Journal of Network Intelligence ©2024 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **9**, Number **3**, August **2024** pp. **000–000**

Chinese Relation Extraction Based on Cross-Attention and Multi-feature Perception

Shiao Xu

School of Electronics and Information Engineering
Jingchu University of Technology, Jingmen 448000, China
946831760@qq.com

Shuihua Sun, Zhiyuan Zhang

School of Computer Science and Mathematics
Fujian University of Technology, Fuzhou 350118, China
ShuiHua.11109029@gmail.com, 1107416742@qq.com

Huan Zhou*

School of Electronics and Information Engineering
Jingchu University of Technology, Jingmen 448000, China
83208132440@qq.com

*Corresponding author: Huan Zhou

Received July 21, 2023, revised November 16, 2023, accepted January 18, 2024.

Abstract. *Relation extraction aims at extracting semantic relations between given pairs of entities from unstructured textual data and is a critical task in information extraction. A relation extraction model based on cross-attention and multi-feature perception is proposed to address the limitations of Chinese relation extraction models that rely solely on characters, which struggle to capture rich semantic information, and the inability to fully utilize the textual structural information during neural network feature extraction. First, the model uses BERT to encode sentences and entities to obtain character vector sequences and word vector sequences of sentences, and uses word cross-attention mechanism to capture the key word and word information in sentences to form character and word cross-fused sentence sequences. Second, uses graph convolutional network and deep separable convolutional network to capture syntactic information between nodes and multi-granularity local semantic information in sentence sequences, respectively, and forms sentence features for classification by attention mechanism. Finally, combines with Softmax classifier for relation classification. Experimental results on the Chinese San-Wen dataset demonstrate the superiority of the proposed method over mainstream neural network-based relation extraction methods, with an F1-score of 71.57%.*
Keywords: relation extraction; graph convolutional network; cross-attention mechanism; depthwise separable convolution

1. **Introduction.** In the era of exponential data volume and digitalization growth, the efficient and rapid extraction of valuable information from massive text data has become important research direction both domestically and internationally. Relation Extraction (RE), as a crucial subtask in the field of information extraction, aims to extract semantic relationships between given entity pairs from texts, providing technological support for tasks such as knowledge graph construction [1], question and answer systems [2], and automatic summarization [3].

Traditional RE methods mainly consist of feature-based approaches and traditional machine learning methods. However, these two categories of methods need to be revised to improve accuracy and limitations imposed by human expertise. In contrast, neural networks can automatically uncover implicit features within sentences, eliminating the need for complex feature engineering. As a result, neural networks have shown significant superiority over traditional methods regarding relation extraction performance, making neural network models mainstream in current relation extraction tasks [4, 5, 6]. Existing Chinese relation extraction methods predominantly fall into two categories: those based on character features and those based on word features. The word-based RE model requires word separation of sentences and converts the processed lexical information into a sequence of sentences with corresponding word features for relation extraction. This approach focuses on obtaining accurate text representations from word-level features, but the dependence on the word separation tool and the quality of the separation seriously affects the performance of the classification model [7]. The word character-based RE model converts word information in a sentence into a corresponding sequence of word feature sentences for relation extraction, which can effectively reduce the impact of word separation errors on the classification model and outperforms the word feature model, but the semantic meaning of individual Chinese characters in Chinese is more ambiguous. For example, the meaning of the Chinese character "处" in the phrase "处理 (deal with)" and "处所 (premises)" is completely different [8]. Therefore, to extract semantic relations between entities more accurately from entity-level contextual semantic information, this paper comprehensively considers different granularity information and constructs a RE model that combines the crossing fusion of characters and words to express the semantic information of sentences fully.

On the other hand, although neural networks have achieved good results in RE tasks, most existing Chinese RE models still need to fully utilize the syntactic structures and dependencies in sentences [9]. In some texts with more complex syntactic relations, extracting the relations between entity pairs requires understanding and inferring the global semantic information. The dependency tree can obtain the dependency relations among the components in the sentence, which helps the classification model understand the global semantic information of the sentence. Considering that the dependency analysis tree can process sentences into graph structures, it is more advantageous to use Graph Convolutional Network (GCN) to process the dependency analysis tree.

To address the aforementioned issues, this paper proposes a Chinese RE model based on Cross Attention and Multi-feature Perception (CAMFP). The model consists of four parts: semantic feature input network, character and word cross-fusion network, feature perception network, and prediction output network. The semantic feature input network utilizes pre-trained BERT [10] to encode the characters in the sentence, generating dynamic character feature vectors. Simultaneously, the dynamic character vectors are processed based on word segmentation results to form word feature vectors for the sentence. The character and word cross-fusion network employs a cross-attention mechanism to process word vectors and word vectors to capture the focused word and word information in sentences, forming word cross-fused sentence sequences. The feature perception network uses GCN and deep separable convolutional networks to capture syntactic features between words in a sentence sequence and multi-grain local features, respectively, and uses an attention mechanism to convert these two features into a sentence vector for relational classification. The prediction output network stitches all sentence vectors and then maps them to the classification space using fully connected layer, combining them with a softmax classifier for classification. The main contributions of this paper are as follows:

(1) We propose a cross-attention mechanism that fuses character and word vectors, capturing and integrating important character and word information to enhance the semantic representation of the sentence, taking into account the structural characteristics of Chinese texts.

