CPMFA: A Character Pair-Based Method for Chinese Nested Named Entity Recognition

Xiayan Ji[1], Lina Chen[2(✉)], Fangyao Shen[2], Hongjie Guo[2], and Hong Gao[2]

[1] College of Physics and Electronic Information Engineering, Zhejiang Normal University, Jinhua, China
[2] School of Computer Science and Technology, Zhejiang Normal University, Jinhua, China
chenlina@zjnu.cn

Abstract. Chinese Nested Named Entity Recognition (CNNER) faces several challenges due to the language diversity phenomena, the complexity of the language, and the imbalanced distribution of entity types in Chinese text. To address these challenges in CNNER, we propose a new method called CPMFA (Character Pair-based method with Multi-feature representation and Attention mechanism). The CPMFA method predicts the predefined relations of character pairs in a sentence, and identifies nested named entities based on these relations. First, our method utilizes the pre-trained language model LERT (Linguistically-motivated Bidirectional Encoder Representation from Transformer), and Bidirectional Long Short-Term Memory (BiLSTM) to generate comprehensive and precise character representations. Second, our method uses multi-feature representation to capture complex semantic information within the text, and employs the Pyramid Squeeze Attention (PSA) module to emphasize key features. Finally, to overcome the challenge of the imbalanced distribution of entity types, PolyLoss function is integrated into our model training process. Results of experiments show that the proposed CPMFA method achieves an F1 score of 83.79%. Compared to other mainstream span-based methods, the proposed CPMFA method has excellent performance in CNNER.

Keywords: Chinese character pair · Chinese nested named entity recognition · Multi-feature representation · Attention mechanism · Pre-trained language model

1 Introduction

Nested named entities are entities that have overlapping structures. The majority of existing named entity recognition models have difficulty in accurately identifying such complex nested named entities, and cannot capture specific and detailed entity information in the text. Therefore, recognizing nested named entities has always been a highly challenging task.

X. Yang et al. (Eds.): ADMA 2023, LNAI 14176, pp. 200–212, 2023.
https://doi.org/10.1007/978-3-031-46661-8_14

In recent years, researchers have increasingly focused on the application of deep learning models in Chinese Nest Named Entity Recognition (CNNER). However, the literature in this field remains limited. Zhang et al. [1] proposed a novel boundary-aware layered neural model (BLNM) with segmentation attention, which captures the potential word information and enhance Chinese character representation, but is ineffective when dealing with a combination of Chinese and English text. Yu et al. [2] introduced a layered regional exhaustive model (LREM), which utilizes a neural network to explore exhaustive combinations of sentences; however, it requires an improved understanding of Chinese semantic language and does not fully utilize critical information within the text. Li et al. [3] developed a multi-layer joint learning model that uses a self-attention mechanism to effectively aggregate entity information features and identify nested entities layer by layer. However, the method faces challenges in handling imbalanced entity classes.

Previous studies on nested named entity recognition (NNER) primarily focus on English texts. There are notable differences between Chinese and English. Models that perform well in English NNER often encounter challenges when applied to Chinese texts, resulting in unsatisfactory outcomes. Existing researches has identified the following difficulties in CNNER:

1. Language diversity phenomena in Chinese text: with the constant integration and evolution of language and culture, the language diversity phenomena in Chinese technical materials and reference documents continue to increase. The mixture of Chinese, English, numbers, symbols, and other linguistic expressions presents a significant challenge for CNNER.
2. Complexity of the Chinese language: in Chinese text, the presence of multiple layers and high frequency of nested named entities, along with polysemous phenomena, results in a significantly challenging task of CNNER.
3. Entity type imbalance in Chinese text: for practical applications, entity type numbers distribution in Chinese text often follows a long-tail distribution where only a few entity types occupy the majority of data, significantly impeding model recognition performance.

