IBNNER: A Biaffine Model-Based Chinese Nested Named Entity Recognition Method for Medical Texts

Ping Lu, Chongkun Shao, Shan Deng, Jiaying Zeng and Kaibiao Lin

***Abstract*—Medical named entity recognition (MNER) enables the automatic identification of key entities in the medical domain, facilitating the extraction of useful information from many medical texts. Consequently, MNER holds significant application value in medical information processing and clinical decision support. Existing approaches encounter challenges such as complex and diverse medical terminologies, contextual ambiguity, and the presence of nested named entities, all of which are pivotal for accurate information extraction. This paper proposes an Improved Biaffine-based Nested Named Entity Recognition Model (IBNNER) to address the above challenges. Firstly, IBNNER utilizes BERT pre-training to obtain token embeddings representations from nested medical datasets, employing two feed-forward neural network layers to learn the starting and ending information of entity fragments; Additionally, introducing a biaffine mapping technique with position Embeddings to generate a score matrix from the two hidden layers; Finally, IBNNER employ a Convolutional neural network (CNN) to enhance the spatial distribution of the score matrix and employ a suitable loss function for training in multi-classification tasks. Experimental results show that IBNNER achieves a recall of 69.12% on the CMeEE dataset, surpassing the BERT model by 5.04%. Moreover, when compared to the BiLSTM model on the CLUENER2020 dataset, IBNNER exhibits improvements of 7.37%, 10.51%, and 8.95% in terms of precision, recall, and F1-score, respectively.**

***Index Terms*—Medical; Named entity recognition; BERT; Biaffine**

I. Introduction

MEDICAL named entity recognition (MNER) is a critical application within the medical domain of artificial intelligence. Its goal is to autonomously detect and categorize medical terminology within clinical texts, including entities such as diseases, symptoms, medications, treatment methods, and surgical procedures. The importance of MNER lies in its potential to enhance healthcare information management and refine clinical decision-making processes. By automating the identification and annotation of medical jargon within patient records, MNER equips physicians with a streamlined means of comprehending and analyzing cases, thereby potentially improving diagnostic accuracy and the efficacy of treatment decisions. The automation also alleviates the manual burden of annotation and data entry, concurrently improving the quality and efficiency of medical information processing. Moreover, MNER's ability to convert unstructured medical narratives into structured data facilitates robust data mining and analysis, which are essential for advancing clinical research and decision-making. In essence, MNER is indispensable for the digital transformation of healthcare and clinical practice, as it not only elevates the processing of medical information but also drives the progression of healthcare AI, providing a reliable decision-support framework for medical professionals.

Despite its significance, MNER faces several challenges within the healthcare context:

(1) Diverse Terminology Expressions: Medical terms are often expressed in various forms, including abbreviations, synonyms, and alternative names, complicating the recognition process. For instance, the term "hypertension" might be abbreviated as "HBP," expressed synonymously as "high blood pressure," or referred to by an alternative name such as "essential hypertension." Therefore, it is crucial during the recognition process to consider these variations and accurately group them under the same entity;

(2) Ambiguity of Entities: The contextual nature of medical terminology introduces ambiguity, where a term may have multiple interpretations depending on its context. For instance, the phrase "blood loss" in the sentence "The patient experienced excessive blood loss during the surgery" could refer to intraoperative hemorrhage or postoperative bleeding. Thus, a nuanced contextual analysis is essential for the accurate identification of medical entities;

(3) Medical texts frequently contain nested entities, where one entity is embedded within another. For example, in the sentence “The hypertension patient was hospitalized for diabetes treatment,” there are two disease entities, “hypertension” and “diabetes,” as well as a person entity, “hypertension patient”. The disease entity “hypertension” is nested within the person entity “hypertension patient”. These nested structures can obscure entity boundaries and

Manuscript received February 29, 2024; revised October 18, 2024.

This work is supported by High Level Talent Project of Xiamen University of Technology under Grant (YSK20002R).

Ping Lu is a associate professor from School of Economic and Management of Xiamen University of Technology, Xiamen, 361024, China. (e-mail: luping@xmut.edu.cn).

Chongkun Shao is a postgraduate student from Department of Computer Science and Technology of Xiamen University of Technology, Xiamen, 361024, China. (e-mail: 1062008505@qq.com).

Shan Deng is a data analyst at Suzhou Museum, Suzhou, 215001, China. (corresponding author to provide phone: 13774684364; e-mail: 411658758@qq.com).

Jiaying Zeng is a teaching assistant at Guangzhou Huali College, Guangzhou, 511325, China. (e-mail: 913512146@qq.com).

Kaibiao Lin is a professor from Department of Computer Science and Technology of Xiamen University of Technology, Xiamen, 361024, China. (e-mail: kblin@xmut.edu.cn).

complicate recognition.

Previously, we proposed a feature-enhanced NER model called the Multi-Feature Fusion-based Named Entity Recognition Model (MFFNER)[1]. MFFNER addresses the underutilization of radicals and vocabulary in the medical domain NER by treating entity recognition as a sequence labeling problem, where the model assigns labels to each word in a given sentence. A consecutive subsequence of words with the same label is considered an entity. However, this approach cannot handle nested entities, as a word may correspond to multiple labels in nested entities. For example, "主动脉夹层" is a standard disease entity but also includes the body part entity "主动脉", where the word "主" must be labeled with two different tags simultaneously. To address the issue of assigning multiple categories to the same word in Nested Named Entity Recognition (Nested_NER), researchers have made various improvements to the models. Strakova et al.[2] approached Nested_NER as a sequence-to-sequence problem. They employed a decoder in the sequence-to-sequence model to gradually generate labels, allowing a word to belong to multiple categories. They achieved the best results on four different corpora. Shibuya et al.[3] proposed an objective function to train a neural model for Nested entity recognition. They treated the sequence of nested entities as the second-best path within their parent entity. The extraction of entities was performed iteratively, starting from the outermost layer. These two approaches modified the original sequence labeling model in terms of sequence generation and hierarchical annotation. However, the observed improvements in performance were not significant. Therefore, Yu et al.[4] proposed the biaffine model. proposed the biaffine model. This model uses two feed-forward neural network layers to obtain hidden layers with different heads and tails. It then generates a score matrix through biaffine mapping and ranks the entity fragments based on their scores, with the highest-scored fragment being the predicted entity. This approach is simple and effective, but it has the following issues:

(1) The biaffine model often enumerates all candidate entity fragments. However, when the text length becomes too long, the ratio of positive to negative samples becomes imbalanced, affecting the trainability of the neural network;

(2) The biaffine-based Nested_NER method only focuses on the entity information within the fragments, neglecting the spatial relationships in the score matrix.

The structure of this paper is as follows: Chapter 2 presents related work; Chapter 3 introduces the theoretical foundation of Nested_NER methods; Chapter 4 details the proposed IBNNER model; Chapter 5 presents comparative and ablation experiments; and Chapter 6 concludes this paper.

II. Related work

NER algorithms primarily encompass rule-based methods, machine learning-based methods (ML), and deep learning-based methods (DL). DL has become a mainstream approach in MNER due to its ability to automatically learn features from data, enabling the handling of more complex language structures. Medical texts often contain nested, discontinuous, and ambiguous entities, which present significant challenges for conventional NER methods. Currently, various improved algorithms have been developed that utilize DL to tackle complex named entity scenarios.

A. Rule-based methods

In the early stages of NER in the medical domain, practitioners used rule-based methodologies and pattern-matching techniques to identify medical entities. Each medical discipline exhibits unique vocabulary usage and adheres to specific sentence structures, tailored to convey specialized content. However, the initial information extraction systems were constrained by their simplicity, rendering them ineffective for extracting intricate information. To overcome these limitations, Canfield et al.[5] introduced a semantic framework that established a unified medical system with enhanced expressive capabilities, facilitating the extraction of entities from free-text echocardiography reports. Sager et al.[6] engineered a system that harnessed semantic methodologies to process clinical narrative texts, enabling the mapping of narrative medical records into a database representation. Furthermore, Hettne et al.[7] devised a method for the identification of small molecules and pharmaceuticals within textual contexts, predicated on a dictionary-based approach. This approach involved manually inspecting high-frequency terms and applying disambiguation rules for term preprocessing. Experimental results have demonstrated that this method improves identification accuracy with minimal loss in recall.

B. Machine learning-based methods

ML requires experts to annotate large volumes of medical data for supervised learning. Zhao[8] enhanced entity recognition in biomedical texts by leveraging unlabeled text corpora to compute word-based similarity, and proposed a Hidden Markov Model augmented with similarity smoothing. This innovation engendered a marked enhancement in the recognition of entities within biomedical texts. Ponomareva et al.[9] integrated Hidden Markov models with Conditional Random Fields (CRF) to devise a novel biomedical NER methodology. They employed two different strategies to integrate additional medical data and used different heuristic rules to correct errors in entity recognition. When evaluated on the GENIA dataset, their method yielded a 2% and 3% relative improvement in F1-score over standalone CRF and HMM models, respectively.

