Article

Not peer-reviewed version

MFGFF-BiLSTM-EGP: A Chinese Nested Named Entity Recognition Model for Chicken Disease Based on Multiple Fine-Grained Feature Fusion and Efficient Global Pointer

Xiajun Wang , Cheng Peng * , Qifeng Li * , Qinyang Yu , Liqun Lin , Pingping Li , Ronghua Gao , Wenbiao Wu , Ruixiang Jiang , Ligen Yu , Luyu Ding , Lei Zhu

Posted Date: 22 August 2024

doi: 10.20944/preprints202408.1620.v1

Keywords: nested named entity recognition; chicken disease; multiple fine-grained feature fusion; RoBERTa; efficient global pointer

Article

MFGFF-BiLSTM-EGP: A Chinese Nested Named Entity Recognition Model for Chicken Disease Based on Multiple Fine-Grained Feature Fusion and Efficient Global Pointer

Xiajun Wang [1,2], Cheng Peng [1,3,4,*], Qifeng Li [1,3,4,*], Qinyang Yu [1,3,4], Liqun Lin [2], Pingping Li [1,2], Ronghua Gao [1,3,4], Wenbiao Wu [1,3,4], Ruixiang Jiang [1,3,4], Ligen Yu [1,3,4], Luyu Ding [1,3,4] and Lei Zhu [1,3,4]

[1] lnformation Technology Research Centre, Beijing Academy of Agriculture and Forestry Sciences, Beijing 100097, China
[2] Faculty of Resources and Environmental Science Hubei University, Wuhan 430061, China
[3] National Innovation Centre of Digital Technology in Animal Husbandry, Beijing 100097, China
[4] National Engineering Research Centre for Information Technology in Agriculture, Beijing 100097, China
* Correspondence: pengc@nercita.org.cn (C.P.); liqf@nercita.org.cn (Q.L.)

Featured Application: This study proposes a multiple fine-grained nested named entity recognition model, which provides a solution for other specialized fields and lays the foundation for subsequent knowledge graph construction and intelligent inquiry system construction.

Abstract: Extracting entities from large volumes of chicken epidemic texts is crucial for knowledge sharing, integration, and application. However, Named Entity Recognition (NER) encounters significant challenges in this domain, particularly due to the prevalence of nested entities and domain-specific named entities, coupled with a scarcity of labeled data. To address these challenges, we compiled a corpus from 50 books on chicken diseases, covering 28 different disease types. Utilizing this corpus, we developed a nested NER model, MFGFF-BiLSTM-EGP. This model integrates the Multiple Fine-Grained Feature Fusion (MFGFF) module with a BiLSTM neural network and employs an Efficient Global Pointer (EGP) for predicts the entity location encoding. In MFGFF module we designed three encoders, character encoder, word encoder and sentence encoder, this design effectively captures fine-grained features and improves the recognition accuracy of nested entities. Experimental results show that the model performs robustly, with F1 scores of 91.98%, 73.32%, and 82.54% on the CDNER, CMeEE V2, and CLUENER datasets, respectively, outperforming other commonly used NER models. Specifically, on the CDNER dataset, the model achieved an F1 score of 79.68% for nested entity recognition. This research not only advances the development of knowledge graph and intelligent question-answering system for chicken diseases but also provides a viable solution for extracting disease information that can be applied to other livestock species.

Keywords: nested named entity recognition; chicken disease; multiple fine-grained feature fusion; RoBERTa; efficient global pointer

1. Introduction

The egg and broiler industries are pivotal to global food production, with China playing a crucial role as the world's largest egg producer and a major supplier of broiler meat. China accounts for 36% of global egg production and 14% of chicken meat, making these industries crucial for both food security and as a significant income source for millions of people in urban and rural areas. However,

 doi:10.20944/preprints202408.1620.v1

the rapid expansion of these industries has heightened the threat of various poultry diseases, which can severely impact production and livelihoods. As a result, effective disease prevention and control have become increasingly critical. Despite the availability of information on poultry diseases, it is often fragmented and poorly organized, making it difficult for farmers to access the necessary knowledge when needed. This paper addresses this issue by proposing a method that utilizes deep learning techniques to accurately identify and extract essential information on poultry diseases from extensive text sources. This approach lays the foundation for developing a comprehensive knowledge graph, which can support advanced applications such as intelligent Q&A systems and efficient knowledge retrieval platforms [1]. These tools will equip farmers with the information they need to protect their flocks and maintain their livelihoods.

Named Entity Recognition (NER) is a critical task in natural language processing, involving the identification of specific entities within text. Its importance has grown significantly with the rapid expansion of biomedical literature and data. In the biomedical domain, Biomedical Named Entity Recognition (BioNER) targets the recognition of entities such as disease names, symptoms, drugs, and anatomical parts [2]. Traditionally, NER models have approached this task as a sequence labelling problem, employing both rule-based and machine learning techniques [3]. Rule-based methods necessitate the manual creation of extensive rule sets, which rely heavily on domain expertise and are often restricted to narrowly defined areas [4]. In contrast, machine learning approaches, including Hidden Markov Models (HMM), Maximum Entropy Models (MEM), Support Vector Machines (SVM), and Conditional Random Fields (CRF) [5–8], rely on feature engineering. This reliance poses challenges, particularly in selecting appropriate features and capturing long-term dependencies between entities [9]. The advent of deep learning has revolutionized the field, such as Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) for NER tasks [10–12], significantly improving both efficiency and recognition accuracy. Notably, the BiLSTM-CRF model has gained widespread use [13–16], as it automates feature extraction, thereby enhancing both efficiency and accuracy without the large scale feature engineering required by earlier methods [17].

As research in NER has advanced, increasing attention has been directed toward the recognition of nested entities. The concept of "nested region names" was first introduced in the task definition of the Automatic Content Extraction (ACE) research [18], Represented as entities that exist within other entities [19,20]. For instance, in the study of chicken diseases, a symptom such as ‘spleen swelling’ includes the nested entities ’spleen’ representing a body part, and ’swelling’ representing a symptom. Recognizing these nested entities, whether they are heterogeneous (different types) or homogeneous (overlapping types), is complex and presents a significant challenge for accurate identification [21]. The introduction of the concept of "nested region names" in ACE research underscores the importance of recognizing these intricate structures within named entities.

To address the challenges posed by nested named entities, researchers have proposed various methods, including hypergraph-based, region-based, transition-based, and span-based techniques. For example, Huang et al. [22] introduced a Hyper Graph Network (HGN) structure to manage nested entities by representing each sentence as a hypergraph, with words as nodes and entities as hyper-edges, thereby transforming the recognition task into hyper-edges classification. Region-based methods treat nested NER as a multi-class classification problem by first representing potential regions (subsequences) and then classifying them. Jiang et al. [23] proposed a candidate region-aware model that utilizes binary sequence labeling followed by candidate region classification, demonstrating significant performance on public datasets. Transition-based methods, inspired by dependency parsers, incrementally construct trees through greedy decoding. Wang et al. [24] developed a neural transition model using Stack LSTM, which effectively captures character-level representations and efficiently represents the system's state. Span-based methods enumerate all possible text segments (spans) and then determine their entity status [25], which naturally suits the nested entity recognition task. Li et al. [26] enhanced this approach by developing a segment-enhanced span-based model (SESNER), which improves model performance while accurately handling complex nested entities. Additionally, the Global Pointer (GP) model, which leverages

 doi:10.20944/preprints202408.1620.v1

relative positions through a multiplicative attention mechanism, offers a global perspective on start and end positions for predicting entities and has demonstrated outstanding performance across various nested NER tasks [27–30].

Despite these advances, current NER models still rely heavily on large training datasets, making pre-trained models crucial for embedding layers in NER tasks [31,32]. Models such as BERT, MCBERT, RoBERTa, XLNet, and ERNIE, trained on extensive corpora, have significantly improved entity recognition accuracy [33–36]. However, challenges remain in accurately recognizing rare entities and domain-specific terms. To address these issues, researchers have sought to enhance pre-trained models with lexicon information. Techniques like Softword, ExSoftword, and SoftLexicon incorporate lexicons to improve the recognition of domain-specific named entities [37]. For instance, Zhao et al. [38]introduced the BERT-IDCNN-CRF model, which integrates the SoftLexicon method, demonstrating impressive efficiency across multiple datasets. Similarly, Zhang et al. [39]improved recognition by incorporating lexicons and similar words into character representations. Liu et al. [40]further enhanced model capabilities by dynamically updating custom lexicon segmentation methods, thereby improving the identification of domain-specific terms and new entities. Additionally, incorporating syntactic information has been shown to significantly enhance NER performance [41]. For example, Tian et al. [42]developed the BioKMNER model, which employs a Key-Value Memory Network (KVMN) to integrate syntactic information, achieving excellent results on biomedical datasets. Luoma et al. [43] demonstrated that adding context through additional sentences in BERT input systematically improves NER performance. These methods, whether introducing lexicon information, syntactic data, sentence context, or even stroke information of Chinese characters [44], are aimed at improving NER model performance in specialized domains.

In summary, NER continues to face significant challenges in the field of biological disease research, particularly in accurately recognizing nested entities and domain-specific terms. These challenges are especially pronounced in the study of chicken diseases, where the acquisition of relevant corpora is difficult and the labour cost of data annotation is high. To address these issues, we collected and organized a chicken disease corpus comprising 20 million characters, trained a specialized word vector tailored to chicken disease terminology, and annotated a portion of this data with high precision. Building on this foundation, we developed a nested NER model, MFGFF-BiLSTM-EGP, which leverages Multiple Fine-Grained Feature Fusion (MFGFF) and the Efficient Global Pointer (EGP). The primary contributions of this paper are as follows:

1. We have constructed the MFGFF-BiLSTM-EGP model, which connects the fusion output of multi-fine-grained features to the BiLSTM neural network layer, and finally through a fully connected layer into the EGP to predicts the entity position.
2. In the MFGFF module we designed, the character encoder obtains character features by fine-tuning the RoBERTa pre-trained model, the word encoder acquires word features through word-character matching, word frequency weighting, and multi-head attention mechanism, and the sentence features are output using SBERT. MFGFF effectively integrates multiple fine-grained features. In addition, the introduction of EGP enables the prediction of nested entities by means of positional coding.
3. We have constructed a comprehensive knowledge base for chicken diseases, which includes a 20-million-character corpus, a vocabulary containing 6760 specialized terms, a 200-dimensional word vector in the field of chicken diseases, and a high-quality annotated dataset CDNER annotated under the guidance of veterinarians.