To address the previously outlined challenges, this study proposes a Character Pair-based method with Multi-feature representation and an Attention mechanism (CPMFA). The proposed method makes the following contributions:

1. To overcome the challenge of language diversity phenomena in Chinese text, we introduce the linguistically-motivated pre-trained language model called LERT (Linguistically-motivated Bidirectional Encoder Representation from Transformer), as well as Bidirectional Long Short-Term Memory (BiLSTM), to vectorize the text. This approach improves the quality of character representation in Chinese text.
2. To address the complexity of Chinese language, this study utilizes multi-feature representation to incorporate comprehensive information from the text, as well as adopts the Pyramid Squeeze Attention (PSA) module to prioritize key features.

3. To deal with the long-tail distribution problem of entity class numbers in Chinese text, PolyLoss loss function is employed to improve the model's recognition performance.

This paper provides an overview of related work in Sect. 2, introduces the CPMFA method in Sect. 3, evaluates its performance on a Chinese nested named entity dataset with a detailed performance analysis in Sect. 4, and draws conclusions in Sect. 5.

2 Related Work

NNER utilizes two primary methods: sequence labeling-based and span-based.

Using a sequence labeling-based method, a label sequence with the highest probability is generated, which infers the boundaries and types of named entities more efficiently. Huang et al. [4] first utilized the BiLSTM-CRF model for named entity recognition (NER), which enables the capture of contextual information and dependencies between labels. To handle nested entities, Strakova et al. [5] combined multiple labels to create new ones. However, because characters in nested entities can have multiple labels, decoding named entities using the sequence labeling-based approach is more complex.

Using a span-based method, named entities are identified by categorizing the subsequences of the text sequence. Li et al. [6] detected entity fragments by exploring every possible text span and applying relationship classification to discover possible relationships among sets of entity fragments, thereby achieving recognition of nested entities. Xia et al. [7] proposed the Multi-Grained Named Entity Recognition (MGNER) model, which comprises a detection system and a classifier. The model aims to identify and categorize all potential fragments of entities. Li et al. [8] accomplished entity boundary determination and entity type recognition by predicting word relationships. However, span-based methods focus primarily on contextual information and do not fully explore the underlying information of the text.

Basic NNER methods have limitations in mining deep textual information. Therefore, some scholars have suggested incorporating attention mechanisms in NNER models to improve their performance and effectiveness [9]. Cui et al. [10] proposed a Multi-Head Adjacent Attention-based Pyramid Layered model to capture the dependency relationships between adjacent characters in the input text. Similarly, Rodríguez et al. [11] proposed an attention mechanism based specifically on the use of elements of the noun syntactic type to capture syntactic information in the text. These models show that attention mechanisms can be used to extract deep textual information.

The pre-trained language models, represented by Bidirectional Encoder Representation from Transformers (BERT) [12], have shown remarkable performance in many NER tasks. To address the issue of polysemy in Chinese, Li et al. [13] suggested a syntactic dependency guided BERT-BiLSTM-GAM-CRF model. Yu et al. [14] incorporated BERT into a previously utilized NER model to extract entities from mineral literature.

Xu et al. [15] proposed an approach that utilized BERT and a supervised multi-head self-attention mechanism to capture lexical correlations. This approach combined both attention mechanism and pre-trained language model, achieving excellent performance in the task of nested named entity recognition. In this study, the LERT pre-trained language model and Pyramid Squeeze Attention (PSA) module were employed to further enhance performance of the named entity recognition model. In real-world scenarios, the uneven distribution of entity quantities is a common issue. Various types of entities can exhibit distinct frequencies and distributions within the text. Such type imbalance can significantly impair the performance of NNER models.

The loss function measures the discrepancy between predicted and true labels during training. Focal Loss [16], an adaptive loss function utilized in the field of image recognition, aims to address type imbalance by reducing the impact of more frequently appearing samples while increasing the weight of the less frequent ones. Leng et al. [17] introduced PolyLoss, a linear combination of polynomial functions that can be customized to match the unique characteristics of the dataset. This method improves upon traditional loss functions by accounting for the nuances of the data, resulting in more accurate predictions. Although there has been extensive research conducted in the field of NNER, there are currently no literature references that examine the effectiveness of implementing PolyLoss within this area. We present the first study to use PolyLoss in NNER and demonstrate its ability in enhancing performance.