Unlike rule-based methods, ML does not require the development and maintenance of medical domain-specific templates and dictionaries. Instead, they rely on a large amount of pre-labeled medical data. However, obtaining large-scale medical datasets is often challenging due to their scarcity. Additionally, ML requires experts to extract features relevant to NER, which can be a complex and costly process.

C. Deep learning-based methods

In the medical domain, the application of DL to Named Entity Recognition (NER) was adopted relatively early. Li et al.[10] developed a biomedical NER system utilizing Recurrent Neural Networks (RNNs), which factor in predictive cues from antecedent nodes and exogenous contextual data to enrich the informational content of the system. In another study, Zhang et al. [11] integrated word and character representations through the application of Simple Lexicon, harnessed Bidirectional Long Short-Term Memory (BiLSTM) for encoding, and deployed CRF for decoding. This approach achieved an F1-score of 90.05% on the CCKS2020 dataset. Pan et al.[12] introduced algorithms such as Re-entity-based CRF, BiLSTM-CRF, and Lattice-LSTM, observing a performance enhancement across all methodologies post the incorporation of the Re-entity technique. Li et al.[13] proffered an entity recognition model for Chinese electronic medical records predicated on pre-trained models, employing a bespoke dictionary to refine word segmentation and bolster the efficacy of the masked language model during pre-training. They also introduced an attention mechanism to learn global features, which effectively improved the accuracy of medical entity recognition. Building on this research, Li et al.[14] incorporated adversarial training into the pre-trained model. By generating samples with word vectors and adversarial perturbations, followed by the application of dilated convolutions to ascertain word dependencies and forecast the ultimate sequence, they achieved an F1-score of 83.19% on the CCKS2019 dataset. Addressing the challenge of static vectors' incapacity to encapsulate polysemous words, Zhang[15] proposed a method for MNER by amalgamating Enhanced Language Representation with Informative Entities 2.0 (ERNIE2.0). This approach began with deriving word vectors from ERNIE2.0, followed by a soft attention mechanism to determine word weights, and concluded with predicting the label sequence. The empirical results substantiated the model's efficacy.

In addition, Huynh et al.[16] presented a graph-based semantic model for document similarity measurement, enriching document representation with semantic and structural information derived from large knowledge bases. Liu et al.[17] proposed RanSeNet for 3D semantic segmentation of large-scale point clouds, demonstrating the efficacy of direct point-to-point processing methods in handling complex spatial data. Yu et al.[18] introduced MFRAN, a model that fuses multi-head self-attention with multi-dimensional dilated convolution for multi-label text classification, showcasing improved performance over traditional methods. In the context of Chinese electronic medical records, Ding[19] combined LSTM with CRF to develop an entity recognition model capable of effectively mining entity information, underscoring the potential of deep learning models in healthcare informatics.

These studies underscore the potential of deep learning models, especially when integrated with pre-trained language models and task-specific optimizations, to advance medical text processing.

D. *Methods for complex entity recognition problems*

The complexities of medical data make conventional sequence labeling models ineffective for entity recognition, requiring specialized methodologies. Medical entities often encompass nested structures, discontinuities, and ambiguous types[20]. For example, the GENIA biomedical dataset contains up to 30% nested entities, highlighting the need for Nested_NER techniques.

Traditional NER methods can only handle independent entities and cannot address nested entities. In traditional models, each character in the input sentence is labeled, and a subsequence of adjacent characters with the same category can be treated as an entity. However, this approach fails when a character in a nested entity may have multiple labels. For example, “主动脉夹层” is both a standard disease and includes the body part “主动脉”. Therefore, the character “主” needs to be assigned to both labels simultaneously. To address this, Ju et al.[21] introduced a dynamic stacking model that enables multi-label assignment by increasing the depth of the encoder and decoder layers. This model, which stacks recognition from the inside out, achieved F1-scores of 74.7% and 72.2% on the GENIA and ACE2005 datasets, respectively. However, its sequential dependency on inner and outer entity identification risks compounding cascading errors. Sohrab et al.[22] countered this with a span-based model that enumerates all potential entity fragments and employs a neural network for classification and entity typing. Yu et al.[4] proposed a span-based biaffine model that uses two feed-forward neural network layers to create distinct head and tail representations, generating a score matrix through biaffine mapping to rank entity fragments. Despite its effectiveness, the biaffine model's overconfidence can impair performance. Zhu et al.[23] mitigated this with a boundary smoothing method, distributing an entity's probability to adjacent positions, thereby enhancing model accuracy. Yuan et al.[24] leveraged a triaffine attention mechanism to integrate various heterogeneous information, including characters, boundaries, and labels, for entity recognition, achieving state-of-the-art performance on multiple nested datasets. Su et al.[25] introduced a span-based Nested_NER method with global normalization to resolve inconsistencies stemming from the biaffine model's dual-module approach, leading to improved performance.

The biaffine model proposed by Yu et al.[4] typically enumerates all candidate entity fragments. However, when the text length is excessively long, the imbalanced ratio of positive and negative samples can affect the training of the neural network, and the spatial relationship between score matrices is overlooked. To address these issues, we propose an improved Nested_NER method based on the biaffine model. This method utilizes a multi-scale CNN to establish the relationship between score matrices and employs a loss function suitable for multi-classification tasks to address the issue of imbalanced positive and negative samples. The proposed method has shown excellent performance across multiple datasets.

III. Theoretical foundations

This section outlines the theoretical foundations of our proposed model. It encompasses a comprehensive exposition of the BERT model, which is instrumental in the BERT encoding layer, the affine transformation and score matrix that constitutes the biaffine layer, and the CNN utilized in the convolutional layer. Each component is carefully selected to address the unique challenges of medical text analysis, collectively forming a synergistic framework that enhances the model's efficacy in Nested_NER.

A. BERT model

The BERT model[26], introduced by Google in 2018, is a pre-trained model designed to enhance computers' understanding of natural language. As Shown in Fig. 1, BERT follows a two-phase learning process: pre-training and fine-tuning. This model employs the bidirectional encoder of the Transformer architecture to meticulously model each word within the input sequence, thereby capturing contextual semantics and generating bidirectional representations. BERT effectively overcomes several limitations of traditional text representation models, including static representations, limited expressive capabilities, and a lack of domain-specific characteristics. Its bidirectional encoding strategy provides a more nuanced and contextually rich representation, which is particularly beneficial for tasks involving complex language structures and nuanced semantic understanding.

During the pre-training phase, BERT employs two tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). In the MLM task, a portion of the input sequence is randomly masked (replaced with “[MASK]”), and the BERT model is trained to predict the original words at the masked positions. To address the mismatch between training and prediction data, BERT adopts a strategy where 15% of the words are selected for masking: 80% of these are replaced by '[MASK]', 10% are left unchanged, and the remaining 10% are randomly replaced with another word. This approach significantly bolsters BERT's capacity to encode contextual information. In the NSP task, BERT predicts whether two sentences in the input are consecutive. The input consists of two sentences separated by the “[SEP]” token. In 50% of cases, the input sentences are semantically coherent, while in the remaining 50%, two random sentences are selected. By placing the “[CLS]” token at the beginning, BERT predicts the continuity of the two sentences: If the value of the “[CLS]” token is 0, the two sentences are consecutive; if the value is 1, they are non-consecutive. This binary classification enables BERT to develop an understanding of sentence context and relationships within the given text.

Upon completion of the pre-training phase, BERT has acquired comprehensive representations of words and sentences, including an understanding of their semantic relationships. The model is subsequently fine-tuned using data specific to downstream tasks. BERT is equipped to handle four primary fine-tuning task types: sentence pair classification, single sentence classification, text question answering, and single sentence tagging. As a pre-trained model, BERT has the following advantages:

(1) Unlike the unidirectional pre-training model GPT, BERT is a bidirectional pre-trained model, providing stronger semantic understanding capabilities;

(2) BERT's introduction has streamlined the natural language processing (NLP) field, which previously encompassed a myriad of models tailored to various domains. By providing a unified model, BERT conserves resources and enhances operational efficiency through its pre-training and fine-tuning methodology;

(3) BERT's flexibility allows it to excel across a wide range of tasks, often achieving performance that rivals or exceeds specialized models designed for specific applications.