(2) To fully utilize the semantic and structural information of the sentence, we utilize uses dependency trees to convert sentences into dependency graphs and uses GCN to capture syntactic information between words to compensate for the inadequate semantic information of words.

(3) Experimental results on the Chinese SanWen relational extraction dataset show that the proposed method outperforms the existing mainstream models with the F1-score of 71.57%.

2. **Related work.** RE methods mainly include traditional machine learning methods [11, 12] and neural network-based methods [13, 14, 15]. Traditional machine learning methods rely heavily on manually constructing feature vectors or designing different kernel functions for feature selection. These methods excessively rely on expert knowledge and are limited by the constraints of human experience. In contrast, neural networks have the ability to automatically capture latent features in sentences, alleviating the time-consuming and laborious process of manually designing features in traditional methods.

In recent years, deep learning has made breakthroughs in areas such as image processing [16], speech recognition, and intelligent transportation systems [17, 18].scholars have started exploring the use of neural networks in relation extraction tasks, aiming to improve their performance by constructing different neural network models. Since the Chinese relation extraction dataset is small and slow to start, the English relation extraction technique is more mature compared with the Chinese. Li et al. [19] used RNN to capture the semantic features of entity context for RE. This approach resulted in substantial enhancements in classification accuracy when compared to rule-based methods. Zhou et al. [20] introduced an attention mechanism based on BiLSTM for RE. This method utilizes BiLSTM to encode semantic information of entity context and calculates the contribution weight of each word using the attention mechanism, obtaining a sentence vector for classification. Li et al. [21] proposed a dual attention GCN model, which captures rich contextual dependency relation based on attention mechanism to address RE, aiming to learn the dependency relationships between nodes better and adaptively integrate local features with global dependency relation.

Neural network-based Chinese relation extraction techniques mainly include word-based and character-based approaches. Zhang et al. [7] proposed a method based on the word-level multi-hop attention mechanism for extracting Chinese medical relations. This model generates multiple weight vectors for the sentence at each attention step, generating different semantic representations of sentences to address the problem of representing contextual information with a single sentence vector. Zhao et al. [22] proposed a CNN network model of the polysemy rethinking mechanism, which uses a thesaurus to integrate word-level information, correct errors caused by word separation, and add polysemy information to the model to alleviate the problem of multiple meanings of the word. Zhang et al. [23] proposed a character-based multi-feature fusion model that uses CNN to capture local features of word vectors to construct word vectors of sentences, combines attention-based BiLSTM to encode word vectors and word vectors separately to generate sentence vectors for classification and fuses entity-aware features to reduce the ambiguity problem of multiple meanings of the word entity, and experimental results on the SanWen dataset demonstrate the superior performance of this approach. Li et al. [24] proposed a Chinese RE framework based on multi-granularity grid networks, which dynamically

integrates word information into character-based methods based on grid networks and combines external databases for relation extraction, mitigating the impact of incorrect word segmentation and one-word multiple meanings on the performance of RE models. The F1-score on Chinese datasets are superior to other models.

3. **Methodology.** This article constructs a RE model based on cross attention and multi-feature perception, and the model structure is shown in Figure 1.