2. Materials and Methods

2.1. Data and Lexicon

To address the lack of available datasets on chicken epidemic diseases, we systematically compiled a 20 million characters comprehensive corpus from various sources, including 50 professional books, medical records, technical reports, academic literature, and reputable online websites. We first utilized automated Python scripts to eliminate duplicates, redundancies, and any obviously incorrect information. Following this initial automated screening, we conducted a manual

 doi:10.20944/preprints202408.1620.v1

review to ensure the authenticity and accuracy of the data. Additionally, with guidance from veterinary experts, we compiled a specialized lexicon comprising 6,760 terms specific to chicken diseases. we integrated this lexicon into the Jieba segmentation tool to facilitate the training of word vectors. Through these steps, we successfully developed a high-quality knowledge base on chicken diseases, providing a robust foundation for further research and practical applications.

2.2. Entity Labelling

Base on chicken disease corpus, we have selected 5000 high-quality data and manually annotated them using the Label-Studio platform with the knowledge of veterinary experts. We established five distinct labels: type, disease, symptom, body part, and drug. Table 1 below provides examples and the frequency of each label.

Table 1. Examples of labels for five entity types.

Entity	Labels	Example	Num
type	TY	adult chickens, laying hens, flocks, sick chickens	4503
disease	DI	Newcastle disease, avian influenza	4098
symptom	SY	clinical warming, oedema, congestion	9265
bodypart	BO	head, eyes, liver, lymphocyte	5744
drug	DR	oxytetracycline, gentamycin, tetracycline	3201
Total			26811

Given the prevalence of nested entities in the task of labelling named entities related to chicken diseases, we developed a unified labelling standard to ensure accurate and comprehensive annotation. For example, In Figure 1, a body part entity "脾脏 (spleen)" and a symptom entity "肿大 (swelling)" is nested within symptom entity "脾脏肿大 (spleen swelling)", illustrating a nesting of different and identical entity types. Under the guidance of veterinary experts, we meticulously distinguished and labelled entities at different levels, ensuring that all named entities were identified and processed independently.

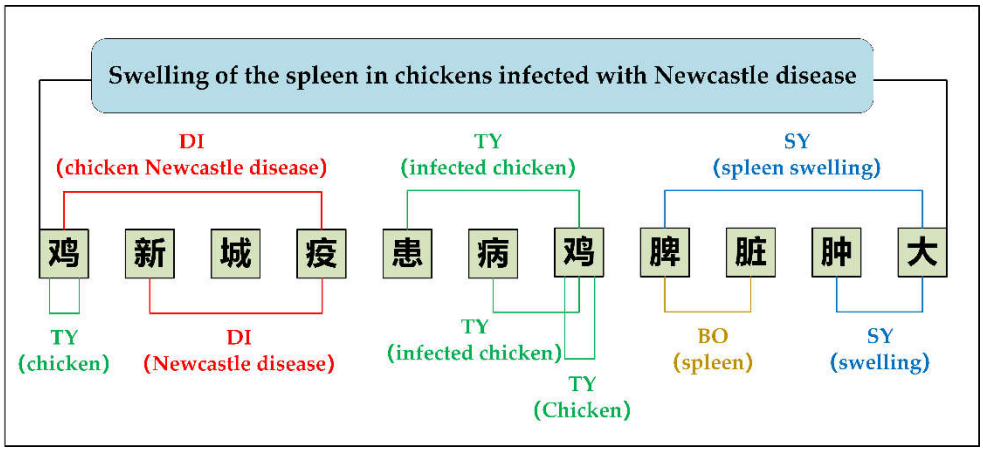

Figure 1. An example of nested entity annotation, with different colored lines representing different entities.

2.3. MFGFF-BiLSTM-EGP

The MFGFF-BiLSTM-EGP model developed in this study,the algorithmic pseudo-code for the model is in Algorithm A1 in the Appendix A. As illustrated in Figure 2, integrates three key modules within its embedding layer: character encoding, word encoding, and sentence encoding.

For character encoding, we fine-tuned the RoBERTa pre-trained model, using its last hidden state output to generate character vectors. The word encoding module utilizes word vectors trained with GloVe, which are aligned with the output from RoBERTa. When a word vector contains the

 doi:10.20944/preprints202408.1620.v1

current input character, a multi-head attention mechanism merges the corresponding word vector to form the final word vector. For sentence encoding, each sentence is processed through the SBERT (Sentence BERT) model, generating a corresponding sentence vector. These three vectors: character, word, and sentence, are concatenated to construct the model's embedding layer. A BiLSTM network is then employed to capture the contextual information within the text. Finally, the EGP component predicts the positions of named entities within the text.

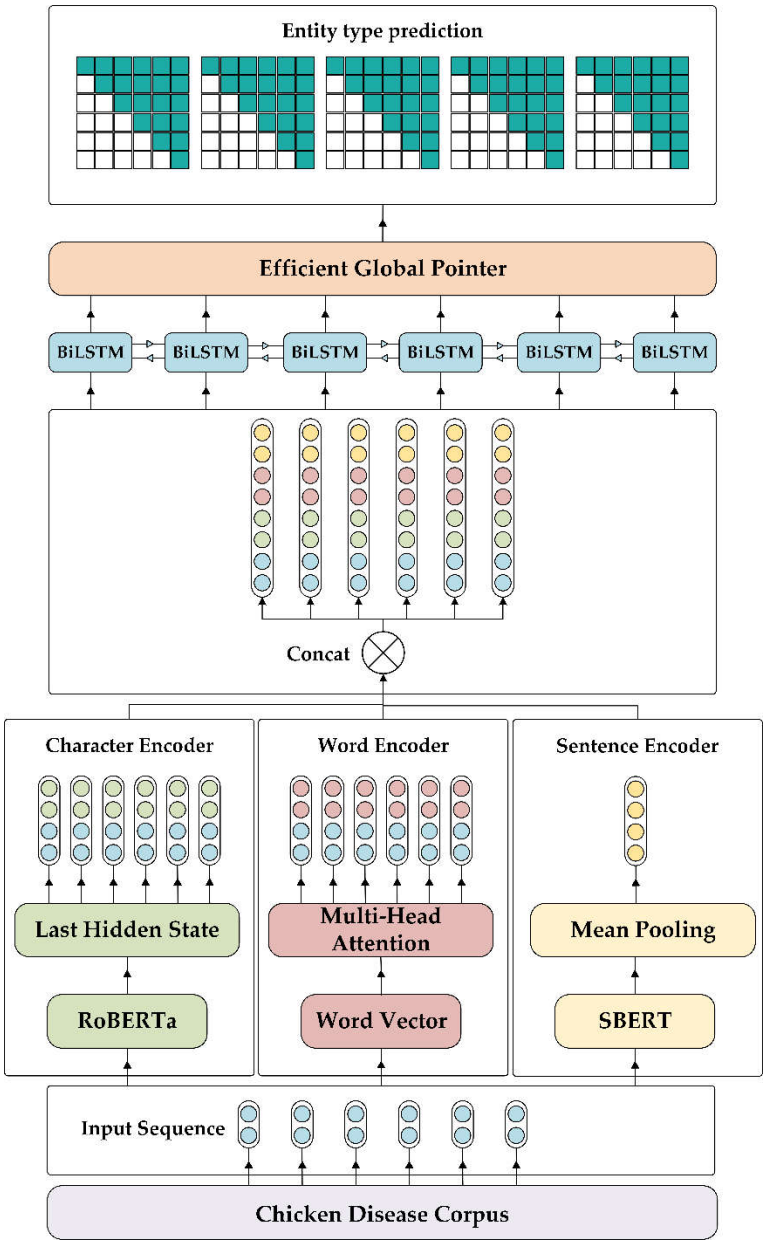

Figure 2. MFGFF-BiLSTM-EGP model framework.

2.3.1. Character Encoder

As shown in Figure 3, in the character encoding module, we fine tune Roberta and use Roberta's last hidden state as the character vector. The pre-trained RoBERTa model serves as our character encoder, building upon the BERT architecture. BERT is a language model based on the Transformer framework, which leverages an attention mechanism to dynamically focus on different parts of the input data, enabling effective sequence processing [45]. RoBERTa enhances BERT's capabilities, particularly for tasks like NER, by introducing several key modifications [46]. As an embedding layer, RoBERTa transforms input character sequences into high-dimensional feature vectors that capture

 doi:10.20944/preprints202408.1620.v1

not only individual word characteristics but also rich contextual information. This context-awareness is crucial for accurately identifying and classifying named entities, as it allows for a nuanced understanding of the text surrounding each word. RoBERTa's capacity to capture these contextual nuances makes it especially effective for NER tasks.

