detailed in Table 6

performing other lexical information models. The efficacy of our proposed approach is further confirmed through extensive ablation experiments.

In our future work, we will consider the following three aspects:

- Behind the visual form of Chinese characters lies rich linguistic information. For instance, characters like “液” (liquid), “河” (river), and “湖” (lake) all share the semantic element “氵” (water), indicating their semantic association with water. Intuitively, leveraging the visual form of Chinese characters could enhance Chinese NLP capabilities. We will explore the integration of character forms into the underlying layers of BERT.
- Chinese characters of ten exhibit polysemy, where a single character can have multiple meanings. For example, the character “乐” has two distinct pronunciations, “乐” (yue) representing music and “乐” (le) representing happiness. We will consider incorporating Chinese character phonetic information (pinyin) into the underlying layers of BERT to address this phenomenon.
- While applying CNN to model the spatial relationships between adjacent spans, both non-entity and entity spans are currently treated uniformly. We plan to enhance the modeling of entity spans' spatial relations by applying specific techniques to these spans, acknowledging their distinct nature.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant No. 61901030); the Fundamental Research Funds for the Central Universities (Grant No. 06500172).

Author Contributions The corresponding author provides financial support and conducts post-paper editing, while the first author proposes ideas and carries out experiments. Together, they collaborate on the writing of the paper.

Data Availability All datasets used in this study are publicly available.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Ethical and Informed Consent for Data Used Not applicable.

References

1. Ji, B., Yu, J., Li, S., Ma, J., Wu, Q., Tan, Y., Liu, H.: Span-based joint entity and relation extraction with attention-based span-specific and contextual semantic representations. In: Proceedings of the 28th International Conference on Computational Linguistics. International Committee on Computational Linguistics, Barcelona, Spain (Online) (2020)
2. Yu, Y., Wang, Y., Mu, J., Li, W., Jiao, S., Wang, Z., Lv, P., Zhu, Y.: Chinese mineral named entity recognition based on bert model. Expert Syst. Appl. **206**, 117727 (2022)
3. Liu, Y., Wei, S., Huang, H., Lai, Q., Li, M., Guan, L.: Naming entity recognition of citrus pests and diseases based on the bert-bilstm-crf model. Expert Syst. Appl. **234**, 121103 (2023)
4. Xi, Q., Ren, Y., Yao, S., Wu, G., Miao, G., Zhang, Z..: In: Jia, Y., Gu, Z., Li, A. (eds.) Chinese Named Entity Recognition: Applications and Challenges, pp. 51–81. Springer, Cham (2021)
5. Zhang, Y., Yang, J.: Chinese NER using lattice LSTM. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1554–1564. Association for Computational Linguistics, Melbourne, Australia (2018)
6. Liu, W., Xu, T., Xu, Q., Song, J., Zu, Y.: An encoding strategy based word-character LSTM for Chinese NER. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 2379–2389. Association for Computational Linguistics, Minneapolis, Minnesota (2019)
7. Li, X., Yan, H., Qiu, X., Huang, X.: FLAT: Chinese NER using flat-lattice transformer. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6836–6842. Association for Computational Linguistics, Online (2020)
8. Liu, W., Fu, X., Zhang, Y., Xiao, W.: Lexicon enhanced Chinese sequence labeling using BERT adapter. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 5847–5858. Association for Computational Linguistics, Online (2021)
9. Nguyen, D.Q., Verspoor, K.: End-to-end neural relation extraction using deep biaffine attention. In: Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part I 41, pp. 729–738 (2019). Springer
10. Du, X., Jia, Y., Zan, H.: Mrc-based medical ner with multi-task learning and multi-strategies. In: Sun, M., Liu, Y., Che, W., Feng, Y., Qiu, X., Rao, G., Chen, Y. (eds.) Chinese Computational Linguistics, pp. 149–162. Springer, Cham (2022)
11. Fei, Y., Xu, X.: Gfmrc: a machine reading comprehension model for named entity recognition. Pattern Recogn. Lett. **172**, 97–105 (2023)
12. Guan, Z., Zhou, X.: A prefix and attention map discrimination fusion guided attention for biomedical named entity recognition. BMC Bioinform. **24**(1), 42 (2023)
13. Sun, L., Sun, Y., Ji, F., Wang, C.: Joint learning of token context and span feature for span-based nested ner. IEEE/ACM Trans. Audio Speech Lang. Process. **28**, 2720–2730 (2020)
14. Yan, H., Sun, Y., Li, X., Qiu, X.: An embarrassingly easy but strong baseline for nested named entity recognition. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 1442–1452. Association for Computational Linguistics, Toronto, Canada (2023)
15. Peng, N., Dredze, M.: Named entity recognition for Chinese social media with jointly trained embeddings. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 548–554. Association for Computational Linguistics, Lisbon, Portugal (2015)
16. Pradhan, S., Ramshaw, L., Marcus, M., Palmer, M., Weischedel, R., Xue, N.: CoNLL-2011 shared task: Modeling unrestricted coreference in OntoNotes. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, pp. 1–27. Association for Computational Linguistics, Portland, Oregon, USA (2011)
17. Levow, G.-A.: The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, pp. 108–117. Association for Computational Linguistics, Sydney, Australia (2006)
18. Zhang, H., Liu, Q.: Automatic recognition of Chinese personal name based on role tagging. Chin. J. Comput. **27**, 85–91 (2004)
19. Bikel, D., Schwartz, R., Weischedel, R.: An algorithm that learns what's in a name. Mach. Learn. **34** (1999)
20. Ya, Z.: Chinese and English basenp recognition based on a maximum entropy model. J. Comput. Res. Dev. (2003)
21. Zhang, Y., Xu, Z., Zhang, T.: Fusion of multiple features for Chinese named entity recognition based on crf model. In: Asia Information Retrieval Symposium, pp. 95–106 (2008). Springer
22. Takeuchi, K., Collier, N.: Use of support vector machines in extended named entity recognition. In: COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002) (2002)
23. Li, L.-S., Huang, D., Chen, C.-R., Yang, Y.-S.: Identification of location names from Chinese texts based on support vector machine. J. Dalian Univ. Technol. **47**, 433–438 (2007)
24. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: Proceedings of the Seventh Conference on