B. Affine transformation

Affine transformation is a well-established technique in computer vision, used to transform vector spaces through linear operations and translations while preserving the original space's “flatness” and “parallelism”. This process ensures the maintenance of key information's proportional relationships. The affine mechanism has since been integrated into NLP, especially for tasks requiring precise contextual semantic representation. In NLP, the biaffine mechanism was initially implemented in syntactic parsing[27], yielding promising results across datasets from six distinct languages. Yu et al.[4] were the first to adapt the biaffine mechanism for NER tasks, achieving state-of-the-art performance on several datasets. The biaffine model operates by learning two separate affine transformation matrices that project the input contextual vectors and the representation vectors of each word onto a two-dimensional plane. This transformation effectively models inter-word relationships, enhancing label prediction precision. Compared to traditional NER models based on Recursive Neural Networks (e.g., LSTM), the biaffine model demonstrates superior accuracy and robustness, especially in handling long-range dependencies and complex text structures.

C. Score matrix

A score matrix is a key tool used to quantify the similarity or correlation among entities or patterns. Within the domain of NER, score matrices are conventionally utilized to illustrate the associative strength between disparate words or characters within a sentence. In the context of Nested_NER tasks, score matrices serve a crucial role in modeling the interrelations among distinct labels. Each matrix element encapsulates a score that signifies the correlation between a pair of labels, which can denote adjacency, nestedness, or the extent of association between said labels. Furthermore, the application of score matrices extends beyond NER, finding utility in a myriad of disciplines including, but not limited to, natural language processing, image processing, computer vision, recommendation systems, medical research, and bioinformatics. Score matrices are versatile because they provide a structured representation of relational data, facilitating complex analyses and enhancing model predictive capabilities.

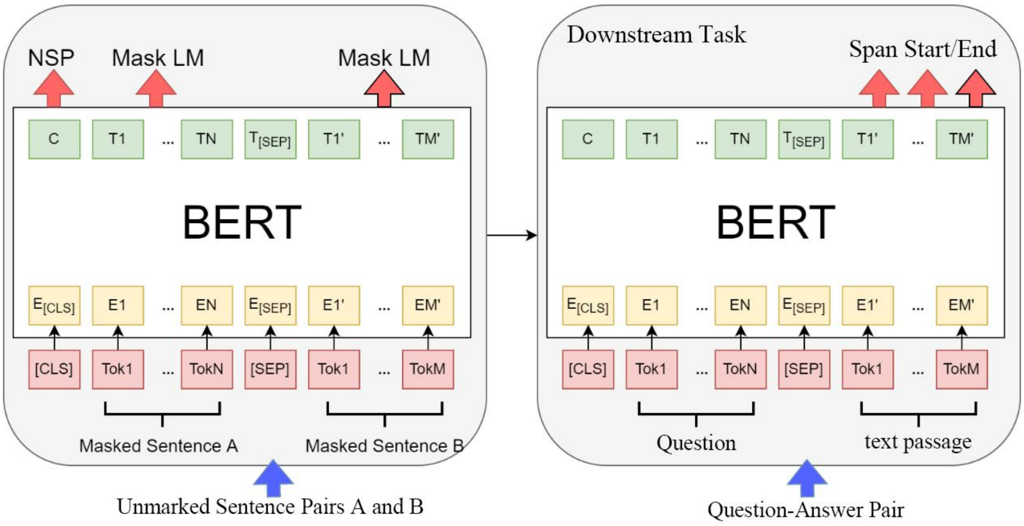


Fig. 1. Pre-training and Fine-tuning of BERT

D. *Convolutional neural networks*

CNN is adept at extracting local features from input sequences and encapsulating contextual information. In this study, a two-dimensional CNN is used to process the score matrix. The CNN processes the score matrix through convolutions, normalizations, and activation functions, extracting local features and facilitating intra-matrix interaction. Since adjacent regions in the score matrix often exhibit correlation, meaning that scores between neighboring segments may be similar or related, using two-dimensional convolution operations allows for capturing spatial correlations within the score matrix and better modeling the dependency relationships between segments. The convolutional operations within the CNN automatically isolate relevant local features from the input matrix. These extracted features are then utilized to augment the model's comprehension and representation of the input data. The subsequent application of an activation function, such as the Gaussian Error Linear Unit (GELU), transforms the linear input combinations into nonlinear outputs. This nonlinear transformation augments the model's expressive power and adaptability. Furthermore, the implementation of normalization techniques, such as Layer Normalization, serves to bolster the model's stability and generalization capabilities. Normalization of the output matrix adjusts the value scale and mitigates the variance in data, thereby optimizing the model's training process and bolstering its generalization performance.

IV. IBNNER MODEL

The overall structure of the IBNNER model, as shown in Fig. 2, consists of several components: the BERT encoding layer, affine transformation layer, convolutional layer, and output layer. Positional information is crucial, simply using the initial positional information from the BERT model is insufficient. Therefore, relative position functions are used in the score matrix computation to differentiate positions, and this information is added to the score matrix. To leverage the correlation between adjacent positions in the score matrix, a CNN is applied to enhance the model's performance.

(1) BERT Encoding Layer: BERT is a pre-trained language model based on the Transformer architecture. It learns rich semantic representations of words and their contextual information. BERT is chosen for its superior ability to handle word sense disambiguation and contextual understanding. The input text undergoes processing such as word embedding and positional encoding, and is encoded by the BERT model to obtain contextual representations for each word;

(2) Affine Transformation Layer: After obtaining the contextual representations from the BERT encoding layer, the affine transformation is applied to capture the nested relationships between entities. Affine transformation is a matrix multiplication operation used to compute scores between entity pairs. By applying affine transformations to the contextual representations corresponding to each entity and calculating the scores, the relationships between entities are captured. Compared to traditional NER models based on RNNs, the biaffine model exhibits better accuracy and robustness in handling long-range dependencies and complex text structures;

(3) Convolutional Layer: The convolutional layer captures local features and contextual information by leveraging the spatial correlation between adjacent positions in the score matrix. The convolutional layer extracts local features by performing operations on local windows. The convolutional kernels learn representations of different sizes to capture nested entity structures of varying lengths;

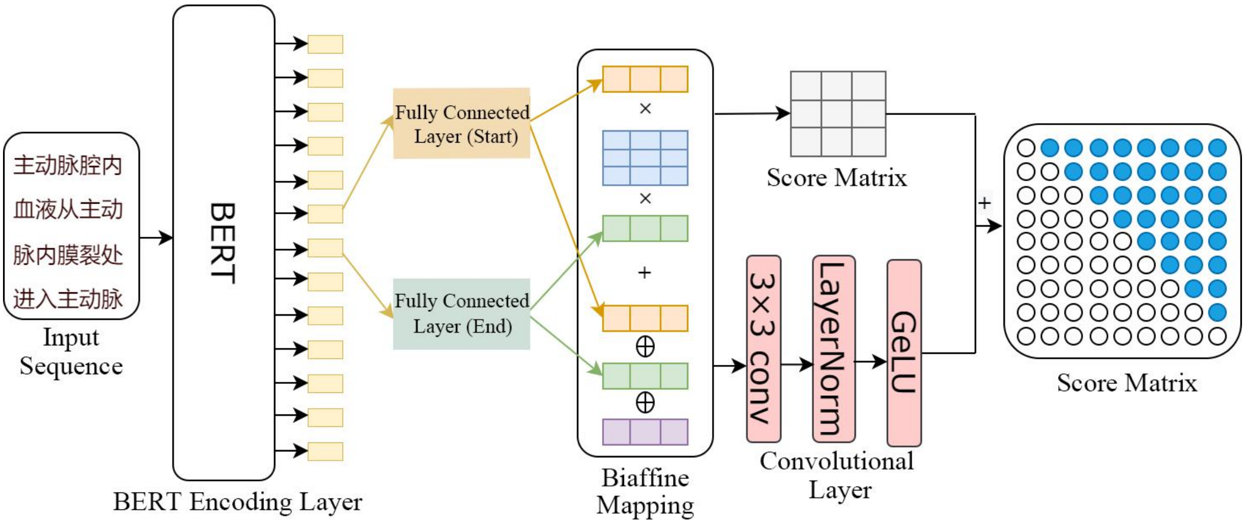


Fig. 2. Overall Framework of the IBNNER Model

(4) Output Layer: The output layer is responsible for computing the optimal label sequence for each position given the input sequence. The score matrix obtained from the CNN is used as the input to the output layer's CRF model. The CRF model captures dependencies between labels, including transition constraints between labels. The “I-ORG” label represents a continuous part of an organization entity, while the “B-GPE” label represents the starting part of a geopolitical entity. According to the rule constraints, the “I-ORG” label cannot directly follow the “B-GPE” label. By learning these label transition constraints, the CRF model automatically captures and follows these dependencies during prediction.

Next, we provide a detailed introduction to the building blocks of the IBNNER model.