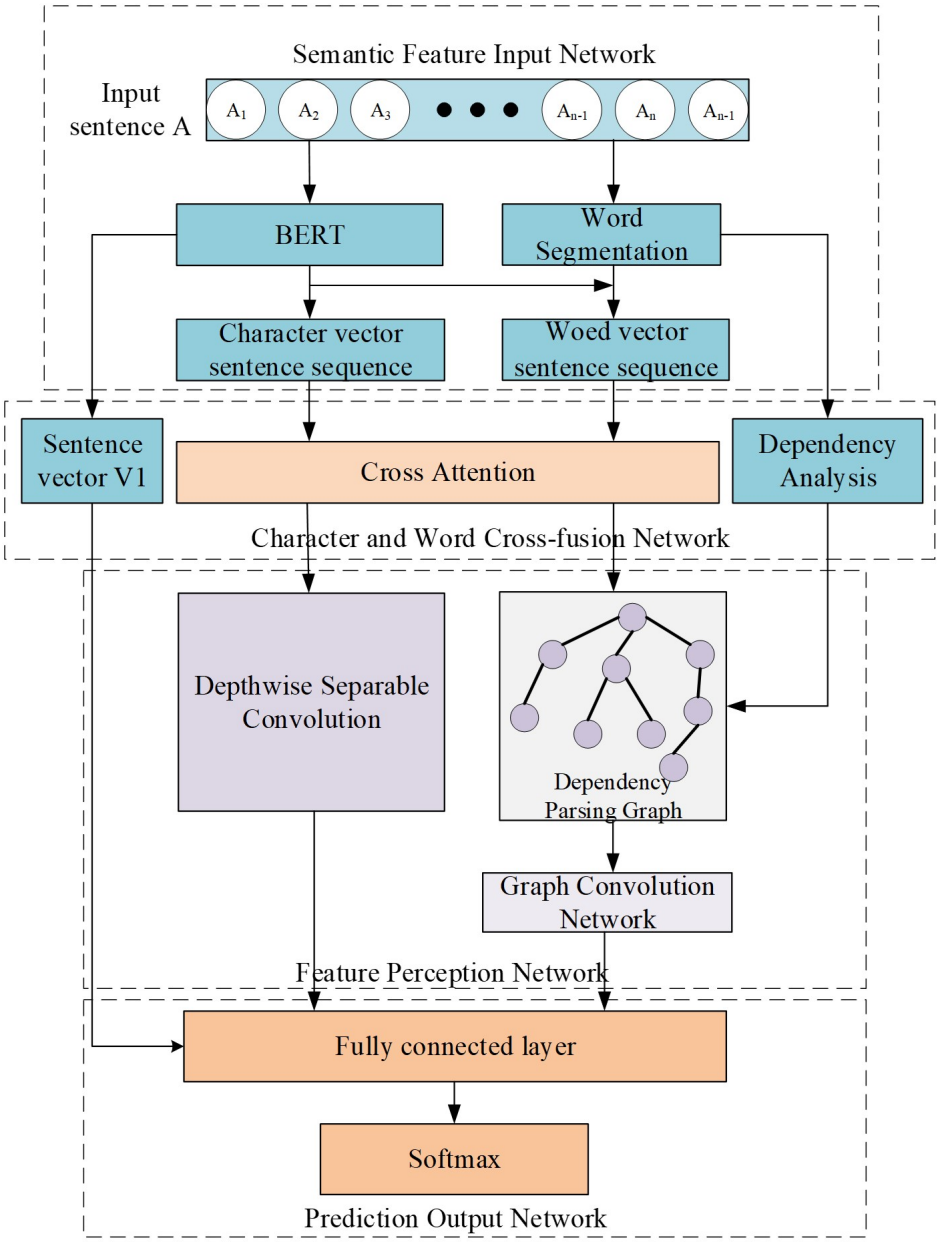

FIGURE 1. The overall framework of the CAMFP model

3.1. **Semantic feature input network.** To solve the problem that static word vectors such as Word2vec and Glove cannot be adjusted according to contextual semantic information, this paper adopts BERT [10] to encode each word in Chinese text to obtain dynamic word vectors. The BERT model typically requires the addition of a "[CLS]" token at the beginning of a sentence, which is used to output the information of the entire

sentence. Additionally, the BERT model uses the "[SEP]" token as a sentence-ending delimiter.

As relation extraction involves identifying the relation category between two entities, the model must capture the information of the two entities in the sentence. Therefore, this study appends entity information at the end of the sentence and uses the "[SEP]" token to separate it, aiming to enhance the expression of entity information within the sentence. For example, the sentence "幽兰在山谷，本自无人识" . ("The orchid is in the valley, it is unknown" .) The two target entities are "orchid" and "valley" , and the modified sentence A with added tags and entity information becomes "[CLS] 幽兰在山谷，本自无人识。[SEP] 幽兰 [SEP] 山谷 [SEP]" .("[CLS] The orchid is in the valley, it is unknown [SEP] orchid [SEP] valley [SEP]" .)

The BERT model is used to encode sentence A to generate a character vector sequence $C = \{c_1, c_2, ..., c_n\} \in R^{n \times d_c}$ of Chinese text, where n is the number of character in the sentence and denotes the character vector dimension of the BERT output. Since "[CLS]" in BERT represents the overall information of the sentence, in order to obtain the sentence vector for classification, the sentence vector $V_1 \in R^{d_c}$ is formed by encoding the "[CLS]" corresponding to the hair vector c_1 through the tanh function and the fully connected layer, as shown in Equation (1):

$$V_1 = W_1(tanh(c_1)) + b_1 \tag{1}$$

where $W_1 \in R^{d_c \times d_c}$,$b_1 \in R^{d_c}$ denote the weight and bias of the fully connected layer, respectively.