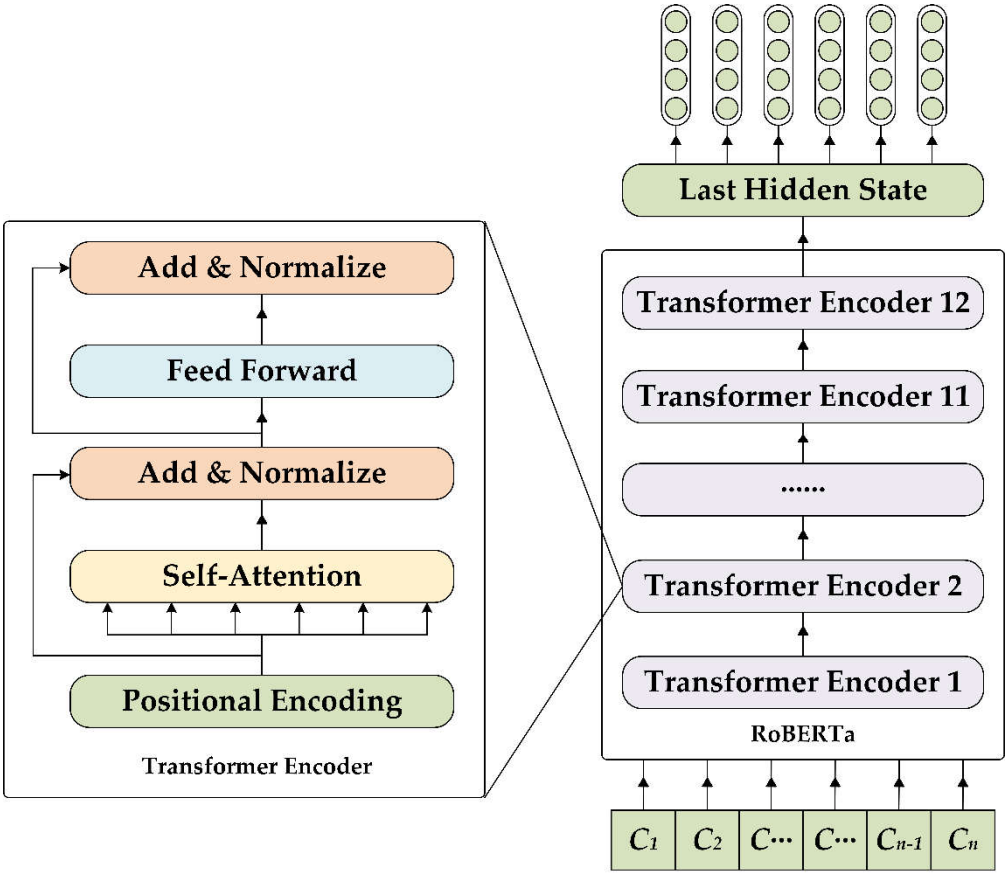

Figure 3. Character encoder framework.

2.3.2. Word Encoder

We developed a 200-dimensional word vector for chicken diseases using the GloVe model This process began with the construction of a co-occurrence matrix X, where X_{ij} represents the frequency with which word i co-occurs with word j within a predefined context window. The probability of co-occurrence between words i and j is then calculated using the formula:

$$P_{ij} = \frac{X_{ij}}{\sum_k X_{ik}} \tag{1}$$

Here, P_{ij} indicates the likelihood of word j appearing within the context of word i.

The GloVe model refines word vectors to accurately reflect these co-occurrence probabilities, aiming to minimize the following objective function:

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log\left(X_{ij}\right)\right)^2 \tag{2}$$

In this equation, w_i^T and w_j are the word vectors for words i and j, respectively, while b_i and b_j are the corresponding bias terms.

The logarithm of the co-occurrence count between the words, $\log\left(X_{ij}\right)$, and the weighting function $f\left(X_{ij}\right)$, are used to reduce the influence of rare co-occurrences, defined as:

 doi:10.20944/preprints202408.1620.v1

$$f(X_{ij}) = \begin{cases} \left(\frac{X_{ij}}{X_{max}}\right)^{\alpha} & \text{if } X_{ij} < X_{max} \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

In this context, X_{max} and α are hyperparameters that shape the weighting function, ensuring that both frequent and rare co-occurrences are appropriately weighted.

As shown in Figure 4, For each sentence $X = \{c_1, c_2, \ldots, c_n\}$ with n tokens, each token c_i is represented by combining character with GloVe word vectors. The set $W_c = \{w_i \mid c_i \in w_i\}$ represents the collection of word vectors w_i for words containing the character c_i. To balance the influence of word vectors according to their frequencies, logarithmic scaling is applied to the word frequencies:

$$f_j' = \log(1 + f_j) \quad (4)$$

where f_j' is the logarithmically scaled frequency of word j, and f_j is the original frequency. The adjusted word vector for word j is then calculated as:

$$w_j' = f_j' w_j \quad (5)$$

Here, w_j' is the weighted word vector, and w_j is the original word vector.

The integration of character vectors with weighted word vectors is achieved through a multi-head attention mechanism. The query, key, and value vectors are first calculated as follows:

$$Q = W_Q h_i \quad (6)$$

$$K_j = W_K w_j' \quad (7)$$

$$V_j = W_V w_j' \quad (8)$$

where Q represents the query vector, K_j is the key vector, and V_j is the value vector. The matrices W_Q, W_K, and W_V are the projection matrices for the query, key, and value, respectively.

The attention mechanism computes the attention weights to evaluate the relevance of each key-value pair to the query.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (9)$$

where d_k is the dimension of the key vector, and the softmax function normalizes the similarity scores, converting them into a probability distribution.

A weighted summation of the value $\text{Attention}(QW_{Qi}, K_j W_{Ki}, V_j W_{Vi})$ vectors using the computed attentional weights yields an output vector for each header:

$$\text{head}_i = \sum_{j=1}^{n} \text{Attention}(QW_{Qi}, K_j W_{Ki}, V_j W_{Vi}) \quad (10)$$

where n is the length of the input sequence, W_{Qi}, W_{Ki}, W_{Vi} are the projection matrices for the i-th attention head. $\text{Attention}(QW_{Qi}, K_j W_{Ki}, V_j W_{Vi})$ is the attention weight calculated using the query matrix QW_{Qi} and the key matrix $K_j W_{Ki}$. The corresponding value matrix $V_j W_{Vi}$ is weighted and summed up.

These probabilities are used to weigh the value vectors, determining their contribution to the final output. The multi-head attention mechanism allows the model to focus on different parts of the input sequence simultaneously. By employing multiple attention heads, the model captures a wider range of relationships within the data. The outputs from each attention head are concatenated and undergo a linear transformation:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W_O \quad (11)$$

where, $\text{MultiHead}(Q, K, V)$ denotes the final weighted output of the word vector through the multi-head attention mechanism, the concatenated outputs are transformed using the projection matrix W_O to produce the final output.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 August 2024 doi:10.20944/preprints202408.1620.v1

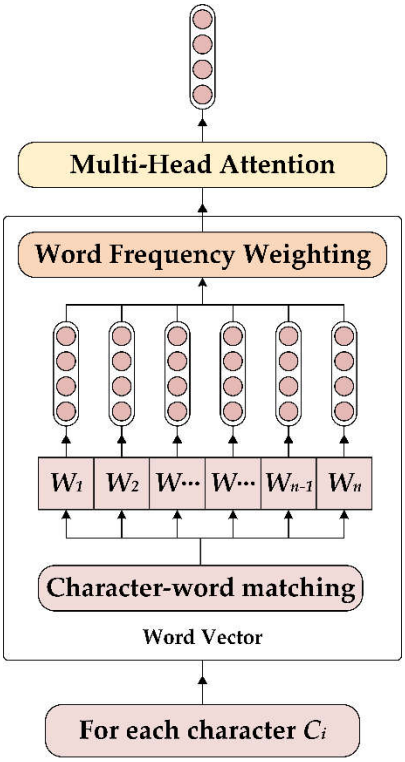

Figure 4. Word encoder framework.

2.3.3. Sentence Encoder

SBERT (Sentence-BERT) enhances BERT by utilizing Siamese and Triplet network architectures to generate fixed-size sentence embeddings. It employs a BERT model with shared weights to process pairs of sentences, creating sentence vectors through pooling operations.

As shown in Figure 5, Initially each input sentence is processed by BERT, producing token representations denoted as $\{h_1, h_2, \ldots, h_n\}$, where h_i is the vector representation of the i-th token. SBERT then applies a mean pooling operation to these token vectors to derive a fixed-length sentence vector:

$$u = \frac{1}{n}\sum_{i=1}^{n} h_i \tag{12}$$

Here, h_i is the vector of the i-th token, and n is the total number of tokens in the sentence.

SBERT uses a Siamese network structure to process sentence pairs, As shown in Figure 5, where each sentence is passed through a BERT model with shared weights to produce its corresponding sentence vector. Given a pair of sentences s_1 and s_2,their respective sentence vectors u_1 and u_2 are generated as follows:

$$u_1 = \text{Pooling}\big(\text{BERT}(s_1)\big) \tag{13}$$

$$u_2 = \text{Pooling}\big(\text{BERT}(s_2)\big) \tag{14}$$

This structure allows SBERT to efficiently compare and measure the similarity between sentence pairs by producing consistent and comparable sentence embeddings.

 doi:10.20944/preprints202408.1620.v1

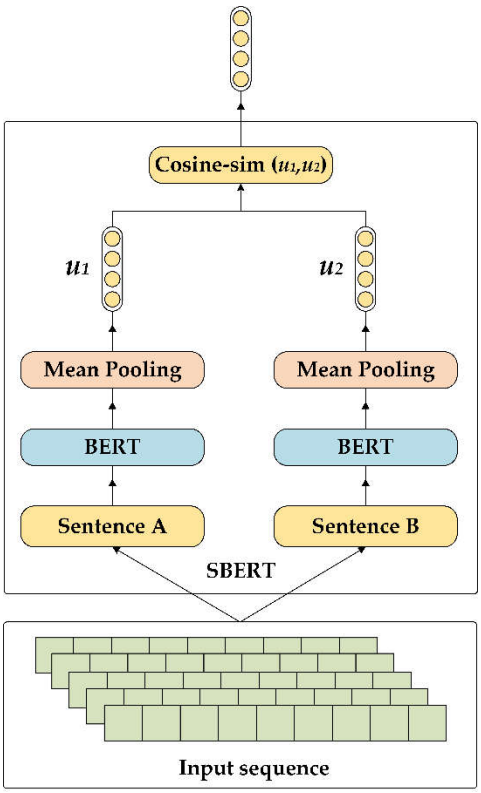

Figure 5. Sentence encoder framework.