3 CPMFA Method

Our inspiration for defining the task of CNNER as the prediction of relations between characters comes from Li et al. [8]. Therefore, we propose a Character Pair-based method with Multi-feature Representation and Attention mechanism (CPMFA).

CPMFA model predicts the relations of character pairs in a sentence, based on three predefined relations: None, Next Neighboring Character (NNC), and Tail-Head Character-* (THC-*, where * represents the entity type). None indicates no relation between two characters. Next Neighboring Character (NNC) indicates whether two characters are adjacent within an entity. Tail-Head Character-* (THC-*) identifies the entity boundary and entity type. THC-* denotes the tail and head boundaries, while * represents the entity type.

3.1 CPMFA Model

Figure 1 depicts the architecture of the CPMFA model, consisting of three discrete components: Encoder Layer, Feature Extraction Layer, and Decoder Layer.

Encoder Layer. The Encoder Layer incorporates LERT and BiLSTM to produce superior character representations. LERT, a pre-trained language model, is applied to attain a comprehensive representation of textual information.

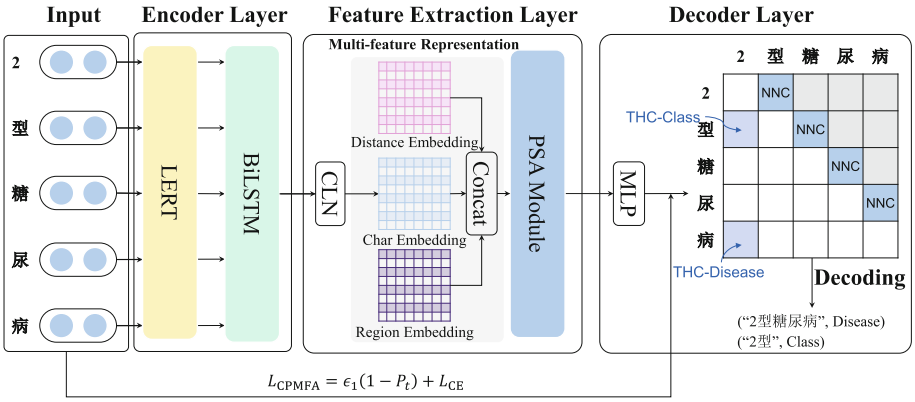


Fig. 1. The architecture of the CPMFA model.

For an input sentence $C = [c_1, c_2, ..., c_n]$, LERT produces vectorized text $X = [x_1, x_2, ..., x_n]$, as demonstrated in Eq. (1). To further enhance the model's understanding of textual context, the BiLSTM is employed to produce the result $H = [h_1, h_2, ..., h_n] \in R^{N \times d_h}$, as mentioned in Eq. (2). Here, d_h is the dimensional aspects of character representation.

$$X = LERT(C) \tag{1}$$

$$H = BiLSTM(X) \tag{2}$$

Feature Extraction Layer. The Feature Extraction Layer is used to extract pertinent features of character pair relations, thereby enabling accurate prediction of named entities. The layer comprises three components: a conditional layer normalization (CLN), multi-feature representation, and a PSA module.

Conditional Layer Normalization. The CLN generates grid representations between two characters, which are fundamental for extracting pertinent features related to character pairs, as illustrated in Eq. (3). This matrix $V \in \mathbb{R}^{N \times N \times d_h}$ is the result of the CLN, where V_{ij} stands for the representation of the character pair (c_i, c_j). Since both NNC and THC-* relations are directional, V_{ij}, which represents the character pair (c_i, c_j), can be considered as a combination of the representations of c_i and c_j, denoted by h_i and h_j respectively. Here, h_i is the condition for producing the gain parameter $\gamma_{ij} = W_\alpha h_i + b_\alpha$ as well as bias $\lambda_{ij} = W_\beta h_i + b_\beta$ of layer normalization. As mentioned in Eq. (4), μ and σ represent the mean and standard deviation of the elements present in h_j.