Considering the problem of semantic ambiguity in character vectors, a word vector is constructed based on the BERT generated to character vectors. In this paper, the Language Technology Platform (LTP) is used to partition sentence A to obtain a sentence $W = \{w_1, w_2, ..., w_i, ..., w_m\}$ containing m words, where $w_i = \{c_k, c_k + 1, ..., c_{k+h}\}$ denotes the word vector of the ith word in the sentence. The word vector sequence $X = \{x_1, x_2, ..., x_i, ..., x_m\} \in R^{m \times d_c}$ of the sentence is constructed using the averaging method based on the word vectors generated by BERT according to the results after the word separation, and the word vectors are calculated as shown in Equation (2):

$$x_i = \frac{1}{(k+h) - k + 1} \sum_{t=k}^{k+h} c_t \tag{2}$$

In addition, to ensure that the length of the word vector is the same as that of the word vector after word separation, this paper uses zero vectors to fill the word vector sequence to form the complete word vector sequence $X = \{x_1, x_2, ..., x_i, ..., x_m, ..., x_n\} \in R^{n \times d_c}$,where x_m to x_n is the filled zero vectors.

3.2. **Character and word cross-fusion network.** To accurately express sentence context and semantic information by combining features of different granularities, this article presents a character and word cross-fusion network based on obtaining dynamic word vectors and word vectors. The network structure is illustrated in Figure 2. The network constructs Cross Attention (CA) mechanism based on Self-attention mechanism in Transformer [25], which differs from Self-attention in that CA focuses on both word information across granularity and enhances the semantic expression of sentences using the focused word information and word information.

The character and word cross-fusion network takes the dynamic character vector sequence C and word vector sequence X as the input of this network, and captures the focused word information in the character vector sequence and word vector sequence using the CA mechanism and fuses them to form a sentence vector containing multi-granularity

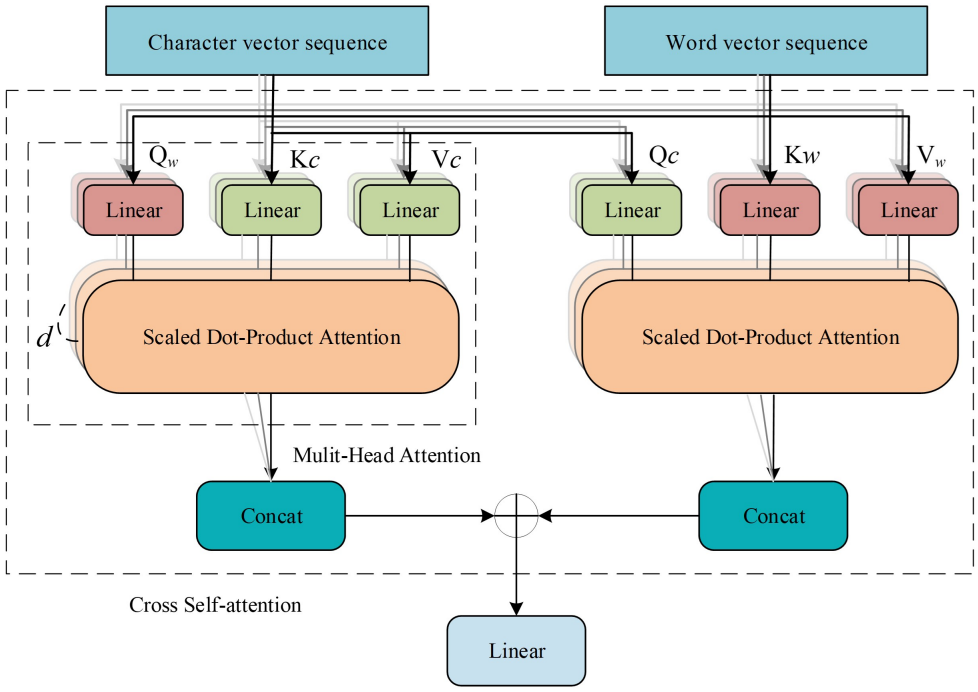

FIGURE 2. Structure diagram of character and word cross-fusion network

information. The CA mechanism in the character and word cross-fusion network contains the word Self-attention mechanism with character vector as query matrix and character Self-attention mechanism with word vector as query matrix. The word Self-attention and word Self-attention are calculated as shown in Equation (3)-Equation (4):

$$Word - Attention(Q_c, K_w, V_w) = softmax\ (\frac{Q_c K_w^T}{\sqrt{d_c}})V_w \tag{3}$$

$$Character - Attention(Q_w, K_c, V_c) = softmax\ (\frac{Q_w K_c^T}{\sqrt{d_c}})V_c \tag{4}$$

where Q_c,K_c,V_c represent the query, key, and value matrix of character vectors, respectively, Q_w,K_w,V_w denote the query, key, and value matrix of the word vector, respectively.