A. Encoding layer

In this study, the BERT model, renowned for its proficiency in encoding rich semantic and contextual information, has been selected to serve as the encoder for the entity recognition task. The BERT model's input comprises three types of embeddings: Token Embeddings, Segment Embeddings, and Position Embeddings, as illustrated in Fig. 3:

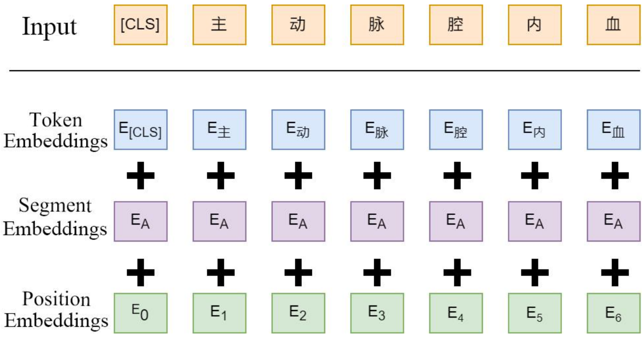


Fig. 3. Input Layer of BERT

(1) Token Embeddings: These embeddings represent the vectorial portrayal of each word within the input sequence. Typically derived from pre-trained word embedding models, or learned autonomously by BERT during its pre-training phase, these embeddings are pivotal for capturing the semantic essence of the text. The sequence is bookended by special tokens: the [CLS] token marks the sentence's inception, while the [SEP] token signifies its conclusion.

(2) Segment Embeddings: Designed to distinguish between distinct sentences within the input sequence, these embeddings assign a segment identifier to each word, indicative of its affiliation with either the first or second sentence. Words in the primary sentence are assigned a segment value of 0, whereas those in the subsequent sentence are designated a value of 1, thereby facilitating the differentiation of sentence origins.

(3) Position Embeddings: Given that BERT operates on a transformer architecture, which does not inherently process sequential order, Position Embeddings are crucial for encoding the positional information of each word. These embeddings offer a vectorized representation of a word's position within the sequence, with each token being paired with a unique Position Embedding. The expressive power of these position embeddings is stronger than the fixed formulas used by Transformers, as shown in Equations (1) and (2).

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \tag{1}$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \tag{2}$$

Where $PE_{(pos,2i)}$ represents the position encoding value of the 2ith element in the position encoding vector, while $PE_{(pos,2i+1)}$ corresponds to the position encoding value of the (2i+1)th element. pos represents the position of the current token, i represents the ith element of the position encoding vector, and d_{model} represents the dimension of the word vectors in the BERT model. The value 10,000 in the calculation formula for position embeddings is an empirical value that helps maintain reasonable distances between position encoding vectors for different positions. Additionally, the Sine and Cosine functions ensure the orthogonality between position encoding vectors, preventing confusion between different position encoding vectors. Finally, the Token Embeddings, Segment Embeddings, and Position Embeddings are added together to form the input for the next layer. These new vectors are modeled by the bidirectional encoder part of the transformer, which processes each word in the input sequence, resulting in a sequence of word embeddings. For a given input sentence $X = [x_1, x_2, \dots x_t]$, encoding X with BERT yields a sequence of word embeddings H, where $H \in R^{t \times d}$, and d represents the feature dimension of the hidden layers, as shown in Equation (3).

$$h_1, h_2, \dots h_t = Bert(x_1, x_2, \dots x_t) \tag{3}$$

B. Affine transformation layerr

Chinese Named Entity Recognition within the medical domain faces significant challenges due to the complexity and diversity of medical terminology, as well as the prevalence of nested entity structures. The nested nature of different entity types escalates the complexity of recognition tasks. Moreover, traditional models that rely on RNNs can encounter limitations when dealing with long-distance dependencies and the intricacies of complex textual structures.

To address these challenges, the biaffine model, which is characterized by its bidirectional and affine transformation properties, has been selected as the cornerstone of our approach. This model's architecture allows for the simultaneous consideration of contextual information and global dependencies within the text. By effectively capturing the semantic relationships and nested structures present in sentences, the biaffine model enhances the accuracy and robustness of entity recognition tasks. The biaffine model's ability to integrate both local context and long-range dependencies makes it particularly well-suited for the nuanced demands of medical NER. Consequently, this paper adopts the Biaffine model as the foundational framework for our entity recognition system.

Initially, the sequence of word embeddings H is input into two feed-forward neural network layers, denoted as $FFNN_s$ and $FFNN_e$, to obtain representations h_i^s and h_j^e representing the starting and ending positions, respectively. With these two distinct representations, we can learn information about the beginning and end of each entity mention. For a given entity mention starting at position i and ending at position j, we compute a score matrix r_{ij} to represent its score.

$$h_i^s = FFNN_s(HW_s) \tag{4}$$

$$h_j^e = FFNN_e(HW_e) \tag{5}$$

$$r_{ij} = (h_i^s)^T U h_j^e + W\left(h_i^s \oplus h_j^e \oplus w_{j-i}\right) + b \tag{6}$$

Where $U \in \mathbb{R}^{d \times c \times d}$, $W \in \mathbb{R}^{c \times (2d + d_w)}$, and $b \in \mathbb{R}^c$ are trainable parameters.

However, the Biaffine model encounters certain issues as it fails to capture sequential information and distinguish relationships and distances between characters. For example, consider the Chinese text “免疫力低下可引发气管炎、感染性腹泻等疾病”. We aim to extract the disease entities “气管炎” and “感染性腹泻”. However, due to the lack of positional information, the model easily mistakes “气管炎、感染性腹泻” as disease entities. This occurs because the model considers “气管” as the beginning of a disease entity and “腹泻” as the end of a disease entity. Hence, positional information is crucial, and relying solely on the initial positional information from the BERT model is insufficient. To address this issue, this paper introduces a relative position function w_{j-i} when calculating the score matrix r_{ij}, which helps differentiate different positions and incorporates relative positional information into the score matrix.

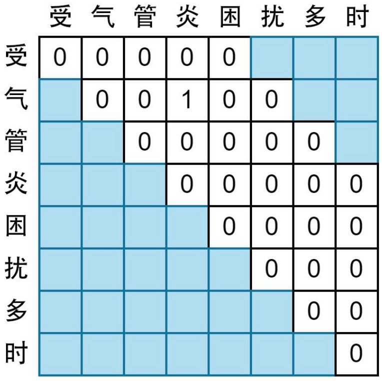
The biaffine model typically enumerates all candidate entity fragments. For a sequence of length n, there may be a total of n(n+1)/2 possible entities. In the score matrix, the value of each cell indicates the presence or absence of an entity, where 1 represents the presence of an entity, and 0 represents its absence. Taking the sentence “受气管炎困扰多时” as an example, we can obtain a score matrix where the vertical column corresponds to the starting position of the entity in the original text, and the horizontal column corresponds to the ending position. As shown in Fig. 4, white cells indicate potential entity positions, and the indices of the vertical columns in all blue cells are smaller than those of the horizontal columns.

	受	气	管	炎	困	扰	多	时
受	0	0	0	0	0	0	0	0
气	🟦	0	0	1	0	0	0	0
管	🟦	🟦	0	0	0	0	0	0
炎	🟦	🟦	🟦	0	0	0	0	0
困	🟦	🟦	🟦	🟦	0	0	0	0
扰	🟦	🟦	🟦	🟦	🟦	0	0	0
多	🟦	🟦	🟦	🟦	🟦	🟦	0	0
时	🟦	🟦	🟦	🟦	🟦	🟦	🟦	0

Fig. 4. The score matrix for the sentence “受气管炎困扰多时”

Due to the constraint that the starting position of an entity cannot be smaller than the ending position, the blue cells in the lower-left part of the matrix indicate the absence of entities. In this example, “气管炎” is a type of disease entity, so the value in the fourth column of the second row corresponds to 1, while the remaining positions do not have any disease entities, hence the value is 0. Additionally, in practical scenarios, entities longer than 128 characters are not encountered. Therefore, the length of the score matrix is limited to 128. Due to the significant disparity in the number of positive and negative samples, the model described in the article restricts the entity length to the maximum length observed in the dataset. Consequently, positions beyond this length are not considered potential locations for entities, resulting in the blue cells in the upper-right part of the matrix indicating the absence of entities.

As shown in Fig. 5 for the sentence “受气管炎困扰多时”, assuming the maximum length of an entity is 5, the longest entity starting from “受” is “受气管炎困”. Therefore, the position of “扰多时” is not considered as an entity. On the other hand, the longest entity starting from “气” is “气管炎困扰”, so the position of “多时” is not considered an entity. By employing this approach to reduce a significant number of negative samples, the ratio between positive and negative samples becomes more balanced, thus enhancing the trainability of the neural network.