2.3.4. BiLSTM

LSTM networks are a type of recurrent neural network (RNN) designed to overcome issues like gradient vanishing in traditional RNNs when dealing with long sequences. They are effective for time series data and capturing long-term dependencies. However, standard LSTMs only process information in one direction, which limits their ability to fully capture context in text sequences. Bidirectional LSTM (BiLSTM) solves this by processing the sequence in both forward and backward directions. It uses two LSTM networks, one for the original sequence and one for the reversed sequence. The combined output captures information from both past and future contexts, making BiLSTM particularly useful for tasks like NER, where understanding context from both directions is important. In this article, BiLSTM is used to optimize the output of the embedding layer.

2.3.5. Efficient Global Pointer

The Global Pointer (GP) method enhances NER by employing a span classification approach, which treats the head and tail of an entity as a unified pair. This approach provides a more holistic view of the entity, ensuring that training, prediction, and evaluation processes are consistently conducted at the entity level. This global perspective allows GP to effectively recognize both nested and non-nested entities. The Efficient Global Pointer (EGP) builds upon the GP approach by addressing the issue of parameter explosion, a common challenge in complex models. EGP decomposes the NER task into two subtasks: extraction and classification. This approach enables parameter sharing during extraction, which significantly reduces the number of parameters needed for classification. Consequently, EGP is more scalable and efficient compared to its predecessor.

Furthermore, EGP introduces a specialized loss function, inspired by circle loss, to manage class imbalance. This loss function ensures that the scores of the target classes consistently exceed those of non-target classes, enhancing the model's ability to distinguish between different entities. These improvements make EGP a robust and scalable solution for NER tasks, capable of delivering superior performance across various datasets and entity types.

 doi:10.20944/preprints202408.1620.v1

Figure 6 illustrates the entity prediction process using EGP for the sentence "Swelling of the spleen in chickens infected with Newcastle disease" EGP accurately labels the entity by employing positional coding, where the end position of the entity is marked as 1. This approach allows the model to effectively identify nested named entities.

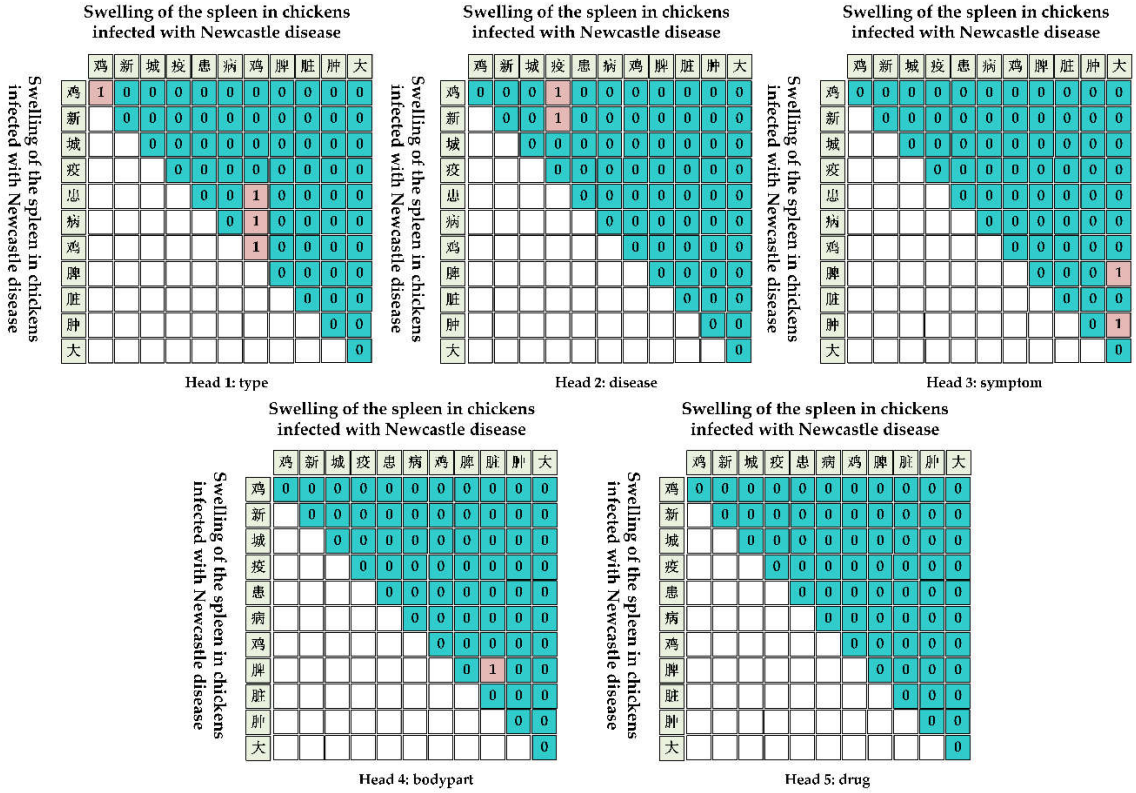

Figure 6. An example of EGP prediction for nested entities, where the end position of the entity part of the label is coded 1 and the non-entity part is coded 0.

After obtaining the integrated token representations H, the next step is to predict spans for entity recognition, which involves identifying the start and end positions of potential entity spans within a sentence. For each token x_i, the span prediction process utilizes feedforward layers to compute the start and end representations as follows:

$$q_{i,\alpha} = W_{q,\alpha} H_i + b_{q,\alpha} \tag{15}$$

$$k_{i,\alpha} = W_{k,\alpha} H_i + b_{k,\alpha} \tag{16}$$

Here, $q_{i,\alpha}$ and $k_{i,\alpha}$ represent the vector representations for the start and end tokens of the span for entity type α. The span score $s_\alpha(i,j)$ for entity type α is then calculated by:

$$s_\alpha(i,j) = q_{i,\alpha}^T k_{j,\alpha} \tag{17}$$

To incorporate positional information, relative position encoding (RoPE) is applied modifying the span score calculation as follows:

$$s_\alpha(i,j) = \left(R_i q_{i,\alpha}\right)^T \left(R_j k_{j,\alpha}\right) = q_{i,\alpha}^T R_{j-i} k_{j,\alpha} \tag{18}$$

In this equation, R_i and R_j are the position encodings for the start and end tokens, respectively.

The EGP approach splits the NER task into extraction and classification, sharing parameters across entity types to reduce the total parameter count. Extraction finds entity spans, and classification assigns the correct type. The scoring function that integrates both tasks is formulated as follows:

 doi:10.20944/preprints202408.1620.v1

$$s_\alpha(i,j) = \left(W_q H_i\right)^T \left(W_k H_j\right) + w_\alpha^T \left[H_i; H_j\right] \tag{19}$$

Here, W_q and W_k are the projection matrices used for span extraction, and w_α^T is the parameter vector associated with entity classification.

To further optimize the use of parameters, we employ concatenated span representations $(q_i; k_i)$ instead of the full token representation H_i. The scoring function is then reformulated as:

$$s_\alpha(i,j) = q_i^T k_j + w_\alpha^T \left[q_i; k_i; q_j; k_j\right] \tag{20}$$

In this equation, w_α^T is a parameter vector specific to the entity type α.

To improve NER performance with imbalanced datasets, EGP introduce a specialized loss function inspired by circle loss. It ensures target class scores are higher than non-target ones. The loss function is defined as follows:

$$\mathcal{L} = \log\left(1 + \sum_{i \in \Omega_{\text{neg}}} e^{s^i} \sum_{j \in \Omega_{\text{pos}}} e^{-s^j}\right) \tag{21}$$

Here, Ω_{pos} and Ω_{neg} represent the sets of positive and negative samples, respectively.

This formulation is extended to handle the imbalance in entity types using the following loss function for a specific entity type α:

$$\mathcal{L}_\alpha = \log\left(1 + \sum_{(q,k) \in P_\alpha} e^{-s_\alpha(q,k)}\right) + \log\left(1 + \sum_{(q,k) \in Q_\alpha} e^{s_\alpha(q,k)}\right) \tag{22}$$

In this equation, E_α denotes the set of entity spans for entity type α, while P_α and Q_α represent the sets of non-entity spans and negative spans for entity type α, respectively. The term $s_{q,k}^\alpha$ is the span score for entity type α between tokens q and k.

The loss function $\mathcal{L}_\alpha$ effectively balances the classification of both entity and non-entity spans, ensuring that the model accurately distinguishes between them, thereby addressing class imbalance and improving overall NER performance.

2.4. Experimental Environment and Hyperparameter

The experiments were conducted using a Tesla P100 GPU and PyTorch 2 as the deep learning framework. Table 2 outlines the specific experimental parameters.

Table 2. Experimental parameters.

Hyperparameter	Value
Optimizer	Adam
Warm up	0.1
LSTM units	256
Batch size	32
Dropout	0.4
Max len	128
Epoch	20

2.5. Evaluation Indicators

The study uses four key metrics: Precision (P), Recall (R), and F1-Score (F1). Precision shows how accurately the model identifies true positives among the predicted positives, with higher values indicating better accuracy. Recall measures the model's ability to identify all relevant positives, with higher values showing greater sensitivity. The F1-Score, the harmonic mean of Precision and Recall, offers a balanced view of the model's performance, with a higher score indicating a stronger, more reliable model.

 doi:10.20944/preprints202408.1620.v1

$$P = \frac{TP}{TP + FP} \tag{23}$$

$$R = \frac{TP}{TP + FN} \tag{24}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{25}$$

Table 3. Positive and Negative Samples.

Sample	Prediction	
	Positive example	Negative example
Positive example	TP	FN
Negative example	FP	TN

3. Results

We conducted experiments on three datasets, each divided into training, validation, and testing sets with a 6:2:2 ratio:

- CDNER (ours): Chicken Disease Named Entity Recognition dataset comprising 5,000 high-quality samples labeled by veterinary experts, containing five types of entities with nested structures.
- CMeEE V2 [47]: A Chinese medicine entity recognition dataset with 20,000 annotated samples across nine categories, including nested entities. The categories include diseases (dis), clinical symptoms (sym), drugs (dru), medical equipment (equ), medical procedures (pro), body (bod), medical examination items (ite), microbiology (mic), and departments (dep).
- CLUENER [48]: A general domain NER dataset with 12,091 annotated samples across ten categories: addresses, books, companies, games, government, movies, names, organizations, positions, and scenes.