In order to capture the weight information of different subspaces in the word self-attention and character self-attention, multi-head attention maps the weights to parallel subspaces for learning. The weight information learned within the subspaces is then concatenated to form the sentence representation $S \in R^{n \times d_c}$ for word and character fusion, as shown in Equation (5)-Equation (9):

$$head_i^w = Word - Attention\left(XW_i^{c,Q}, CW_i^{w,K}, CW_i^{w,V}\right) \tag{5}$$

$$head_i^c = Character - Attention\left(CW_i^{w,Q}, XW_i^{c,K}, XW_i^{c,V}\right) \tag{6}$$

$$MultiHead_w = [head_1^w \oplus ... \oplus head_d^w] \tag{7}$$

$$MultiHead_c = [head_1^c \oplus ... \oplus head_d^c] \tag{8}$$

$$S = W_s(MultiHead_w \oplus MultiHead_c) + b_s \tag{9}$$

where $W_i^{w,Q}$, $W_i^{w,K}$, $W_i^{w,V} \in R^{d_c \times d_c/d}$ are trainable word vector mapping weight matrices, and $W_i^{c,Q}$, $W_i^{c,K}$, $W_i^{c,V} \in R^{d_c \times d_c/d}$ are trainable word vector mapping weight matrices. $W_s \in R^{2d_c \times d_c}$ and $b_s \in R^{d_c}$ are weight matrices and bias vectors of fully connected layers. d denotes the number of heads of multi-headed attention, and $\oplus$ is the cascade operation.

3.3. **Feature perception network.** The feature perception network is designed to extract essential semantic features for classification by fusing word information into sentence vectors. The network consists of a structural feature perception network and a local feature perception network. The structural feature perception network uses dependency parsing trees to process text into graph structures, and GCN is used to capture the syntactic structural features between graph nodes. The local feature perception network extracts N-gram features of sentence vectors using different CNNs to achieve the multi-granularity local perception of sentence vectors.

3.3.1. *Structural feature perception network.* When capturing the relation between two entities, a classification model needs to comprehend and infer the global information of the sentence. Dependency parsing trees can provide the dependency relationships between different constituents in the sentence, effectively aiding the classification model in understanding the overall semantic information of the sentence. Therefore, this paper proposes using a structural feature perception network to capture the dependency relationships between nodes in the textual structure graph, thereby enhancing the expression of semantic information in the sentence. The network structure is illustrated in Figure 3.

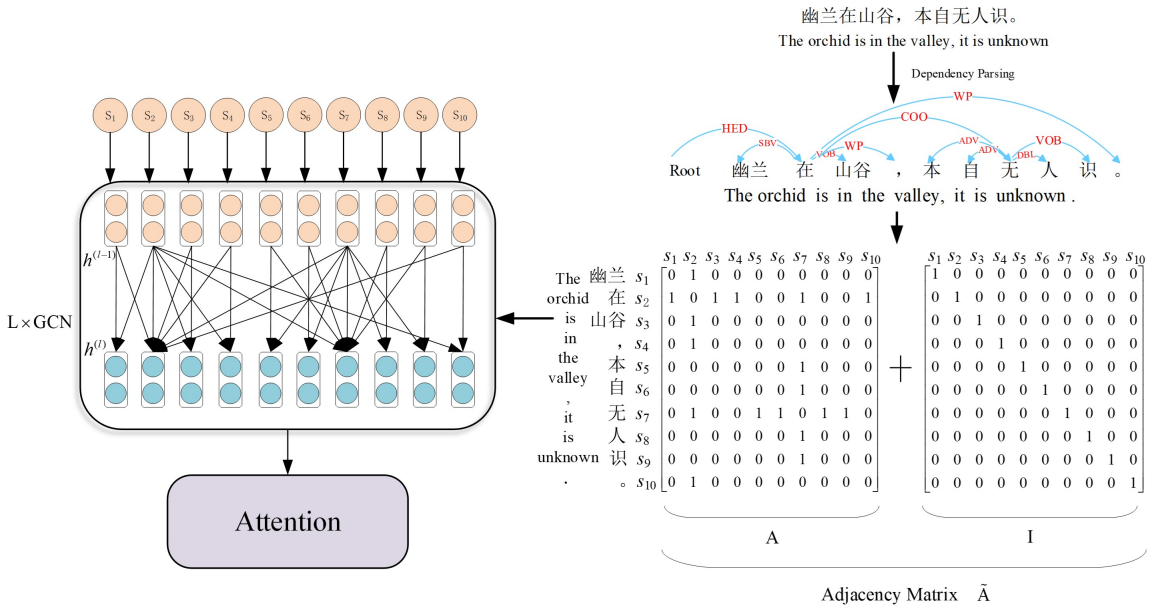

FIGURE 3. Structural feature perception network

The network uses LTP to analyze the sentences to construct the dependency parsing tree, using the words in the sentences as the nodes of the graph and the dependency relations between words as the edges of the graph to construct the dependency graph. The construction of the dependency graph does not consider the direction and type of the dependency relationship. With the help of the method in reference [26], the adjacency matrix A is used to store the dependency graph. For example, if there are n nodes in the dependency graph, then the $n \times n$ adjacency matrix A can be constructed. If there is a dependency between node i and node j in the dependency graph, then $A_{ij} = A_{ji} = 1$ in the adjacency matrix A, otherwise $A_{ji} = 0$. In addition, to prevent ignoring the nodes own information transfer in the process of graph convolution, a self-loop I is added to each node in the dependency graph, and the adjacency matrix is $\tilde{A} = A + I$, where I is the unit matrix of $n \times n$. After generating the adjacency matrix $\tilde{A}$ of the sentence, given