	受	气	管	炎	困	扰	多	时
受	0	0	0	0	0			
气		0	0	1	0	0		
管			0	0	0	0	0	
炎				0	0	0	0	0
困					0	0	0	0
扰						0	0	0
多							0	0
时								0

Fig. 5. The score matrix for “受气管炎困扰多时” after limiting the entity length

C. Convolutional layer

The convolutional layer is composed of multiple convolutional units and is used to extract different features from input data. The lower-level convolutional layers primarily extract low-level features, such as edge information, while the higher-level convolutional layers can extract more complex features, such as shapes. To leverage the correlation between adjacent positions in the score matrix, this article employs a CNN to perform interactive operations on the score matrix, aiming to improve the model's performance. The score matrix obtained from the output of the affine transformation layer serves as the input to the convolutional layer, denoted as r_{ij}, representing the score matrix for each $i-j$ segment. The complete score matrix for the input sentence is denoted as R . By applying convolutional operations to the score matrix R using a two-dimensional convolutional layer, a new matrix R' is obtained, which is then passed through an activation function to obtain R''. Zero-padding is employed during the convolutional operation, as shown in formulas (7) and (8).

$$R' = Conv2d(R) \quad (7)$$

$$R'' = GeLU\big(LayerNorm(R + R')\big) \quad (8)$$

Where $Conv2d$ represents a two-dimensional convolution, LayerNorm refers to normalization techniques, and GELU denotes the activation function. As shown in Fig. 6, this is an example of a multi-channel convolution operation. The input from two channels undergoes convolution with their respective convolution kernels, and the results are summed to obtain the output of the convolution operation.

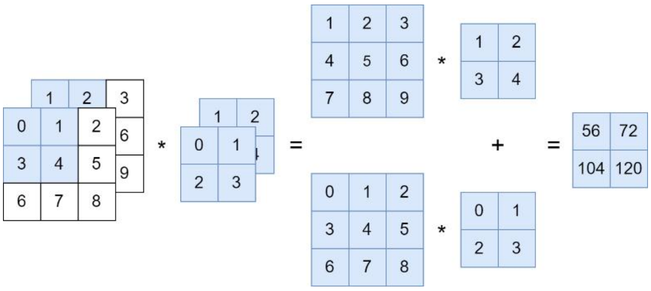


Fig. 6. Convolution operation

D. Output layer

The output layer determines the entity type through a loss function. For single-label classification problems, a common loss function is shown in formula (9), which is trained to maximize the score of the target class.

$$-log\frac{e^{s_t}}{\sum_{i=1}^{n} e^{s_i}} = log\left(1 + \sum_{i=1,\ i\neq t}^{n} e^{s_i - s_t}\right) \quad (9)$$

Where $s_1, s_2, \ldots s_n$ represents the scores for each class, and s_t represents the set of target classes. In multi-label classification problems, a common approach is to convert them into n binary classification problems. For each class, the sigmoid function is used to transform its value between 0 and 1, and binary cross-entropy is employed to calculate the loss. The losses for each class are then summed. However, in this case, less-represented classes may lose their advantages, and in extreme situations, they might not learn effectively. To address this issue, a common approach is to introduce weight factors to assign different weight proportions to positive and negative samples. However, determining the weight threshold can be cumbersome, requiring multiple parameter adjustments to obtain the optimal value. Therefore, the model introduces the Focal Loss function[28], which reduces the loss weight of correctly classified samples, enabling the model to focus more on difficult-to-classify samples. This mechanism helps the model better learn minority or more challenging samples, making Focal Loss potentially more effective on imbalanced datasets. Assuming that the label of the i-th word in a sentence is y_i, and the model's predicted probability for label y_i is p_{y_i} , the formula for Focal Loss is shown in formulas (10):

$$FL(p_{y_i}) = -\alpha_{y_i}(1 - p_{y_i})^{\gamma} log(p_{y_i}) \quad (10)$$

Where α_{y_i} is the weight for category y_i,and γ is the focusing parameter. Each predicted entity category in the Focal Loss has a corresponding loss function, and the final loss value is obtained by averaging or summing the losses across all words in the entire sequence.

V. Experiment

This chapter is dedicated to demonstrating the effectiveness of the IBNNER model in the task of medical text NER. Experiments will be conducted across four datasets: the specialized CMeEE[29] dataset and the general CLUENER2020[30] dataset, the ERTCMM dataset focused on Traditional Chinese Medicine, and the MSRA[31] dataset, which pertains to the news domain. These datasets will help assess the model's performance in both niche, broader, and cross-domain contexts. Additionally, ablation studies will be performed to dissect the impact of key model components, thereby informing further refinements to the model's architecture.

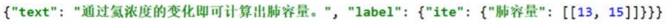
A. Dataset

(1) CMeEE Dataset[29]

The CMeEE (Chinese Medical Entity Extraction) dataset was utilized in this experiment. Comprising 20,000 medical texts, this dataset features nested named entities derived from the domain of medicine. The corpus, annotated and extracted from "Clinical Pediatrics," encompasses nine distinct entity types: diseases (dis), symptoms (sym), drugs (dru), medical equipment (equ), medical procedures (pro), body parts (bod), medical test items (ite), microbiology (mic), and departments (dep). The dataset has been meticulously partitioned into a training set, validation set, and test set, adhering to a 6:2:2 ratio, respectively. A detailed breakdown of the dataset is delineated in TABLE I. Notably, the "symptom(sym)" entity type is capable of nesting the other eight entity types; in contrast, the remaining entity types are non-nested. A representative example from the dataset is depicted in Fig. 7, where "text" corresponds to the original medical record, and "label" corresponds to the associated entity labels. Specifically, in the example provided, the entity "肺容量" (lung capacity) is categorized under "ite" and is annotated from positions 13 to 15 within the text.

TABLE I
CMeEE DATASET

Entity type	dis	sym	pro	equ	dru	ite	bod
Entity count	20778	16399	8389	1126	5370	3504	23580



```
{"text": "通过氦浓度的变化即可计算出肺容量。", "label": {"ite": {"肺容量": [[13, 15]]}}}
```

Fig. 7. The data format of the CMeEE dataset

(2) CLUENER2020 Dataset[30]

To evaluate the performance of the model on a non-nested dataset, we selected the CLUENER2020 dataset for Chinese fine-grained named entity recognition. CLUENER2020 is annotated based on the THUCTC dataset and includes ten types of entities: organization, name, address, company, government, book, game, movie, position, and scene. The dataset consists of 10,748 training samples, 1,343 validation samples, and 1,345 test samples. Part of the data is shown in TABLE II.

TABLE II
CLUENER2020 DATASET

Entity type	name	address	company	book	game	movie	scene
Entity count	3661	2829	2897	1131	2325	1109	1462

(3) ERTCMM Dataset

The ERTCMM (Entity Recognition of Traditional Chinese Medicine's Manual) dataset is derived from Traditional Chinese Medicine drug manuals, comprising a total of 1500 deduplicated drug instructions. This dataset defines 13 entity categories: Drug , Drug Ingredient, Disease, Symptom, Syndrome, Disease Group, Food, Food Group, Person Group , Drug Group, Drug Dosage, Drug Taste, and Drug Efficacy.

(4) MSRA Dataset

To demonstrate that our model performs well in other domains, we also conducted experiments on the MSRA dataset. The MSRA dataset, annotated by Microsoft Research Asia, is a Chinese named entity recognition dataset tailored for the news domain, comprising 50,729 annotated Chinese entities. The dataset defines three entity categories: Person (PER), Location (LOC), and Organization (ORG).

B. *Experimental environment and parameters setting*

In this paper, PyTorch 1.7.0 was used as the framework for this experiment, with Python version 3.6. The training environment configuration for the experiment is shown in TABLE III.

TABLE III
TRAINING ENVIRONMENT CONFIGURATION

Parameter	Value
CPU	AMD Ryzen 7 2.9GHz
GPU	NVIDIA GeForce GTX 1650Ti
Python	3.6
Pytorch	1.7.0

During the training process, considering the convergence speed of the model, a learning rate of $2e^{-5}$ was chosen. The batch size was set to 4, and the number of epochs was set to 50. A Decay_rate of 0.999 and Decay_steps of 100 were applied. The reasonableness of the parameter settings was verified through multiple experiments. Additional parameters are shown in TABLE IV.

TABLE IV
PARAMETERS SETTING

Parameter	Value
Transformer	12
Decay_rate	0.999
Decay_steps	100
Batch_size	4
Epoch	50
Hidden layer dimensions	768
Optimizer	Adam

C. *Evaluation Criteria*

To accurately identify medical named entities, it is necessary to accurately determine the boundaries of entities and correctly identify their respective categories. In this chapter, recall (R), precision (P), and F1-score are used to evaluate the model's recognition performance, which are common evaluation metrics for model classification performance. Taking binary classification as an example, TABLE V and the corresponding formulas are used to illustrate the calculation of these three metrics.