For the CDNER dataset, we employed chicken disease word vectors trained on a 20 million characters corpus. For the CMeEE V2 and CLUENER datasets, we utilized 200-dimensional Chinese word vectors from Tencent AI Lab [49].

3.1. Main Results Compared with Other Models

We evaluated the performance of several mainstream NER models on these datasets:

- Lattice LSTM [50]: Encodes input characters and all potential words in the matching dictionary.
- FLAT [51]: A Transformer-based model that utilizes a unique position encoding to integrate the Lattice structure, seamlessly introducing lexical information.
- BERT-CRF: Combines a traditional CRF decoder with the BERT pre-trained model.
- BERT-MRC [52]: Reframes the NER task as a reading comprehension problem.
- SLRBC [53]: Integrates lexical boundary information using the Softlexicon method with RoBERTa, BiLSTM, and CRF.

As summarized in Table 4, the MFGFF-BiLSTM-EGP model consistently outperforms other models across all three datasets, achieving the highest F1 scores: 91.98% on CDNER, 73.32% on CMeEE V2, and 82.54% on CLUENER. The superior performance of the MFGFF-BiLSTM-EGP model can be attributed to its integration of character, word, and sentence vectors, fused with BiLSTM and EPG, which enhances recognition accuracy by incorporating specialized vocabulary and contextual information.

 doi:10.20944/preprints202408.1620.v1

Table 4. Comparison of the results of 5 commonly used obvious and our model on three datasets.

Model	CDNER			CMeEE V2			CLUENER		
	P/%	R/%	F1/%	P/%	R/%	F1/%	P/%	R/%	F1/%
Lattice LSTM	82.43	84.51	83.46	61.26	62.33	61.79	71.69	70.78	71.23
FLAT	80.55	82.99	81.75	58.18	61.11	59.61	64.64	68.45	66.49
BERT-CRF	87.86	89.53	88.69	70.93	73.24	71.98	79.15	81.86	80.48
BERT-MRC	80.16	85.91	82.93	68.00	68.35	67.97	75.60	78.22	76.89
SLRBC	88.58	90.72	89.64	71.54	72.59	72.06	81.55	80.51	81.03
MFGFF-BiLSTM-EGP (ours)	91.42	92.18	**91.98**	73.12	73.68	**73.32**	85.21	80.38	**82.54**

The SLRBC model also exhibits strong performance, particularly on the CDNER dataset, with an F1 score of 89.64%. This success is largely due to the Softlexicon method, which enhances vocabulary representation, combined with RoBERTa's robust contextual embeddings, BiLSTM's capacity to capture sequence data, and CRF's sequence tagging capabilities. Although BERT-CRF lacks the Softlexicon and BiLSTM modules present in the SLRBC model, it still performs competitively due to BERT's powerful representation capabilities, albeit with slightly lower metrics across all datasets. BERT-MRC, which reinterprets NER as a reading comprehension task, delivers adequate but not outstanding results, with F1 scores of 82.93% on CDNER, 67.97% on CMeEE V2, and 76.89% on CLUENER. Its performance could potentially be improved by refining the description of entity types during MRC parameter settings.

3.2. Entity Level Evaluation

Figure 7 and Table 5 presents the entity-level evaluation results of the MFGFF-BiLSTM-EGP model across the CDNER, CMeEE, and CLUENER datasets, detailing Precision, Recall, and F1 Scores. In the CDNER dataset, the model demonstrates exceptional performance in recognizing the "symptom," "bodypart," "disease," "drug," and "type" categories, achieving F1 scores of 87.6%, 92.43%, 92.98%, 93.22%, and 93.68%, respectively. This consistency underscores the high quality of our dataset, indicating that the annotations are both balanced and precise, thereby facilitating robust model training. These results validate the model's superior performance in entity recognition within this dataset.

For the CMeEE V2 dataset, the model's performance varies across different categories. It excels in the "dis" (disease) and "dru" (drug) categories, with F1 scores of 83.81% and 81.1%, respectively. However, the model encounters difficulties in the "equ" (equipment) and "ite" (medical examination items) categories, where F1 scores drop to 62.62% and 62.04%, respectively. Notably, in the "equ" category, the model's Precision is only 55.23%, likely due to the limited representation of specialized vocabulary within the general domain word vectors, resulting in weaker recognition performance in these areas.

In the CLUENER dataset, the model shows a relatively balanced performance across categories, achieving high F1 scores in the "company," "government," and "position" categories—86.05%, 84.28%, and 84.33%, respectively. This indicates strong recognition capabilities with minimal disparity between Precision and Recall, reflecting good stability. However, the model's performance in the "movie" and "book" categories is less satisfactory, with F1 scores of 75.76% and 80.2%, respectively. Overall, the model maintains balanced performance across multiple categories.

 doi:10.20944/preprints202408.1620.v1

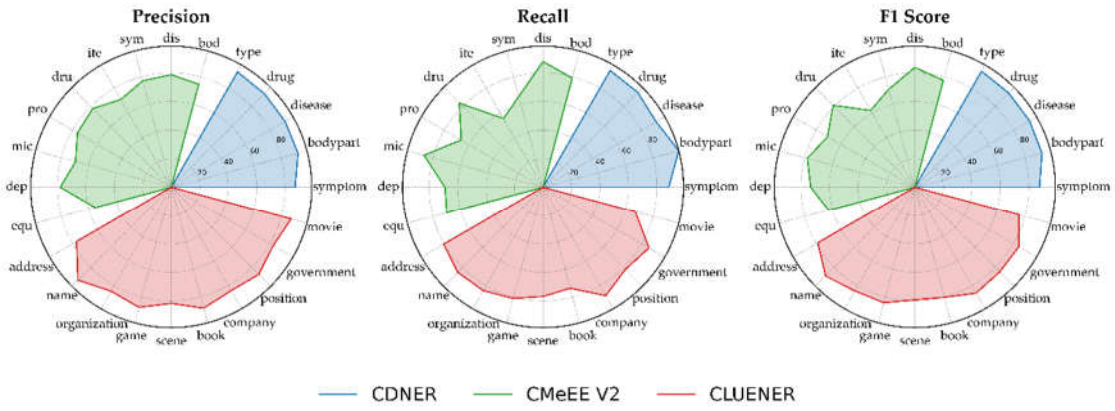

Figure 7. Radar plots of the entity-level assessment results of the MFGFF-BiLSTM-EGP model on three datasets, including precision, recall and F1.

Table 5. Entity Level Evaluation.

Data	Category	P %	R %	F1 %	Macro P %	Macro R %	Macro F1 %
CDNER	symptom	86.99	88.26	87.6	91.42	92.18	91.98
	bodypart	92.19	92.70	92.43			
	disease	92.29	91.70	92.98			
	drug	92.50	93.97	93.22			
	type	93.11	94.25	93.68			
CMeEE V2	bod	74.56	79.05	77.30	73.12	73.68	73.32
	dis	78.73	87.85	83.81			
	sym	77.10	65.75	70.9			
	ite	70.87	55.58	62.04			
	dru	77.94	83.23	81.10			
	pro	76.09	66.30	70.84			
	mic	69.56	86.60	78.19			
	dep	77.97	68.78	73.1			
	equ	55.23	70.01	62.62			
CLUENER	address	76.81	80.71	78.71	85.21	80.38	82.54
	name	92.67	84.76	88.54			
	organization	84.52	84.02	84.27			
	game	87.42	80.98	84.08			
	scene	81.75	76.87	79.23			
	book	88.04	73.64	80.20			
	company	84.09	88.10	86.05			
	position	86.92	81.88	84.33			
	government	82.72	85.90	84.28			
	movie	87.21	66.96	75.76			

3.3. Ablation Study

Table 6 shows the results of ablation experiments performed by the MFGFF-BiLSTM-EGP model on the CDNER dataset to evaluate the impact of different module combinations on the model's F1 score. The study investigates the contributions of the pre-trained model, word encoder, and sentence encoder. The results indicate that each module significantly enhances the model's performance, with the highest F1 score of 91.98% achieved when all three modules are integrated.

 doi:10.20944/preprints202408.1620.v1

Table 6. Ablation Study.

Model	Pre-trained Model	Word Encoder	Sentence Encoder	F1/%
1	✓			88.01
2		✓		82.32
3		✓	✓	82.67
4	✓	✓		91.33
5	✓		✓	88.34
6	✓	✓	✓	**91.98**

3.3.1. Effect of Pre-Trained Model

The pre-trained model substantially improved the overall performance. When only the pre-trained model (Model 1) was employed, the F1 score reached 88.01%, which is 5.69% higher than that of the model without the pre-trained model (82.32% for Model 2). This finding underscores the pre-trained model's effectiveness in capturing underlying features. The combination of the pre-trained model with the word encoder (Model 4) further increased the F1 score to 91.33%, representing a 9.01% improvement over Model 2 (which used only the word encoder). This emphasizes the significance of pre-trained models in complex feature representation. When the pre-trained model was combined with both the word and sentence encoders (Model 6), the F1 score peaked at 91.98%, showing a 9.31% improvement over Model 3, which combined the word and sentence encoders. This result further demonstrates the pre-trained model's capacity to maximize overall model performance.

3.3.2. Effect of Word Encoder

The word encoder plays a crucial role in enhancing the model's performance, especially when integrated with other modules. The F1 score for the word encoder alone (Model 2) is 82.32%, which, although lower than that of Model 1, which utilized only the pre-trained model, still illustrates the word encoder's value in word-level feature extraction. When the word encoder is combined with the pre-trained model (Model 4), the F1 score rises to 91.33%, a 3.32% improvement over Model 1. This finding indicates that the word encoder significantly contributes to refining the pre-trained model's fine-grained feature representation. In Model 6, where the word encoder is combined with both the pre-trained model and the sentence encoder, the F1 score further improves to 91.98%, a 3.64% increase over Model 5, which combined the pre-trained model and the sentence encoder. This demonstrates the pivotal role of the word encoder in a multi-module combination.