the input set $S = \{s_1, ..., s_i, ..., s_n\}$ of the GCN, the semantic relationships between the nodes are captured using the GCN to form a sentence feature representation containing syntactic structural features. In the l layer GCN, the output vector of node i at layer l is h_i^l is represented by node i at the previous layer and its neighboring nodes, as shown in Equation (10):

$$h_i^l = \sigma\left(\sum_{j=1}^{n} \tilde{A}_{ij} W^l h_j^{l-1} / d_i + b^l\right) \tag{10}$$

where σ denotes the Relu function, $d_i = \sum_{j=1}^{n} \tilde{A}_{ij}$ is the degree of the ith node, and W^l and b^l denote the trainable weight and bias of the lth layer GCN, respectively.

After each node in the character and word cross-fused sentence sequences S goes through the L-layer GCN, the syntactic structure feature representation $H^{(L)} = \left\{h_1^L, ..., h_i^L, ..., h_n^L\right\} \in R^{n \times d}$ of the sentence is obtained, and the syntactic structure feature is transformed into the sentence vector $H_1 \in R^{d_c}$ for classification by the f function, as shown in Equation (11):

$$H_1 = f(H^{(L)}) = f(GCN^{(L)}(H^{(0)})) \tag{11}$$

where $H^{(0)} \in R^{n \times d_c}$ denotes the input $H^{(0)} = S$ of the first layer GCN. The $f: R^{n \times d_c} \rightarrow R^{d_c}$ function is a sentence-level attention mechanism for converting the captured syntactic structural features into a sentence vector H_1 for classification, as computed in Equation (12)-Equation (14):

$$Q_i = tanh(H_i^{(L)}) \tag{12}$$

$$\alpha_i = \frac{exp(W_q^T Q_i)}{\sum_{j=1}^{n} exp(W_q^T Q_i)} \tag{13}$$

$$H_1 = \sum_{i=1}^{n} \alpha_i Q_i \tag{14}$$

where $H_i^{(L)}$ denotes the output vector of the ith node in the L-layer GCN and W_q^T is the trainable weight matrix in the attention mechanism of the sentence layer.

3.3.2. *Local feature perception network.* Since important information regarding some relationship instances usually occurs in local regions of the sentence sequence, relying solely on syntactic structural features of the sentence may result in the loss of critical semantic information. This paper constructs a local feature perception network to enable the classification model to capture local features of different granularity, as illustrated in Figure 4. The model uses Depthwise Separable Convolution (DSC) network in combination with the approach of literature [27] to capture different granularity features of sentences and enhance the classification model's ability to perceive local features of sentences.

Depthwise Convolution uses a filter to extract a channel feature to capture the sentence information. Pointwise Convolution uses a filter of size 1×1 to map the channels of the feature vector to a new channel space to form the local features of the sentence. This paper uses Depthwise Convolution with filter sizes $1 \times d_c$,$3 \times d_c$, $5 \times d_c$ to capture multi-granularity local features of the sentences. In order to ensure that the length of the convolved sentences remains consistent with the input sentence length, a zero-padding strategy is applied to pad the input sentences. Taking the character and word cross-fused sentence sequences S of the word cross-fusion network as input, the local features $T^k \in R^{n \times d_w}$ of the character and word cross-fused sentence sequences S are captured