TABLE V
MODEL'S GROUND TRUTH AND PREDICTED VALUE

Ground Truth Value /Predicted Value	Positive	Negative
Positive	True positive (TP)	False negative (FN)
Negative	False positive (FP)	True negative (TN)

Precision is a metric that measures the accuracy of a model in identifying positive samples. It represents the ratio between the number of correctly predicted positive samples and the total number of samples predicted as positive, as shown in Formula 12.

$$P = \frac{TP}{TP + FP} \tag{12}$$

Recall is a metric that measures the ability of a model to identify positive samples. It represents the ratio between the number of correctly predicted positive samples and the total number of actual positive samples, as shown in Formula 13.

$$R = \frac{TP}{TP + FN} \tag{13}$$

F1-score combines the performance of recall and precision. It is the harmonic mean of the two and measures the model's ability to balance between identifying positive and negative samples. F1-score is used to evaluate the overall performance of the model. The calculation of the F1-score is shown in Formula 14.

$$F1 = \frac{2 * P * R}{P + R} \tag{14}$$

D. Comparative Experiment

This section analyzes the experimental results of the IBNNER model and the comparative models on the CmeEE, CLUENER2020, ERTCMM, and MSRA datasets. The evaluation metrics include precision, recall, and F1-score.

TABLE VI
THE VALUES OF THE CMEEE DATASET FOR DIFFERENT MODELS

Model	P(%)	R(%)	F1(%)
BERT-CRF[26]	58.34	64.08	61.07
Lattice-LSTM[32]	57.10	43.60	49.44
BERT-Biaffine[33]	64.17	61.29	62.29
TPORE[34]	63.73	66.25	64.97
RICON[35]	66.25	64.89	65.57
Lattice-LSTM+Med-BERT[36]	56.84	47.58	51.80
BERT-MRC[37]	**67.86**	61.46	64.50
IBNNER	64.12	**69.12**	**66.53**

Based on the results from the CMeEE dataset, as shown in TABLE VI, the following conclusions can be drawn:

(1) IBNNER outperforms Lattice-LSTM[32] significantly in terms of precision, recall, and F1-score. Lattice-LSTM is an LSTM-based model that leverages Chinese lexical information. However, its entity recognition ability is limited by LSTM's relatively weaker information encoding capability. As a result, our model achieves a 17.09% higher F1-score than Lattice-LSTM;

(2) IBNNER exhibits a 5.78% improvement in precision, a 5.04% improvement in recall, and a 5.46% improvement in F1-score compared to BERT-CRF[26]. BERT-CRF is a model that employs sequence labeling for entity recognition but struggles with effectively labeling entities with nested structures. The "symptom" entities in the CMeEE dataset contain a significant number of nested entities. The superior performance of our model is due to its span-based approach, which enumerates all possible entity fragments and predicts their boundaries and types. This approach easily handles nested entities. Additionally, BERT-CRF treats NER as a classification problem with K categories, requiring all entities to be correctly predicted. In contrast, our model uses biaffine transformation to divide possible entities into feasible parts, leading to more accurate predictions.

(3) Compared to other models, our model has a higher recall and F1-score, but the precision is not outstanding.

To provide a more intuitive representation of the model's performance, bar graphs depicting the precision, recall, and F1-score of each model on the CMeEE dataset are shown in Fig. 8 to 10. Although our model's precision is not outstanding, it demonstrates strong performance in recall and F1-score.

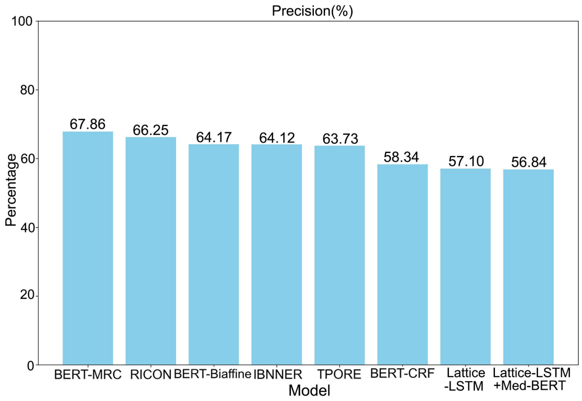


Fig. 8. Comparison of Precision among Models on CMeEE dataset

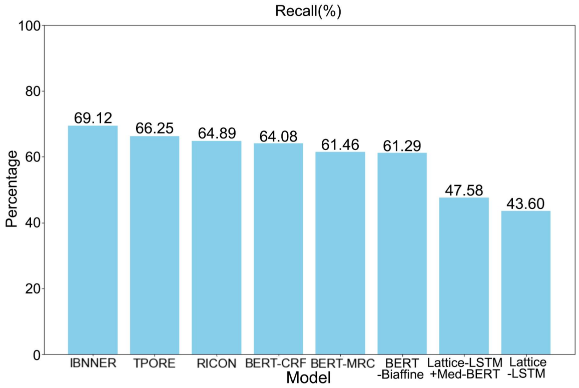


Fig. 9. Comparison of Recall among Models on CMeEE dataset

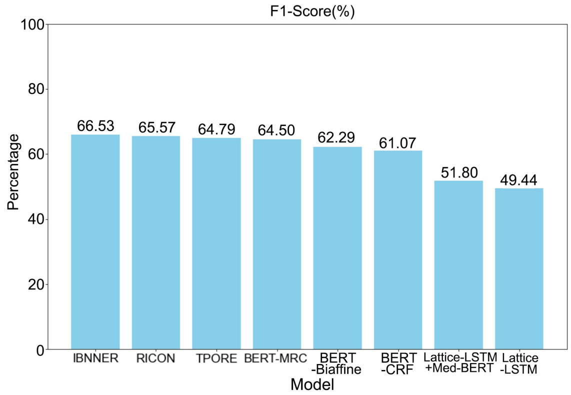


Fig. 10. Comparison of F1-score among Models on CMeEE dataset

To evaluate the performance of the model on a non-nested dataset, we conducted experiments using the CLUENER2020 dataset, which is a commonly used general-purpose dataset. The results are shown in TABLE VII.

TABLE VII
RESULTS ON CLUENER2020 DATASET

Model	P(%)	R(%)	F1(%)
BiLSTM-CRF[38]	71.06	68.97	70.00
BERT-CRF[26]	77.24	**80.46**	78.82
ALBERT[39]	79.92	64.59	71.07
ALBERT-CRF[40]	**80.94**	61.20	69.36
Human Performance[41]	65.74	62.17	63.41
IBNNER	78.43	79.48	**78.95**

The BiLSTM-CRF[38] model, a classic sequence labeling approach, uses BiLSTM for contextual semantic feature extraction followed by CRF for entity labeling. When evaluated on the CLUENER2020 dataset, this model achieves an F1-score of 70.00%. While the CRF imposes intrinsic constraints on label sequences, such as preventing an “I-ORG” label from following a “B-GPE” label, the biaffine model operates without these restrictions. Although the CRF layer can learn certain constraints during training, the IBNNER model, enhanced by BERT's advanced information extraction, effectively captures granular word-level and semantic features. This early feature capture reduces the demands on subsequent processing stages, resulting in strong outcomes. Therefore, our model outperforms BiLSTM-CRF by 7.37% in precision, 10.51% in recall, and 8.95% in F1-score. Although our model's F1-score is slightly lower than BERT-CRF, its similar precision suggests that it can also perform well on non-nested datasets. The comparisons with other models are shown in Fig. 11 to 13. Similar to its performance on the CMeEE dataset, our model exhibits notable performance in terms of recall and F1-score.

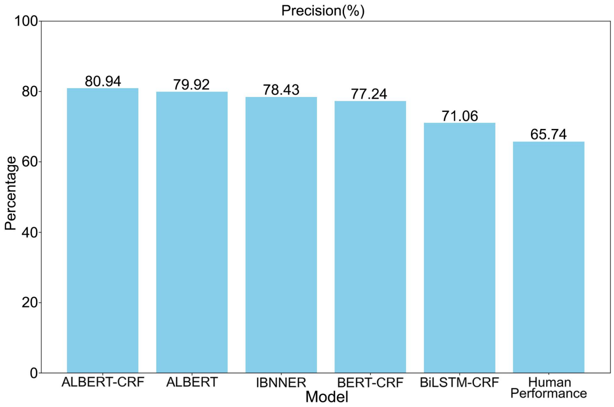


Fig. 11. Comparison of Precision among Models on CLUENER2020 dataset

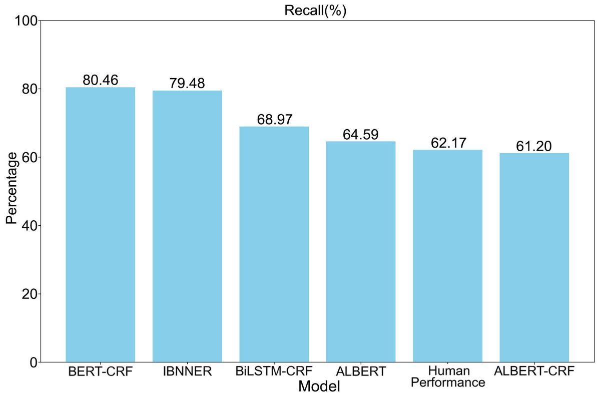


Fig. 12. Comparison of Recall among Models on CLUENER2020 dataset

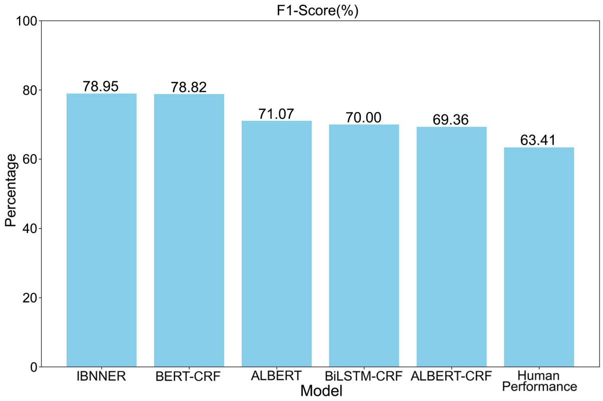


Fig. 13. Comparison of F1-score among Models on CLUENER2020 dataset

We conducted experiments on the ERTCMM dataset and compared our model with several others. The data for this dataset comes from 1,500 Traditional Chinese Medicine drug manuals, with a lower proportion of nested entities compared to the CMeEE dataset. Table VII presents the comparison results, with data for other models sourced from Hui et al.[42].