Natural Language Learning at HLT-NAACL 2003, pp. 188–191 (2003)

25. Feng, Y.-Y., Sun, L., Zhang, D.-K., Li, W.-B.: Study on the Chinese named entity recognition using small scale character tail hints. Tien Tzu Hsueh Pao/Acta Electron. Sin. **36**, 1833–1838 (2008)
26. Yan, Y., Wen, D., Wang, Y., Wang, K.: Named entity recognition in Chinese medical records based on cascaded conditional random field. J. Jilin Univ. (Eng. Technol. Ed.) **44**(6), 1843–1848 (2014)
27. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging (2015). ArXiv arXiv:1508.01991
28. Žukov-Gregorič, A., Bachrach, Y., Coope, S.: Named entity recognition with parallel recurrent neural networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 69–74. Association for Computational Linguistics, Melbourne, Australia (2018)
29. Yang, Q., Jiang, J., Feng, X., He, J., Chen, B., Zhang, Z.: Named entity recognition of power substation knowledge based on transformer-bilstm-crf network, pp. 952–956 (2020)
30. An, Y., Xia, X., Chen, X., Wu, F.-X., Wang, J.: Chinese clinical named entity recognition via multi-head self-attention based bilstm-crf. Artif. Intell. Med. **127**, 102282 (2022)
31. Wu, Y., Jiang, M., Lei, J., Qi, W.: Named entity recognition in Chinese clinical text using deep neural network. Stud. Health Technol. informat. **216**, 624–8 (2015)
32. Kong, J., Zhang, L., Jiang, M., Liu, T.: Incorporating multi-level cnn and attention mechanism for Chinese clinical named entity recognition. J. Biomed. Inform. **116**, 103737 (2021)
33. Strubell, E., Verga, P., Belanger, D., McCallum, A.: Fast and accurate entity recognition with iterated dilated convolutions. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2670–2680. Association for Computational Linguistics, Copenhagen, Denmark (2017)
34. Jiang, X., Ma, J., Yuan, H.: Named entity recognition in the field of ecological management technology based on bilstm-idcnn-crf model. Comput. Appl. Softw **38**(3), 134–141 (2021)
35. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach (2019). arXiv preprint arXiv:1907.11692
36. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: A lite bert for self-supervised learning of language representations (2019). arXiv preprint arXiv:1909.11942
37. Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J.: Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)
38. Chang, Y., Kong, L., Jia, K., Meng, Q.: Chinese named entity recognition method based on bert. In: 2021 IEEE International Conference on Data Science and Computer Application (ICDSCA), pp. 294–299 (2021). IEEE
39. Liu, Y., Wei, S., Huang, H., Lai, Q., Li, M., Guan, L.: Naming entity recognition of citrus pests and diseases based on the bert-bilstm-crf model. Expert Syst. Appl. **234**, 121103 (2023)
40. Gan, Y., Yang, R., Zhang, C., Jia, D.: Chinese named entity recognition based on bert-transformer-bilstm-crf model. In: 2021 7th International Symposium on System and Software Reliability (ISSSR), pp. 109–118 (2021). IEEE
41. Cai, X., Sun, E., Lei, J.: Research on application of named entity recognition of electronic medical records based on bert-idcnn-crf model. In: Proceedings of the 6th International Conference on Graphics and Signal Processing. ICGSP '22, pp. 80–85. Association for Computing Machinery, New York, NY, USA (2022)