3.3.3. Effect of Sentence Encoder

The sentence encoder's impact on model performance is more nuanced and depends on its combination with other modules. When combined with the word encoder (Model 3), the F1 score reached 82.67%, a modest increase of 0.35% compared to Model 2, which used only the word encoder. This slight improvement may be due to the introduction of the sentence encoder, which could add redundant information in the absence of a pre-trained model. When the sentence encoder was combined with the pre-trained model (Model 5), the F1 score increased slightly to 88.34%, just 0.33% higher than Model 1 (88.01%), which used only the pre-trained model. In Model 6, the F1 score achieved its maximum of 91.98% when all three modules were used together, an improvement of 0.65% compared to Model 4. These results suggest that the sentence encoder, when used alongside the pre-trained model and word encoder, can marginally enhance the global semantic representation of the model.

4. Discussion

4.1. Visualization of Token Representations in Feature Space

 doi:10.20944/preprints202408.1620.v1

We conducted feature visualization and analysis across three datasets: CDNER, CMeEE V2, and CLUENER. By labeling 50 entities per category and extracting their features, we applied t-SNE for dimensionality reduction to facilitate visualization. Our analysis focused on two approaches: word vectors and MFGFF.

Figure 8 illustrates that the effectiveness of MFGFF varies significantly across different datasets. In the CDNER dataset, the original word vector representations exhibited minimal distinction between feature vectors across entity categories, leading to substantial overlap and dispersion. This lack of clear differentiation was evident. However, following MFGFF, the feature representations improved markedly, with data points clustering more centrally within their respective categories. This resulted in more distinct category clusters and enhanced inter-category distinguishability. Despite these improvements, challenges remain, such as the observed similarity between categories like 'body part' and 'symptom.' This overlap likely arises from semantic similarities or nested relationships within the text.

Similarly, the CMeEE V2 dataset initially showed poor category aggregation, with blurred boundaries and significant overlap under the original word vectors. The application of MFGFF significantly clarified these boundaries and improved data point aggregation, highlighting its effectiveness in enhancing feature differentiation. However, certain categories, such as medical examination item, medical procedure, and body still exhibited similarities, likely due to semantic overlaps in medical terminology.

In contrast, the CLUENER dataset, a general domain corpus, displayed a more even distribution of entities under the original word vectors, though noticeable category overlap was still present. The application of MFGFF greatly improved differentiation between entity types, which may be attributed to the dataset's inherent diversity in entity text, allowing fused features to better segregate categories.

In summary, our analysis highlights the complexities inherent in biomedical NER, including challenges like semantic overlap, domain-specific vocabulary, and nested entities. While MFGFF demonstrates significant advantages in enhancing feature representation, it still faces challenges, particularly in addressing category similarity. The varying performance of this technique across different datasets is closely tied to the datasets' characteristics and the semantic features of the entity categories, indicating a need for further optimization and adaptation in specific applications.

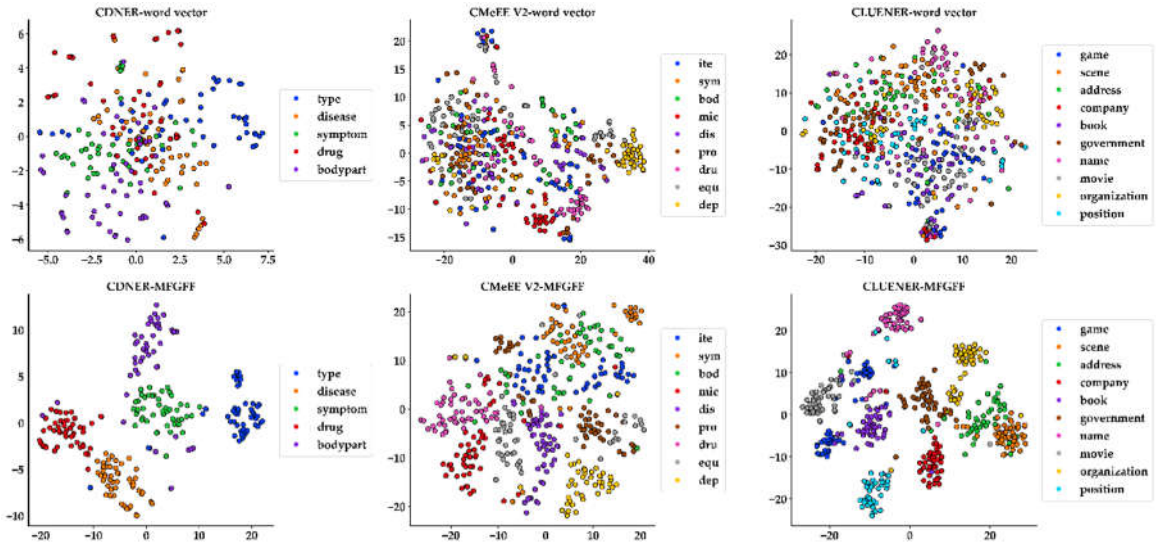

Figure 8. Visualization of token representations in feature space using t-SNE for data dimensionality reduction, with different colored points representing labelled entities of different types, including word vector features and MFGFF features, visualized on three datasets.

4.2. Nested Entity Predictive Analytic

 doi:10.20944/preprints202408.1620.v1

Based on the experimental results presented in Figure 9, we conducted a detailed evaluation of the effectiveness of nested entity recognition, comparing the performance of Fine-Tuning (FT) RoBERTa and MFGFF across different datasets. For the CDNER dataset, Fine-Tuning RoBERTa achieved a precision of 79.68%, a recall of 72.54%, and an F1 score of 75.98%. On the CMeEE V2 dataset, these metrics were notably lower, with a precision of 42.13%, a recall of 47.15%, and an F1 score of 44.50%. Compared to overall entity recognition, the F1 scores for CDNER and CMeEE V2 decreased by 16.03% and 28.82%, respectively, highlighting the significant challenges posed by nested entities in NER.

To address this challenge, we developed a high-quality nested entity dataset to enhance model training with more representative data. Implementing the MFGFF method further improved the model's capability to recognize nested entities. On the CDNER dataset, MFGFF achieved precision, recall, and F1 scores of 71.90%, 72.13%, and 71.91%, respectively, closely aligning with the performance of Fine-Tuning RoBERTa. On the more complex CMeEE V2 dataset, MFGFF yielded precision, recall, and F1 scores of 39.56%, 41.10%, and 40.32%, respectively. Although MFGFF's absolute values were slightly lower than those of Fine-Tuned RoBERTa, its stability and robustness in recognizing nested entities were more pronounced, particularly in the challenging CMeEE V2 dataset.

In conclusion, while nested entity recognition continues to present significant challenges in NER tasks, constructing high-quality datasets and adopting a MFGFF approach can significantly enhance model performance. The MFGFF method, with its superior stability and robustness, outperforms the fine-tuning-only strategy in recognizing nested entities, demonstrating both the effectiveness and potential application value of our proposed method.

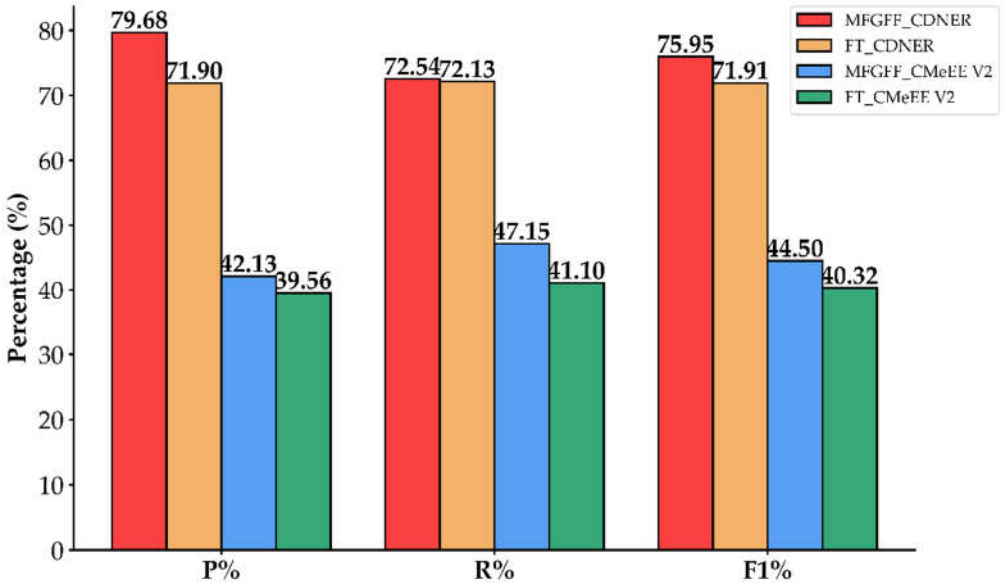

Figure 9. P, R, F1 evaluation results of fine-tuning method and MFGFF method on CDNER and CMeEE V2 datasets.

4.3. Comparative Analysis of Pre-rained Models

The choice of a pre-trained model for embedding is crucial to the recognition effectiveness of NER models. We compared the performance of five pre-trained models—BERT, RoBERTa, XLNet, MCBERT, and ERNIE—as illustrated in Figure 10.

RoBERTa outperforms the other models across all metrics, particularly in recall (92.18%) and F1 score (91.98%), where it leads significantly. This superior performance can be attributed to RoBERTa's use of a larger dataset, extended training time, and the removal of the Next Sentence Prediction task from BERT. These modifications enhance RoBERTa's ability to capture contextual information and manage long dependencies effectively. MCBERT, a model pre-trained specifically for the Chinese

 doi:10.20944/preprints202408.1620.v1

medical domain, performs comparably to BERT in medical text classification tasks, with accuracy and F1 scores of 90.25% and 90.49%, respectively. MCBERT's strength lies in its pre-training on a vast corpus of medical domain texts, enabling it to handle medical terminology and specialized sentence structures effectively. However, compared to RoBERTa and ERNIE, MCBERT is slightly weaker in precision and recall, likely due to the more extensive pre-training data and optimization strategies employed by RoBERTa and ERNIE, which result in better performance even in specialized areas like chicken disease NER. XLNet, on the other hand, lags behind the other models in precision (88.87%), recall (90.08%), and F1 score (89.44%). Despite its autoregressive architecture, which aims to merge the benefits of BERT and Transformer-XL to capture richer contextual information, XLNet's performance may degrade when handling shorter texts or when there is insufficient contextual information.