$$V_{ij} = CLN\left(h_i, h_j\right) = \gamma_{ij} \odot \left(\frac{h_j - \mu}{\sigma}\right) + \lambda_{ij} \tag{3}$$

$$\mu = \frac{1}{d_h}\sum_{k=1}^{d_h} h_{jk}, \sigma = \sqrt{\frac{1}{d_h}\sum_{k=1}^{d_h}\left(h_{jk} - \mu\right)^2} \tag{4}$$

Multi-feature Representation. Constructing multi-feature representations facilitates the integration of features from various perspectives and improves the accuracy of predicting character pair relations. Referring to Eq. (5), the multi-feature representation $E \in \mathbb{R}^{N\times N\times d_e}$ is obtained through concatenating distance embedding $E_D \in \mathbb{R}^{N\times N\times d_d}$, region embedding $E_R \in \mathbb{R}^{N\times N\times d_r}$, and character embedding $V \in \mathbb{R}^{N\times N\times d_h}$. The distance embedding shows the relative position of characters; the region embedding displays up-down triangle area information on the grid; and character embedding conveys semantic information.

$$E = Concat([E_D, E_R, V]) \tag{5}$$

PSA module. The PSA module can efficiently focus on key features in multi-feature representations and process character-to-character interaction information. The PSA module consists of two modules: the Squeeze and Concat (SPC) module and the SEWeight module [18].

As shown in Fig. 2, the SPC module is composed of multiple parallel branches that operate independently. Each branch takes E' as input and contains a number of channels d_e. E' is obtained by permuting the dimensions of the input multi-feature representation E. Grouped convolutions and convolution kernels of various sizes are utilized to compress the channels and capture spatial information across different scales. The resulting feature maps from the SPC module are represented by $F \in \mathbb{R}^{d_e\times N\times N}$, as depicted in Eq. (6).

$$F = SPC(E') \tag{6}$$

The SEWeight module utilizes input F, as shown in Eq. (7), to generate an attention vector Z_i for that branch.

$$Z_i = SEWeight\left(F_i\right), \quad i \in 1, 2, 3, 4 \tag{7}$$

Equation (8) shows the concatenation of attention weights from each branch, which creates the multi-scale channel weight. Multiplying the multi-scale fusion feature map with the multi-scale fusion channel weight in a channel-wise operation creates an adaptive channel weight for the feature map. The "Concat" denotes the concatenate operator, while $\odot$ denotes the element-wise product operator.

$$M_F = Concat([W_1, W_2, \ldots, W_4]) \odot F \tag{8}$$

Decoder Layer. The Decoder Layer comprises two main parts: predicting character pair relations and decoding identified named entities.

Predict. Our proposed model is designed to predict character pair relations by calculating the relation's probability of belonging to a specific class. The feature extraction layer and dimensional transposition produce the feature grid representation of character pairs, denoted as M'_F. The relation score y'_{ij} of the

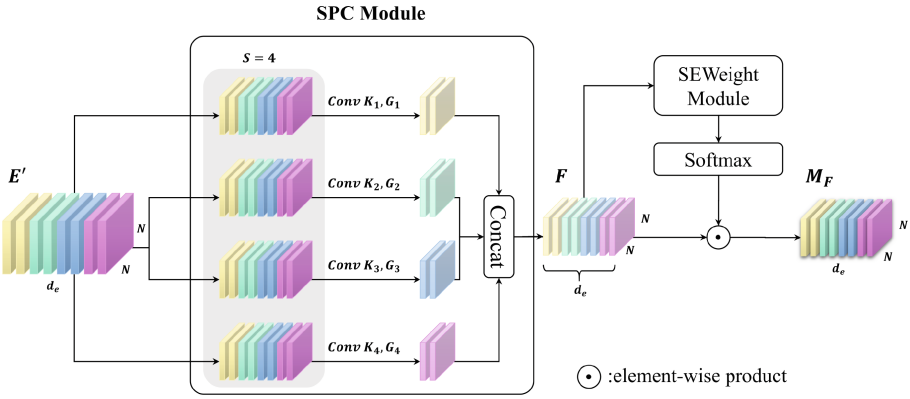


Fig. 2. The PSA module's structure diagram and a detailed description of the SPC module at S=4. K refers to the convolution kernel's size, G represents the group size, and "Concat" represents the feature concatenation in the channel dimension.