TABLE VII
THE VALUES OF THE ERTCMM DATASET FOR DIFFERENT MODELS

Model	P(%)	R(%)	F1(%)
RoBERT-CRF	65.79	73.97	69.64
RoBERT-BiLSTM-CRF	67.56	73.36	70.34
Lattice LSTM	60.52	76.15	67.44
Soft-Lexicon(LSTM)	62.27	75.52	68.26
BERT-Soft-Lexicon(LSTM)	67.01	76.66	71.51
Dual-branch TENER[42]	68.34	81.56	74.37
IBNNER	**82.17**	**84.93**	**83.53**

Our model achieved an accuracy of 82.17%, a recall of 84.93%, and an F1-score of 83.53% on the ERTCMM dataset. Fig. 14 shows a bar chart comparing the F1-scores of various models.

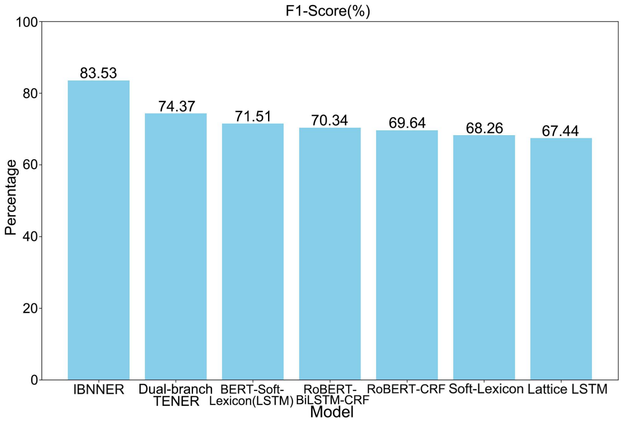


Fig. 14. Comparison of F1-score among Models on ERTCMM dataset

To demonstrate that our model also has good generalization capability in other domains, we conducted experiments on the MSRA dataset. Table VIII presents the comparison results, with data for other models sourced from Shen et al.[43].

TABLE VIII
THE VALUES OF THE MSRA DATASET FOR DIFFERENT MODELS

Model	P(%)	R(%)	F1(%)
LSTM-CRF	85.49	86.22	85.84
BiLSTM-CRF	85.40	88.76	86.93
IDCNN-CRF	83.01	91.28	86.48
BiLSTM-Gate-NER[44]	93.33	92.15	92.74
Bi-CRNN-GRU[45]	93.62	92.49	93.05
DNN-Fusion-Model[46]	94.58	94.47	94.52
IBNNER	**94.85**	**96.23**	**95.53**

Our model achieved an accuracy of 94.85%, a recall of 96.23%, and an F1-score of 95.53% on the MSRA dataset. Fig. 15 shows a bar chart comparing the F1-scores of various models.

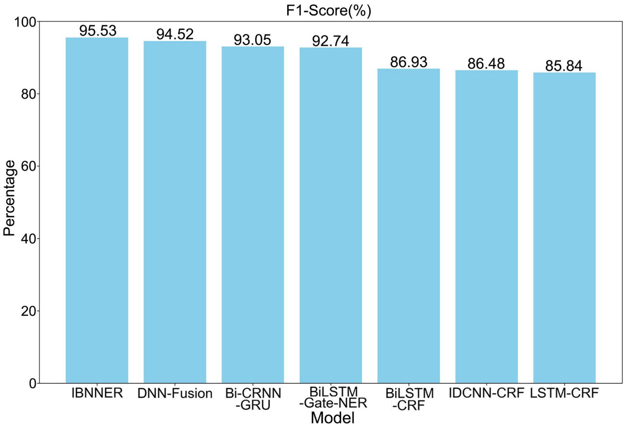


Fig. 15. Comparison of F1-score among Models on MSRA dataset

E. Ablation Experiment

To validate the effectiveness of the CNN focusing on entity fragments and the loss function, we conducted ablation experiments on the CMeEE dataset. The experimental results are presented in TABLE IX.

TABLE IX
THE VALUES OF THE CMEEE DATASET IN THE ABLATION EXPERIMENTS

Model	P(%)	R(%)	F1(%)
BERT-Biaffine	**64.17**	61.29	62.29
BERT-Biaffine-ZLPR	62.42	68.75	65.43
IBNNER-ZLPR	62.75	69.48	65.94
IBNNER	64.12	**69.12**	**66.53**

(1) The BERT-Biaffine model refers to the model that does not use CNN to interact with the score matrix and replaces the Focal loss function with a regular Softmax cross-entropy loss function.

(2) The BERT-Biaffine-ZLPR model refers to the model that does not use CNN to interact with the score matrix and replaces the Focal loss function with ZLPR loss function[47].

(3) The IBNNER-ZLPR model refers to the model that replaces the Focal loss function with ZLPR loss function.

Compared to the BERT-Biaffine model, the BERT-Biaffine-ZLPR model exhibits a 7.46% enhancement in recall and a 3.14% increase in F1-score. These improvements underscore the efficacy of the BERT-Biaffine-ZLPR model in refining the loss function, adeptly tackling sample imbalance, and bolstering overall performance. Similarly, compared to the IBNNER-ZLPR model, the IBNNER model achieved a 1.37% increase in precision and a 0.59% increase in F1-score, while the recall decreased by 0.36%. This is because the Focal Loss function focuses more on difficult-to-classify samples, making it more suitable for this task than the ZLPR loss function. Relative to the BERT-Biaffine-ZLPR model, our proposed model realizes incremental gains of 0.33% in precision, 0.73% in recall, and 0.51% in F1-score. These results indicate that the incorporation of a CNN layer, through its dynamic engagement with the score matrix, fortifies the intra-entity correlation, thereby bestowing a distinct advantage in the recognition of nested entities.

VI. CONCLUSION

This paper introduces the Improved Biaffine-based Nested Named Entity Recognition Model (IBNNER), designed to address the challenges of Nested Named Entity Recognition (Nested_NER) that conventional sequence labeling models, such as MFFNER[1] are not well-equipped to handle. The IBNNER model integrates positional information through two feed-forward neural network layers, which are adept at capturing the initial and terminal information of entity fragments. It then uses convolutional networks for dynamic interaction with the score matrix and employs the ZLPR loss function to enhance training efficacy.

Our experimental evaluation, including comparative analyses and ablation studies, demonstrates the superiority of the IBNNER model. On the CmeEE dataset, the IBNNER model achieves a 5.78% increase in precision, a 5.04% improvement in recall, and a 5.46% boost in F1-score compared to the BERT-CRF model. Compared to the BERT-Biaffine model, the IBNNER model shows a notable 7.83% increase in recall and a 4.24% improvement in F1-score. These results corroborate the model's proficiency in leveraging convolutional neural networks to accentuate entity fragments and the efficacy of the selected loss function. The IBNNER model also demonstrates strong capability in discerning both nested and non-nested entities, with performance comparable to the BERT-CRF model in non-nested scenarios.

REFERENCES

[1] S. Deng, K.-B. Lin, and P. Lu, "A Chinese named entity recognition method fusing word and radical features," in *Proceedings of the 2022 5th International Conference on Artificial Intelligence and Pattern Recognition*, 2022, pp. 502-508.

[2] J. Straková, M. Straka, and J. J. a. p. a. Hajič, "Neural architectures for nested NER through linearization," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5326–5331.

[3] T. Shibuya and E. J. T. o. t. A. f. C. L. Hovy, "Nested named entity recognition via second-best sequence learning and decoding," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 605-620, 2020.