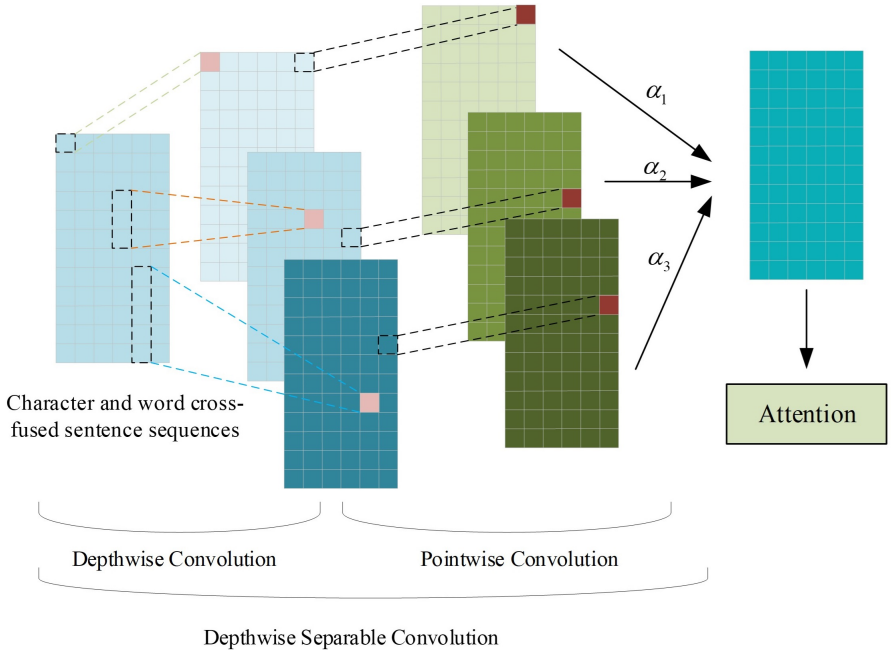

FIGURE 4. Local feature perception Network

using Depthwise Convolution with filter size $k \times d_c$ and Pointwise Convolution with filter size 1×1, as shown in Equation (15)-Equation (16):

$$o_{i,c}^{k} = DepthwiseConv_k(S) = \sum_{j=1}^{k} \left(W_{j,c} S_{i+j-\frac{k+1}{2},c} \right) \tag{15}$$

$$T^k = Pointwiseconv(W_t, O^k) \tag{16}$$

where $O^k \in R^{n \times d_w}$ is the feature vector captured using Depthwise Convolution, $o_{i,c}^{k} \in R$ denotes the feature value of the i-th word c-th channel captured using Depthwise Convolution, $W_{j,c}$ is the trainable weight matrix, k denotes the width of the convolution kernel, d is the size of the hidden layer, and W_t is the parameter matrix of Pointwise Convolution.

Since different local features of the same sentence contribute differently to the relation extraction task, in order to highlight the importance of each granularity local feature, this paper introduces a kind of gating mechanism to dynamically select the weights of different local information. The local features of different granularities are aggregated by the weight values and combined with the f function in Section 3.3.1 to form the sentence feature $H_2 \in R^{d_w}$, as shown in Equation (17):

$$H_2 = f\left(\sum_{i=1}^{k} \frac{exp(\alpha_i)}{\sum_{j=1}^{k} exp(\alpha_j)} T^k \right) \tag{17}$$

where $alpha \in R^k$ s a trainable vector with all initial values of 1 and k takes the values of 1, 3, and 5.

3.4. **Prediction output network.** This paper first splices the sentence vector V_1 generated by the semantic feature input network with the feature vectors H_1 and H_2 captured by the feature-aware network. And the fully connected network is used to map the spliced

features to $|y|$ categorical label spaces to obtain the prediction information $P \in R^{|y|}$, as shown in Equation (18):

$$P = W_p(V_1 \oplus H_1 \oplus H_2) + b_p \tag{18}$$

where $W_p \in R^{|y| \times (2d_c + d_w)}$ and $b_p \in R^{|y|}$ denote the weight matrix and bias vector of the fully connected network, respectively. After generating the prediction information P, the relationship prediction is combined with the Softmax function to generate the probability distribution $\hat{P}(y|A)$ for each relation category, as shown in Equation (19):

$$\hat{P}(y|A, \theta) = Softmax(P) \tag{19}$$

where y denotes the class of the target relation. A is the input sentence and θ denotes the learnable parameters in the network.

During the model training process, to alleviate the overfitting problem, L2 regularization is applied to penalize the network parameters and improve the generalization performance of the network, as shown in theEquation (20):

$$L = -\frac{1}{|B|} \sum_{i=1}^{|B|} y_i log(\hat{P}(y|A, \theta)) + \lambda \|\theta\|_2^2 \tag{20}$$

4. Experiments.

4.1. Datasets and evaluation metrics.

4.1.1. *Dataset.* In order to validate the effectiveness of the proposed model, this study utilizes the Chinese SanWen dataset, constructed based on reference [28], for model training and testing. This dataset consists of 837 fully annotated Chinese literary works, and after preprocessing, it encompasses 21,240 sentences. Among these sentences, 17,227 are allocated for training, 2,220 for testing, and 1,793 for validation. The Chinese SanWen dataset comprises ten relationship categories, and the distribution of training and testing data for each category is illustrated in Figure 5. The category "Other" is designated to represent relationships that do not belong to any of the other nine categories.