character pair (c_i, c_j) is computed by the Multilayer Perceptron (MLP), as shown in Eq. (9). To evaluate the probability $\hat{y}_{ij}$ of character pair (c_i, c_j) belonging to specific classes, the Softmax function is employed, as indicated in Eq. (10).

$$y'_{ij} = MLP(M'_{F_{ij}}) \tag{9}$$

$$\hat{y}_{ij} = Softmax(y'_{ij}) \tag{10}$$

Decode. We extract named entities by decoding the relations of character pairs. Relations of character pairs establish a directed graph, including nodes for characters and edges for relations. The model identifies specific pathways connecting distinct characters to one another, with each pathway mapped to a unique entity.

3.2 Training

Loss Function. We integrate the PolyLoss framework into our CPMFA model to resolve the long-tail entity type distribution issue in Chinese text. Tuning the polynomial coefficients in the PolyLoss-based loss function can optimize the model's performance for different datasets and tasks.

Our objective during training is to minimize L_{CPMFA}, as shown in Eq. (11). Here, L_{CE}, P_t, and ϵ represent the cross-entropy loss function, the probability of the model's true class label, and an adjustable hyperparameter, respectively.

In Eq. (12) and Eq. (13), the symbols used include N, representing the number of characters present in the sentence. Additionally, y_{ij} denotes a binary vector used to represent the actual relation of the character pair (c_i, c_j). The predicted probability vector, in contrast, is denoted as $\hat{y}_{ij}$. Lastly, r signifies the r-th relation contained in a predefined relation set, $\mathcal{R}$.

$$L_{CPMFA} = L_{CE} + \epsilon\,(1 - P_t) \tag{11}$$

$$L_{CE} = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{r=1}^{|\mathcal{R}|} y_{ij} log \hat{y}_{ij} \tag{12}$$

$$P_t = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{r=1}^{|\mathcal{R}|} y_{ij} \hat{y}_{ij} \tag{13}$$

4 Experiments and Results

4.1 Dataset

We utilized DiaKG, the authoritative dataset for the Chinese diabetes domain, and followed the division protocol established by Chang et al. [19] to extract the train, validation, and test sets.

Table 1 displays the statistical information for different granularities of the DiaKG dataset. The dataset contains a proportion of nested entities, up to 22% of it. Improving the precision of nested named entity recognition has the potential to enhance the overall model's performance.

Table 1. The statistical information on the granularity of the DiaKG dataset.

Granularity	Statistics	Train	Dev	Test	Total
Sentence	Total	4906	1636	1636	8178
	Sentences with nested named entities	3550 (72.36%)	1205 (73.66%)	1164 (71.15%)	6255 (76.49%)
	Avg. sentence length	151.34	153.68	150.63	151.68
Entity	Total	65774	22417	21496	109687
	nested named entities	**11101(16.88%)**	**3771(16.82%)**	**3516(16.36%)**	**24155(22.02%)**
	Avg. entity length	4.37	4.37	4.38	4.37
	Max number of nested layers	3	3	2	3

Figure 3 presents the frequency of annotations for the 18 entity types in the DiaKG dataset. The figure shows that the number of entity types are diverse and follow a long-tailed distribution, indicating data imbalance.

4.2 Evaluation Metrics

In this study, we follow the exact matching pattern to evaluate the performance of NNER. That is to say, a predicted entity is considered as correctly identified only when its predicted boundaries and types exactly match the annotated results in the dataset. Currently, in CNNER tasks, precision (P), recall (R), and F1 score (F1) are commonly used to evaluate the performance [20]. Precision measures the model's ability to correctly predict entities, while Recall measures the model's ability to identify all entities. The F1 score is the harmonic mean of precision and recall.