[4] J. Yu, B. Bohnet, and M. Poesio, "Named entity recognition as dependency parsing," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, 2020, pp. 6470-6476: Association for Computational Linguistics.

[5] K. Canfield, B. Bray, and S. Huff, "Representation and database design for clinical information," in *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 1990, pp. 350-353: American Medical Informatics Association.

[6] N. Sager, M. Lyman, C. Bucknall, N. Nhan, and L. J. Tick, "Natural language processing and the representation of clinical data," *Journal of the American Medical Informatics Association,* vol. 1, no. 2, pp. 142-160, 1994.

[7] K. M. Hettne *et al.*, "A dictionary to identify small molecules and drugs in free text," *Bioinformatics*, vol. 25, no. 22, pp. 2983-2991, 2009.

[8] S. Zhao, "Named entity recognition in biomedical texts using an HMM model," in *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications*, 2004, pp. 87-90.

[9] N. Ponomareva, P. Rosso, F. Pla, and A. Molina, "Conditional random fields vs. hidden markov models in a biomedical named entity recognition task," in *The Recent Advances in Natural Language Processing*, 2007, pp. 479-483.

[10] L. Li, L. Jin, Z. Jiang, D. Song, and D. Huang, "Biomedical named entity recognition based on extended recurrent neural networks," in *The 2015 IEEE International Conference on bioinformatics and biomedicine* 2015, pp. 649-652: IEEE.

[11] Z. Xu, Y.-H. Zhu, W.-T. Liang, and F. Zhan, "A medical entity recognition model based on SoftLexicon," *Journal of Hunan University of Technology,* vol. 35, no. 05, pp. 77-84, 2021.

[12] C.-R. Pan, Q.-H. Wang, B.-Z. Tang, L. Jiang, X. Huang, and L. Wang, "Chinese electronic medical record named entity recognition based on sentence-level Lattice-long short-term memory neural network," *Academic Journal of Second Military Medical,* vol. 40, no. 5, pp. 497-506, 2019.

[13] X.-L. Li and Z.-R. Hu, "Entity recognition of Chinese electronic medical records based on pre-training model," *Computer Engineering and Design,* vol. 44, no. 02, pp. 535-540, 2023.

[14] M.-Y. Li and L. Yu, "Named entity recognition of chinese electronic medical record integrated with confrontation training," *Modern Information Technology*, vol. 7, no. 02, pp. 90-93, 2023.

[15] F.-L. Zhang, "Medical Chinese named entity recognition model combined with ERNIE2.0," *Electronic Design Engineering,* vol. 31, no. 04, pp. 38-42, 2023.

[16] ThanhThuong T. Huynh, TruongAn PhamNguyen, and Nhon V. Do, "A Keyphrase Graph-Based Method for Document Similarity Seasurement," *Engineering Letters*, vol. 30, no.2, pp692-710, 2022.

[17] Sihan Liu, Wenyu Zhang, Yujun Zhang, Zhijian Wang, and Dongxiang Gao, "A 3D Semantic Segmentation Method for Large-Scale Point Cloud on Deep Learning," *Engineering Letters*, vol. 31, no.4, pp1667-1674, 2023.

[18] Xinmiao Yu, Zhengpeng Li, Jiansheng Wu, and Mingao Liu, "Multi-module Fusion Relevance Attention Network for Multi-label Text Classification," *Engineering Letters*, vol. 30, no.4, pp1237-1245, 2022.

[19] Gang Ding, "Research on Record Named Entity Recognition of Chinese Electronic Medical based on LSTM-CRF," Lecture Notes in Engineering and Computer Science: *Proceedings of The International MultiConference of Engineers and Computer Scientists 2023*, IMECS 2023, 5-7 July, 2023, Hong Kong, pp88-93.

[20] Y.-C. Jia and D.-J. Zhu, "Medical named entity recognition based on deep learning," *Computer Systems & Applications,* vol. 31, no. 9, pp. 70-81, 2022.

[21] M. Ju, M. Miwa, and S. Ananiadou, "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*, 2018, pp. 1446-1459.

[22] M. G. Sohrab and M. Miwa, "Deep exhaustive model for nested named entity recognition," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 2843-2849.

[23] E. Zhu and J. Li, "Boundary smoothing for named entity recognition," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dublin, Ireland, 2022, pp. 7096-7108: Association for Computational Linguistics.

[24] Z. Yuan, C. Tan, S. Huang, and F. Huang, "Fusing heterogeneous factors with triaffine mechanism for nested named entity recognition," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dublin, Ireland, 2022, pp. 3174-3186: Association for Computational Linguistics.

[25] J. Su *et al.*, "Global pointer: Novel efficient span-based approach for named entity recognition," *arXiv preprint arXiv:2208.03054,* pp. 1-12, 2022.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, 2019, pp. 4171-4186: Association for Computational Linguistics.

[27] T. Dozat and C. D. Manning, "Deep biaffine attention for neural dependency parsing," presented at the *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980-2988.

[29] Z. Hongying, L. Wenxin, Z. Kunli, Y. Yajuan, C. Baobao, and S. Zhifang, "Building a pediatric medical corpus: Word segmentation and named entity annotation," in *Chinese Lexical Semantics: 21st Workshop*, 2021, pp. 652-664: Springer.

[30] L. Xu *et al.*, "CLUENER2020: Fine-grained named entity recognition dataset and benchmark for chinese," *arXiv preprint arXiv:2001.04351,* pp. 1-7, 2020.

[31] G.-A. Levow, "The third international chinese language processing bakeoff: Word segmentation and named entity recognition," In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, 2006, pp. 108-117.

[32] Y. Zhang and J. Yang, "Chinese NER Using Lattice LSTM," presented at the Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, July, 2018. Available: https://aclanthology.org/P18-1144 https://doi.org/10.18653/v1/P18-1144

[33] J. Yu, B. Bohnet, and M. Poesio, "Named entity recognition as dependency parsing," *arXiv preprint arXiv:2005.07150*, 2020.

[34] N. Yang, S. H. Pun, M. I. Vai, Y. Yang, and Q.-L. Miao, "A unified knowledge extraction method based on BERT and handshaking tagging scheme," vol. 12, no. 13, p. 6543, 2022.

[35] Y. Gu, X. Qu, Z. Wang, Y. Zheng, B. Huai, and N.-J. Yuan, "Delving deep into regularity: A simple but effective method for Chinese named entity recognition," *arXiv preprint arXiv:2204.05544*, 2022.

[36] N. Liu, Q. Hu, H. Xu, X. Xu, and M.-X. Chen, "Med-BERT: A pretraining framework for medical records named entity recognition," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5600-5608, 2021.

[37] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J.-W. Li, "A unified MRC framework for named entity recognition," *arXiv preprint arXiv:1910.11476*, 2019.

[38] Z. Huang, X. Wei, and Y. Kai, "Bidirectional LSTM-CRF models for sequence tagging," *Computer Science*, pp. 35-45, 2015.

[39] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. J. a. p. a. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[40] J. Liu, C. Xia, H. Yan, and W.-J. Xu, "Innovative deep neural network modeling for fine-grained Chinese entity recognition," *Electronics*, vol. 9, no. 6, p. 1001, 2020.

[41] L. Xu, Q.Dong, Y.Liao, C.Yu, Y.Tian, et al., "CLUENER2020: Fine-grained named entity recognition dataset and benchmark for Chinese," *arXiv preprint arXiv:2001.04351*, 2020.

[42] H. Peng, Z. Zhang, D. Liu, X. J. B. M. I. Qin, and D. Making, "Chinese medical entity recognition based on the dual-branch TENER model," *BMC Medical Informatics and Decision Making*, vol. 23, no. 1, p. 136, 2023.

[43] T.-P. Shen, Q. Meng, and Z.-H. Zhan, "Named entity recognition of Chinese text based on attention mechanism," *Journal of Network Intelligence*, vol. 8, no. 4, pp. 504-517, 2023.

[44] Y.-L. Jin, J.-F. Xie, and D.-J. Wu, "Chinese nested named entity recognition based on hierarchical tagging," *Journal of Shanghai University/Shanghai Daxue Xuebao*, vol. 28, no. 2, 2022.

[45] X.-H. Huang, L.-L. Qiao, W.-T. Yu, J. Li, and H. Xue, "Joint learning of Chinese word separation and named entity recognition," *Journal of National University of Defense Technology/Guofang Keji Daxue Xuebao*. vol. 43, no. 1, 2021.

[46] D. Zhao, J. Cao, D. Huang, J. Meng, and P. Zhang, "Dual neural network fusion model for Chinese named entity recognition," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 471-481, 2021.

[47] J. Su, M. Zhu, A. Murtadha, S. Pan, B. Wen, and Y. Liu, "ZLPR: A novel loss for multi-label classification," *arXiv preprint arXiv:2208.02955,* pp. 1-17, 2022.