42. Wang, J., Xu, W., Fu, X., Xu, G., Wu, Y.: Astral: adversarial trained lstm-cnn for named entity recognition. Knowl.-Based Syst. **197**, 105842 (2020)
43. Huang, H., Lei, M., Feng, C.: Hypergraph network model for nested entity mention recognition. Neurocomputing **423**, 200–206 (2021)
44. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880. Association for Computational Linguistics, Online (2020)
45. Guo, Q., Guo, Y.: Lexicon enhanced Chinese named entity recognition with pointer network. Neural Comput. Appl. **34**(17), 14535–14555 (2022)
46. Hu, Z., Ma, X.: A novel neural network model fusion approach for improving medical named entity recognition in online health expert question-answering services. Expert Syst. Appl. **223**, 119880 (2023)
47. Yu, J., Bohnet, B., Poesio, M.: Named entity recognition as dependency parsing. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6470–6476. Association for Computational Linguistics, Online (2020)
48. Ji, B., Yu, J., Li, S., Ma, J., Wu, Q., Tan, Y., Liu, H.: Span-based joint entity and relation extraction with attention-based span-specific and contextual semantic representations. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 88–99. International Committee on Computational Linguistics, Barcelona, Spain (Online) (2020)
49. Nguyen, L.: Implementing Bi-LSTM-based deep biaffine neural dependency parser for Vietnamese Universal Dependency parsing. In: Proceedings of the 7th International Workshop on Vietnamese Language and Speech Processing, pp. 60–63. Association for Computational Lingustics, Hanoi, Vietnam (2020)
50. Li, Y., Li, Z., Zhang, M., Wang, R., Li, S., Si, L.: Self-attentive biaffine dependency parsing. In: IJCAI, pp. 5067–5073 (2019)
51. 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: Carpuat, M., Marneffe, M.-C., Meza Ruiz, I.V. (eds.) Findings of the Association for Computational Linguistics: NAACL 2022, pp. 1863–1873. Association for Computational Linguistics, Seattle (2022)
52. Zhu, Y., Wang, G.: CAN-NER: Convolutional Attention Network for Chinese Named Entity Recognition. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 3384–3393. Association for Computational Linguistics, Minneapolis, Minnesota (2019)
53. Gui, T., Ma, R., Zhang, Q., Zhao, L., Jiang, Y.-G., Huang, X.: Cnn-based chinese ner with lexicon rethinking. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, pp. 4982–4988 (2019)
54. Gui, T., Zou, Y., Zhang, Q., Peng, M., Fu, J., Wei, Z., Huang, X.: A lexicon-based graph neural network for Chinese NER. 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), pp. 1040–1050. Association for Computational Linguistics, Hong Kong, China (2019)
55. Mengge, X., Yu, B., Liu, T., Zhang, Y., Meng, E., Wang, B.: Porous lattice transformer encoder for Chinese NER. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 3831–3841. International Committee on Computational Linguistics, Barcelona, Spain (Online) (2020)
56. Ma, R., Peng, M., Zhang, Q., Wei, Z., Huang, X.: Simplify the usage of lexicon in Chinese NER. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguis-

tics, pp. 5951–5960. Association for Computational Linguistics, Online (2020)

57. Wu, S., Song, X., Feng, Z.: MECT: Multi-metadata embedding based cross-transformer for Chinese named entity recognition. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 1529–1539. Association for Computational Linguistics, Online (2021)