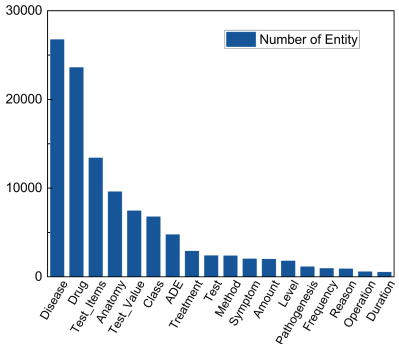


Fig. 3. Distribution of entities and number of entities in DiaKG dataset.

Precision, recall, and F1 score are calculated based on the number of true positives (TP), false positives (FP), and false negatives (FN), as shown in Eq. (14), Eq. (15) and Eq. (16).

$$P = \frac{TP}{TP + FP} \tag{14}$$

$$R = \frac{TP}{TP + FN} \tag{15}$$

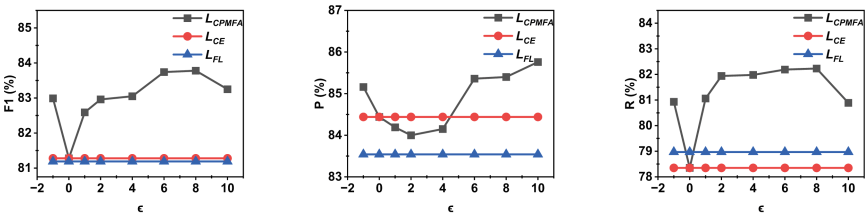
$$F1 = \frac{2 \times P \times R}{P + R} \tag{16}$$

4.3 Experimental Setting

The development language used in this study is Python and the deep learning framework used is Pytorch, which was trained on an NVIDIA 3090 Ti graphics card. Regarding the model hyper-parameters, we set the dimension of the LERT embedding to 768, the dimension of the LSTM hidden layer to 478, the batch size to 8, and the initial learning rate to 0.001. To prevent the model from overfitting, early stopping criteria and a dropout rate of 0.5 were employed in this study.

4.4 Results

Comparison of Loss Function. This study evaluates the performance of three loss functions: L_{CPMFA}, cross-entropy (L_{CE}), and focal (L_{FL}). The hyperparameter ϵ of L_{CPMFA} can be adjusted. Figure 4(a) and Fig. 4(c) demonstrate that models trained with L_{CPMFA} perform better in F1 score and recall than those using L_{CE} and L_{FL} functions. Figure 4(b) illustrates that, when $\epsilon \in 1, 2, 4$, models trained with L_{CPMFA} have lower precision compared to models trained with L_{CE}. However, models trained with L_{CPMFA} have higher precision than models trained with L_{FL}. In conclusion, using the L_{CPMFA} loss function and customizing the hyperparameters based on dataset characteristics could improve the CPMFA model's ability to identify named entities.



(a) F1 comparison (b) P comparison (c) R comparison

Fig. 4. Performances of our model trained with different loss functions.

Necessity of Multi-feature Representation. We conducted ablation experiments on the embeddings to verify the significance of multi-feature representation. The results are presented in Table 2. Using only character embedding for input features yielded the lowest F1 score. Integrating distance and region embedding resulted in a noteworthy improvement in the performance of the model. The integration of character and region embedding attained optimal precision, with a score of 85.41%. However, the multi-feature representation recorded only 0.01% lower precision, indicating a negligible difference between them. Our multi-feature representation, which integrated character, distance, and region embedding, achieved excellent results in both recall and F1 score. Therefore, the incorporation of multi-feature representation could effectively represent the relations of character pairs and enhance the model's recognition performance.

Table 2. Performances of our model with different feature embedding. The bold value indicates the optimal results.