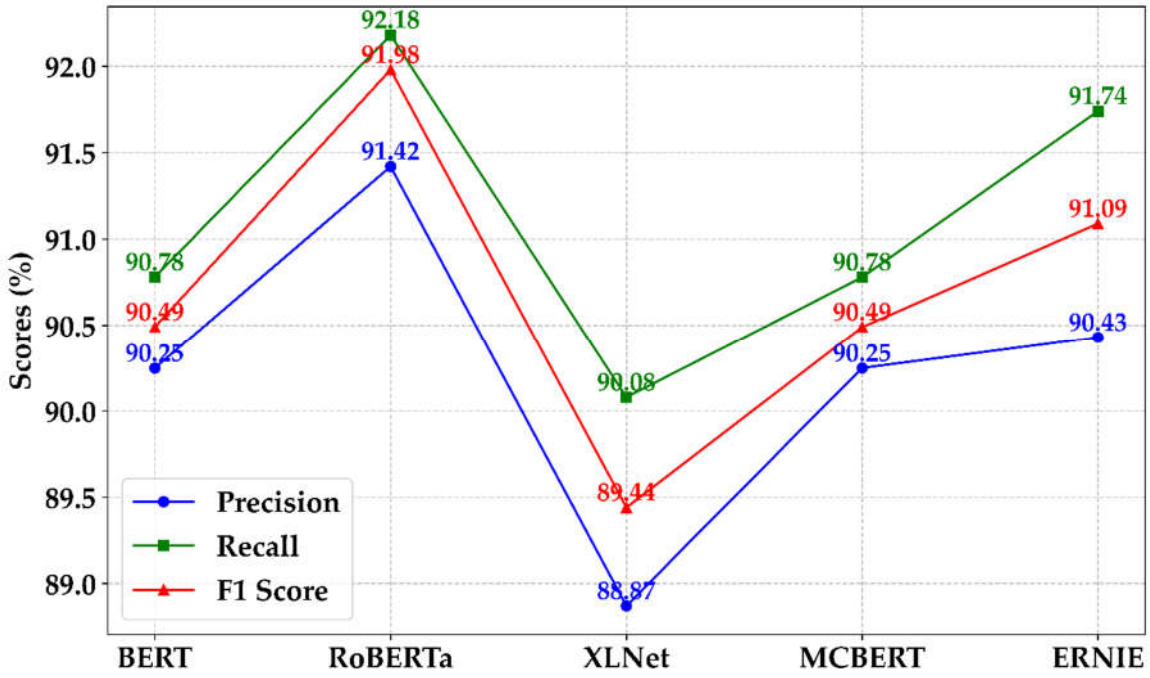

Figure 10. Comparison of evaluation results of 5 pre-trained models under MFGFF-BiLSTM-EGP model.

5. Conclusions

In this study, we developed the CDNER dataset specifically for chicken disease nested NER and proposed a nested NER model, MFGFF-BiLSTM-EGP, which leverages MFGFF and the EGP. The key conclusions are as follows:

1. The MFGFF-BiLSTM-EGP model exhibited strong performance across multiple datasets, achieving F1 scores of 91.98% on CDNER, 73.32% on CMeEE V2, and 82.54% on CLUENER. These results indicate that the model has a robust ability to generalize.
2. The model's recognition accuracy improved across all three encoders tested. The character encoder provided the most significant enhancement, increasing the model's performance by 9.31%, while the sentence encoder contributed a smaller improvement of 0.65%. Additionally,the MFGFF module effectively integrates various fine-grained feature vectors, further enhancing the model's accuracy .
3. The model also performed well in identifying nested entities, with F1 scores of 75.95% on CDNER and 44.50% on CMeEE V2. This represents an improvement of 4.04% and 4.18%, respectively, compared to using word vectors as embeddings, highlighting the effectiveness of the EGP module.

 doi:10.20944/preprints202408.1620.v1

In the future, we will focus on optimizing the model, improving its performance in small sample environments, and adapting it for use with other livestock species. Additionally, we plan to develop a chicken disease knowledge graph and an intelligent diagnosis system based on the model.

Author Contributions: Conceptualization, X.W. and C.P.; methodology, X.W.; software, X.W.; validation, X.W., Q.L. and Q.Y.; formal analysis, L.L. and L.D.; investigation, P.L.; resources, C.P.; data curation, C.P.; writ-ing—original draft preparation, X.W.; writing—review and editing, X.W. and L.Z.; visualization, X.W. and R.J.; supervision, R.G. and W.W.; project administration, Q.L. and L.Y.; funding acquisi-tion, Q.L.

Funding: This research was funded by the National Science and Technology Major Project (2021ZD0113802) and the Technological Innovation Capacity Construction of Beijing Academy of Agricultural and Forestry Sciences (KJCX20240318).

Informed Consent Statement: Not applicable.
Data Availability Statement: The original data presented in the study are openly available in https://github.com/wangxiajun68/chicken-disease.

Acknowledgments: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

The steps of the MFGFF-BiLSTM-EGP model are shown in Algorithm A1 below:

Algorithm A1. Pseudo-code for the MFGFF-BiLSTM-EGP nested named entity recognition mode.

Input: Pre-trained model RoBERTa, word vectors GloVe, sentence embeddings SBERT, network BiLSTM, Efficient Global Pointer EGP, The number of iterations E.
Output: Predicted entity spans E, optimal model weights $\hat{\theta}$.

For $e = 1, \ldots, E$ do
    $h_i \leftarrow \mathrm{RoBERTa}(c_i), \forall c_i \in X$
    $w_i' \leftarrow \log(1 + f_j) \times \mathrm{GloVe}(w_j)$, according to Equation (4) to (5).
    $h_i' \leftarrow \mathrm{MultiHead}(h_i, \mathrm{w}_i'), \forall \mathrm{w}_j$ containing c_i, according to Equation (6) to (11).
    $u \leftarrow \mathrm{SBERT}(X)$, according to Equation (12) to (14).
    $H_i \leftarrow \mathrm{Concat}(\mathrm{h}_i, \mathrm{h}_i', u), \forall i$
    Contextual Output $\leftarrow \mathrm{BiLSTM}(\{H_1, H_2, \ldots, H_n\})$
    $s_{i,j}^{\alpha} \leftarrow (W_q H_i + b_q) \times (W_k H_j + b_k), \forall (c_i, c_j)$, according to Equation (17) to (20).
    $\mathcal{L} \leftarrow \mathrm{CircleLoss}(s_{i,j}^{\alpha}, \hat{Y})$, according to Equation (21) to (22).
    $\hat{\theta} \leftarrow \mathrm{argmin}_{\theta} \mathcal{L}$
    If $F1 > F1_{max}$
        $F1_{max} \leftarrow F1$
        save $\hat{\theta}$
End for
Output: Final predicted spans $\hat{Y}$, best weights $\hat{\theta}$.

References

1. Han, H.; Xue, X.; Li, Q.; Gao, H.; Wang, R.; Jiang, R.; Ren, Z.; Meng, R.; Li, M.; Guo, Y.; et al. Pig-Ear Detection from the Thermal Infrared Image Based on Improved YOLOv8n. *Intelligence & Robotics* **2024**, *4*, 1–19, doi:10.20517/ir.2024.02.
2. Hou, G.; Jian, Y.; Zhao, Q.; Quan, X.; Zhang, H. Language Model Based on Deep Learning Network for Biomedical Named Entity Recognition. *Methods* **2024**, *226*, 71–77, doi:10.1016/j.ymeth.2024.04.013.
3. Nadeau, D.; Sekine, S. A Survey of Named Entity Recognition and Classification. *LI* **2007**, *30*, 3–26, doi:10.1075/li.30.1.03nad.
4. Jehangir, B.; Radhakrishnan, S.; Agarwal, R. A Survey on Named Entity Recognition — Datasets, Tools, and Methodologies. *Natural Language Processing Journal* **2023**, *3*, 100017, doi:10.1016/j.nlp.2023.100017.

5. Morwal, S. Named Entity Recognition Using Hidden Markov Model (HMM). *IJNLC* **2012**, *1*, 15–23, doi:10.5121/ijnlc.2012.1402.
6. De Martino, A.; De Martino, D. An Introduction to the Maximum Entropy Approach and Its Application to Inference Problems in Biology. *Heliyon* **2018**, *4*, e00596, doi:10.1016/j.heliyon.2018.e00596.
7. Ju, Z.; Wang, J.; Zhu, F. Named Entity Recognition from Biomedical Text Using SVM. In Proceedings of the 2011 5th International Conference on Bioinformatics and Biomedical Engineering; IEEE: Wuhan, China, May 2011; pp. 1–4.
8. Liu, K.; Hu, Q.; Liu, J.; Xing, C. Named Entity Recognition in Chinese Electronic Medical Records Based on CRF. In Proceedings of the 2017 14th Web Information Systems and Applications Conference (WISA); IEEE: Liuzhou, Guangxi Province, China, November 2017; pp. 105–110.
9. Dash, A.; Darshana, S.; Yadav, D.K.; Gupta, V. A Clinical Named Entity Recognition Model Using Pretrained Word Embedding and Deep Neural Networks. *Decision Analytics Journal* **2024**, *10*, 100426, doi:10.1016/j.dajour.2024.100426.
10. Zhang, R.; Zhao, P.; Guo, W.; Wang, R.; Lu, W. Medical Named Entity Recognition Based on Dilated Convolutional Neural Network. *Cognitive Robotics* **2022**, *2*, 13–20, doi:10.1016/j.cogr.2021.11.002.
11. Lerner, I.; Paris, N.; Tannier, X. Terminologies Augmented Recurrent Neural Network Model for Clinical Named Entity Recognition. *Journal of Biomedical Informatics* **2020**, *102*, 103356, doi:10.1016/j.jbi.2019.103356.
12. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Computation* **1997**, *9*, 1735–1780, doi:10.1162/neco.1997.9.8.1735.
13. Chang, C.; Tang, Y.; Long, Y.; Hu, K.; Li, Y.; Li, J.; Wang, C.-D. Multi-Information Preprocessing Event Extraction With BiLSTM-CRF Attention for Academic Knowledge Graph Construction. *IEEE Trans. Comput. Soc. Syst.* **2023**, *10*, 2713–2724, doi:10.1109/TCSS.2022.3183685.
14. An, Y.; Xia, X.; Chen, X.; Wu, F.-X.; Wang, J. Chinese Clinical Named Entity Recognition via Multi-Head Self-Attention Based BiLSTM-CRF. *Artificial Intelligence in Medicine* **2022**, *127*, 102282, doi:10.1016/j.artmed.2022.102282.
15. Deng, N.; Fu, H.; Chen, X. Named Entity Recognition of Traditional Chinese Medicine Patents Based on BiLSTM-CRF. *Wireless Communications and Mobile Computing* **2021**, *2021*, 1–12, doi:10.1155/2021/6696205.
16. Ma, P.; Jiang, B.; Lu, Z.; Li, N.; Jiang, Z. Cybersecurity Named Entity Recognition Using Bidirectional Long Short-Term Memory with Conditional Random Fields. *Tinshhua Sci. Technol.* **2021**, *26*, 259–265, doi:10.26599/TST.2019.9010033.
17. Baigang, M.; Yi, F. A Review: Development of Named Entity Recognition (NER) Technology for Aeronautical Information Intelligence. *Artif Intell Rev* **2023**, *56*, 1515–1542, doi:10.1007/s10462-022-10197-2.
18. Wang, Y.; Tong, H.; Zhu, Z.; Li, Y. Nested Named Entity Recognition: A Survey. *ACM Trans. Knowl. Discov. Data* **2022**, *16*, 1–29, doi:10.1145/3522593.
19. Shen, D.; Zhang, J.; Zhou, G.; Su, J.; Tan, C.-L. Effective Adaptation of a Hidden Markov Model-Based Named Entity Recognizer for Biomedical Domain. In Proceedings of the Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine -; Association for Computational Linguistics: Sapporo, Japan, 2003; Vol. 13, pp. 49–56.
20. Yang, J.; Zhang, T.; Tsai, C.-Y.; Lu, Y.; Yao, L. Evolution and Emerging Trends of Named Entity Recognition: Bibliometric Analysis from 2000 to 2023. *Heliyon* **2024**, *10*, e30053, doi:10.1016/j.heliyon.2024.e30053.
21. Ming, H.; Yang, J.; Gui, F.; Jiang, L.; An, N. Few-Shot Nested Named Entity Recognition. *Knowledge-Based Systems* **2024**, *293*, 111688, doi:https://doi.org/10.1016/j.knosys.2024.111688.
22. Huang, H.; Lei, M.; Feng, C. Hypergraph Network Model for Nested Entity Mention Recognition. *Neurocomputing* **2021**, *423*, 200–206, doi:10.1016/j.neucom.2020.09.077.
23. Jiang, D.; Ren, H.; Cai, Y.; Xu, J.; Liu, Y.; Leung, H. Candidate Region Aware Nested Named Entity Recognition. *Neural Networks* **2021**, *142*, 340–350, doi:10.1016/j.neunet.2021.02.019.
24. Nivre, J.; McDonald, R. Integrating Graph-Based and Transition-Based Dependency Parsers.
25. Huang, P.; Zhao, X.; Hu, M.; Tan, Z.; Xiao, W. *T* 2 *-NER* : A T Wo-Stage Span-Based Framework for Unified Named Entity Recognition with T Emplates. *Transactions of the Association for Computational Linguistics* **2023**, *11*, 1265–1282, doi:10.1162/tacl_a_00602.
26. Li, F.; Wang, Z.; Hui, S.C.; Liao, L.; Zhu, X.; Huang, H. A Segment Enhanced Span-Based Model for Nested Named Entity Recognition. *Neurocomputing* **2021**, *465*, 26–37, doi:10.1016/j.neucom.2021.08.094.
27. Jiang, W. A Method for Ancient Book Named Entity Recognition Based on BERT-Global Pointer. *IJCSIT* **2024**, *2*, 443–450, doi:10.62051/ijcsit.v2n1.47.

28. Zhang, P.; Liang, W. Medical Name Entity Recognition Based on Lexical Enhancement and Global Pointer. *IJACSA* **2023**, *14*, doi:10.14569/IJACSA.2023.0140369.
29. Zhang, X.; Luo, X.; Wu, J. A RoBERTa-GlobalPointer-Based Method for Named Entity Recognition of Legal Documents. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN); IEEE: Gold Coast, Australia, June 18 2023; pp. 1–8.
30. Cong, L.; Cui, R.; Dou, Z.; Huang, C.; Zhao, L.; Zhang, Y.; Chen, C.; Su, C.; Li, J.; Qu, C. Named Entity Recognition for Power Data Based on Lexical Enhancement and Global Pointer. In Proceedings of the Third International Conference on Electronic Information Engineering, Big Data, and Computer Technology (EIBDCT 2024); Zhang, J., Sun, N., Eds.; SPIE: Beijing, China, July 19 2024; p. 61.
31. Yadav, V.; Bethard, S. A Survey on Recent Advances in Named Entity Recognition from Deep Learning Models 2019.
32. Liu, Z.; Jiang, F.; Hu, Y.; Shi, C.; Fung, P. NER-BERT: A Pre-Trained Model for Low-Resource Entity Tagging 2021.
33. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.
34. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding.
35. Zhang, N.; Jia, Q.; Yin, K.; Dong, L.; Gao, F.; Hua, N. Conceptualized Representation Learning for Chinese Biomedical Text Mining 2020.
36. Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; Liu, Q. ERNIE: Enhanced Language Representation with Informative Entities 2019.
37. Ma, R.; Peng, M.; Zhang, Q.; Wei, Z.; Huang, X. Simplify the Usage of Lexicon in Chinese NER. In Proceedings of the Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; Association for Computational Linguistics: Online, 2020; pp. 5951–5960.
38. Zhao, J.; Cui, M.; Gao, X.; Yan, S.; Ni, Q. Chinese Named Entity Recognition Based on BERT and Lexicon Enhancement. In Proceedings of the Proceedings of the 2022 4th International Conference on Robotics, Intelligent Control and Artificial Intelligence; ACM: Dongguan China, December 16 2022; pp. 597–604.
39. Zhang, J.; Guo, M.; Geng, Y.; Li, M.; Zhang, Y.; Geng, N. Chinese Named Entity Recognition for Apple Diseases and Pests Based on Character Augmentation. *Computers and Electronics in Agriculture* **2021**, *190*, 106464, doi:10.1016/j.compag.2021.106464.
40. Liu, Q.; Zhang, L.; Ren, G.; Zou, B. Research on Named Entity Recognition of Traditional Chinese Medicine Chest Discomfort Cases Incorporating Domain Vocabulary Features. *Computers in Biology and Medicine* **2023**, *166*, 107466, doi:10.1016/j.compbiomed.2023.107466.
41. Sun, M.; Yang, Q.; Wang, H.; Pasquine, M.; Hameed, I.A. Learning the Morphological and Syntactic Grammars for Named Entity Recognition. *Information* **2022**, *13*, 49, doi:10.3390/info13020049.
42. Tian, Y.; Shen, W.; Song, Y.; Xia, F.; He, M.; Li, K. Improving Biomedical Named Entity Recognition with Syntactic Information. *BMC Bioinformatics* **2020**, *21*, 539, doi:10.1186/s12859-020-03834-6.
43. Luoma, J.; Pyysalo, S. Exploring Cross-Sentence Contexts for Named Entity Recognition with BERT 2020.
44. Jia, B.; Wu, Z.; Wu, B.; Liu, Y.; Zhou, P. Enhanced Character Embedding for Chinese Named Entity Recognition. *Measurement and Control* **2020**, *53*, 1669–1681, doi:10.1177/0020294020952456.
45. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need.
46. 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**, doi:10.48550/ARXIV.1907.11692.
47. Hongying, Z.; Wenxin, L.; Kunli, Z.; Yajuan, Y.; Baobao, C.; Zhifang, S. Building a Pediatric Medical Corpus: Word Segmentation and Named Entity Annotation. In *Chinese Lexical Semantics*; Liu, M., Kit, C., Su, Q., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, 2021; Vol. 12278, pp. 652–664 ISBN 978-3-030-81196-9.
48. Xu, L.; Liu, W.; Li, L.; Liu, C.; Zhang, X. CLUENER2020: FINE-GRAINED NAMED ENTITY RECOGNITION DATASET AND BENCHMARK FOR CHINESE.
49. Song, Y.; Shi, S.; Li, J.; Zhang, H. Directional Skip-Gram: Explicitly Distinguishing Left and Right Context for Word Embeddings. In Proceedings of the Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language

 doi:10.20944/preprints202408.1620.v1

Technologies, Volume 2 (Short Papers); Association for Computational Linguistics: New Orleans, Louisiana, 2018; pp. 175–180.
50. Zhang, Y.; Yang, J. Chinese NER Using Lattice LSTM. **2018**, doi:10.48550/ARXIV.1805.02023.
51. Li, X.; Yan, H.; Qiu, X.; Huang, X. FLAT: Chinese NER Using Flat-Lattice Transformer 2020.
52. Li, X.; Feng, J.; Meng, Y.; Han, Q.; Wu, F.; Li, J. A Unified MRC Framework for Named Entity Recognition 2022.
53. Cui, X.; Yang, Y.; Li, D.; Qu, X.; Yao, L.; Luo, S.; Song, C. Fusion of SoftLexicon and RoBERTa for Purpose-Driven Electronic Medical Record Named Entity Recognition. *Applied Sciences* **2023**, *13*, 13296, doi:10.3390/app132413296.