	F1 (%)	P (%)	R (%)
Char Embedding	80.61	82.99	78.36
Char Embedding+Distance Embedding	82.90	84.11	81.73
Char Embedding+Region Embedding	83.05	**85.41**	80.81
Ours (Char Embedding+Distance Embedding+Region Embedding)	**83.79**	85.40	**82.23**

Comparison with Baselines. Table 3 provides a comparison between our proposed model and previous work on the DiaKG dataset. The span-based method outperformed the sequence labeling-based method. Our CPMFA model achieved the highest F1 score and recall at 83.79% and 82.23%, respectively. However, the model's precision was suboptimal, at 85.40%, which was 1.58% lower than that of the Efficient Global Pointer model. The analysis indicated that the pointer-based architecture of the Efficient Global Pointer model had a higher accuracy in identifying longer entities in input sequences. It successfully identified spans by

learning patterns of head and tail word pairs, even if the intermediate words had changed. In contrast, our CPMFA model comprehensively recognized entities by classifying the relation between each character pair.

Table 3. Performances of our model and baseline models on DiaKG dataset. The bold value indicates the optimal results.

	Method	F1 (%)	P (%)	R (%)
Sequence labeling-based	Cascade-CRF [21]	62.58	59.52	65.97
Span-based	W2NER [8]	80.51	81.52	79.52
	Global Pointer [22]	69.35	73.36	65.76
	Efficient Global Pointer [23]	82.95	**86.98**	79.28
Ours	CPMFA	**83.79**	85.40	**82.23**

Ablation Experiments. We selected the DiaKG dataset to evaluate the effectiveness of our model's components. We conducted an analysis by removing one component at a time to observe the impact on performance. Table 4 shows the performance of the model's variations with "w/o" representing "without." The results signify that all model components are essential for optimal performance.

First, we replaced the LERT pre-trained language model with the BERT pre-trained language model in our CPMFA model, causing a 2.26% decrease in the F1 score. This indicates that pre-trained language model with rich linguistic features can significantly tackle the challenge of linguistic diversity phenomena in Chinese text.

Then, we removed the multi-feature representation from our model CPMFA. This resulted in a 3.18% decrease in the F1 score, indicating that the multi-feature representation module can perform deeper information mining as a complement to pre-trained language models.

Lastly, we evaluated the effectiveness of using the PSA module by removing it from the experiment, resulting in a 1.18% reduction in F1-score. The result indicates that the PSA module can help to focus on the most essential aspects of the input features.

Table 4. Ablation study on DiaKG dataset. The bold value indicates the optimal results.

	F1 (%)	P (%)	R (%)
CPMFA (ours)	**83.79**	**85.40**	**82.23**
w/o LERT	81.52	84.38	78.85
w/o multi-feature representation	80.61	82.99	78.36
w/o PSA	82.61	84.03	81.23

5 Conclusion

This paper proposes a CPMFA model for CNNER and evaluate its performance in medical nested text related to diabetes. First, our model utilizes a pre-trained language model LERT and BiLSTM to acquire high-quality character embeddings that effectively tackle the challenge of linguistic diversity phenomena in Chinese text. Second, the model integrates multi-feature representation and the PSA module to capture critical features for effective text mining. To address the issue of imbalanced entity types, the PolyLoss-based function is employed during training. Ablation Experiments validate the effectiveness of each model component. Notably, the CPMFA model outperforms existing NNER models in terms of F1 score, demonstrating its potential to enhance CNNER performance for Chinese medical text and offer a novel technical solution in other domains.

Acknowledgement. This study was supported by the Key Project of Regional Innovation and Development Joint Fund of National Natural Science Foundation of China (Grant No. U22A2025).

References

1. Rujia, Z., Lu, D., Peng, G., Bang, W.: Chinese nested named entity recognition algorithm based on segmentation attention and boundary-aware. Comput. Sci. **50**(01), 213–220 (2023)
2. Shiyuan, Y., Shuming, G., Ruiyang, H., Jianpeng, Z., Nan, H.: Layered regional exhaustive model for Chinese nested named entity recognition. Comput. Technol. Dev. **32**(09), 161–166+179 (2022)
3. Li, H., Xu, H., Qian, L., Zhou, G.: Multi-layer joint learning of Chinese nested named entity recognition based on self-attention mechanism. In: Natural Language Processing and Chinese Computing: 9th CCF International Conference, NLPCC 2020, Zhengzhou, China, 14–18 October 2020, Proceedings, Part II, pp. 144–155. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60457-8_12
4. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
5. Straková, J., Straka, M., Hajič, J.: Neural architectures for nested NER through linearization. arXiv preprint arXiv:1908.06926 (2019)
6. Li, F., Lin, Z., Zhang, M., Ji, D.: A span-based model for joint overlapped and discontinuous named entity recognition. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 4814–4828 (2021)
7. Xia, C., et al.: Multi-grained named entity recognition. In: 57th Annual Meeting of the Association for Computational Linguistics, ACL 2019, pp. 1430–1440. Association for Computational Linguistics (ACL) (2020)
8. Li, J., et al.: Unified named entity recognition as word-word relation classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 10965–10973 (2022)

9. Islam, T., Zinat, S.M., Sukhi, S., Mridha, M.F.: A comprehensive study on attention-based NER. In: Khanna, A., Gupta, D., Bhattacharyya, S., Hassanien, A.E., Anand, S., Jaiswal, A. (eds.) International Conference on Innovative Computing and Communications. AISC, vol. 1388, pp. 665–681. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-2597-8_57
10. Cui, S., Joe, I.: A multi-head adjacent attention-based pyramid layered model for nested named entity recognition. Neural Comput. Appl. **35**(3), 2561–2574 (2023)
11. Rodríguez, A.J.C., Castro, D.C., García, S.H.: Noun-based attention mechanism for fine-grained named entity recognition. Expert Syst. Appl. **193**, 116406 (2022)
12. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
13. Li, D., Yan, L., Yang, J., Ma, Z.: Dependency syntax guided BERT-BiLSTM-GAM-CRF for Chinese NER. Expert Syst. Appl. **196**, 116682 (2022)
14. Yu, Y., et al.: Chinese mineral named entity recognition based on BERT model. Expert Syst. Appl. **206**, 117727 (2022)
15. Xu, Y., Huang, H., Feng, C., Hu, Y.: A supervised multi-head self-attention network for nested named entity recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 14185–14193 (2021)
16. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
17. Leng, Z., et al.: PolyLoss: a polynomial expansion perspective of classification loss functions. arXiv preprint arXiv:2204.12511 (2022)
18. Zhang, H., Zu, K., Lu, J., Zou, Y., Meng, D.: EPSANet: an efficient pyramid squeeze attention block on convolutional neural network. In: Wang, L., Gall, J., Chin, T.J., Sato, I., Chellappa, R. (eds.) Proceedings of the Asian Conference on Computer Vision, pp. 1161–1177. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-26313-2_33
19. Chang, D., et al.: DiaKG: an annotated diabetes dataset for medical knowledge graph construction. In: Qin, B., Jin, Z., Wang, H., Pan, J., Liu, Y., An, B. (eds.) Knowledge Graph and Semantic Computing: Knowledge Graph Empowers New Infrastructure Construction: 6th China Conference, CCKS 2021, Guangzhou, China, 4–7 November 2021, Proceedings, vol. 1466, pp. 308–314. Springer, Cham (2021). https://doi.org/10.1007/978-981-16-6471-7_26
20. Li, J., Sun, A., Han, J., Li, C.: A survey on deep learning for named entity recognition. IEEE Trans. Knowl. Data Eng. **34**(1), 50–70 (2020)
21. Wei, Z., Su, J., Wang, Y., Tian, Y., Chang, Y.: A novel cascade binary tagging framework for relational triple extraction. arXiv preprint arXiv:1909.03227 (2019)
22. Su, J., et al.: Global pointer: novel efficient span-based approach for named entity recognition. arXiv preprint arXiv:2208.03054 (2022)
23. Su, J.: Efficient globalpointer: fewer parameters, more effects, January 2022. https://spaces.ac.cn/archives/8877