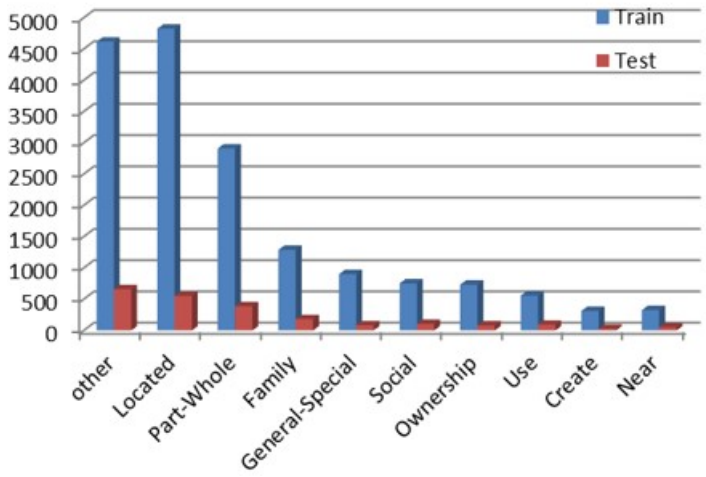

FIGURE 5. Distribution of Chinese SanWen dataset

Table 1. The hyper-parameter settings of model

Hyper-parameters	Description	Value
d_c	BERT model character vector dimension	768
d	Number of head of crossed self-attention	4
k	Filter size	1,3,5
d_w	Number of CNN convolutional kernel channels	768
lr	Learning rate	$5e^{-5}$
$\|B\|$	Batch Size for Training	16
λ	L2 Regularization Coefficient	$5e^{-4}$
L	Number of GCN layers	2
Dr	dropout ratio	0.3

4.1.2. *Evaluation metrics.* The current RE task mainly uses precision (P) and recall (R) as evaluation metrics for relational extraction models, and precision and recall are metrics with contradictory relations. Therefore, the introduction of the harmonic-mean F1-scores of P and R can provide a comprehensive consideration of the two. In the multi-classification problem, the harmonic mean F1-scores of P and R are calculated as shown in Equation (21):

$$F1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2 \times P_i \times R_i}{P_i + R_i} \tag{21}$$

where n is the number of relation categories.

4.2. **Hyper-parameter settings.** This paper uses the pre-trained model BERT word vector as the model input, whereas the BERT uses the BERT-base-chinese model. The main hyperparameters of the relational extraction model in this paper are set as shown in Table 1.

4.3. **Experimental results.** To validate the relationship extraction performance of the CAMFP model, this study conducted experiments on the Chinese SanWen dataset and compared it with seven mainstream models. The results of the comparison are shown in Table 2. The experiment's compared models mainly include CNN-based, RNN-based, and BERT-based models, with the experimental comparison data sourced from the literature. The CNN-based models include CR-CNN, DepCNN, and BRCNN. The RNN-based models include SDP-LSTM, Att-BLSTM+C-Att-BLSTM, and MG Lattice. The BERT-based model is LAN. The experimental results of the CR-CNN, DepCNN, BRCNN, and SDP-LSTM models are all sourced from reference [27].

Table 2 shows that the CAMFP model proposed in this study performs best on the Chinese SanWen dataset, with an F1-score of 71.57%. Compared to the best-performing models in each category, the BRCNN, MG Lattice, and LAN, the F1-score of CAMFP has been improved by 15.97%, 5.96%, and 1.72%, respectively. The main reason for this improvement is that the CAMFP model utilizes BERT to encode both the character-level and word-level information of the sentences, enhancing the contextual information and semantic expression of the sentences through the fusion network of characters and words. Additionally, the feature-aware network is used to perceive the sentences local features and grammatical structure features. It combines them to comprehensively express the

Table 2. Comparison of CAMFP with existing methods for Chinese SanWen dataset

Model Class	Model	F1-score
CNN-based Models	CR-CNN[4]	54.10%
	DepCNN[29]	55.20%
	BRCNN[30]	**55.60%**
RNN-based Models	SDP–LSTM[5]	55.30%
	Att-BLSTM+C-Att-BLSTM[23]	56.20%
	MG Lattice[24]	**65.61%**
BERT-based Models	LAN[31]	**69.85%**
	CAMFP(Our)	**71.57%**

Table 3. Effects of different input features on CAMFP performance

Input feature	Model	P	R	F1-score
Word vectors	CAMFP	67.11%	71.33%	69.15%
Character and word vectors		70.97%	72.18%	**71.57%**

important semantic features of the sentences, which enhances the accuracy of relation extraction.

4.4. **Analysis and discussion of experimental result.**

4.4.1. *Effectiveness analysis of different input features.* In the semantic feature input network, this study utilizes BERT encoding to generate character vectors and construct word vectors specific to this study. Both the character vectors and word vectors are simultaneously used as inputs to the model, aiming to address the issue of insufficient semantic information when using character or word features alone. In order to validate the effectiveness of this approach, two sets of comparative experiments were designed and conducted on the Chinese SanWen dataset. The first set of comparative experiments compares the input of word features and the input of character and word features in the CAMFP model. Considering that the GCN module in this study's model captures the syntactic feature information of the segmented sentences, which cannot verify the impact of character vectors as inputs on the model's performance, the second set of experiments was performed by removing the GCN network from the CAMFP model, aiming to explore the influences of character features, word features, and the combination of character and word features on the model. The experimental results are shown in Table 3 and Table 4. Since the character and word cross-attention networks in CAMFP are designed for both word vectors as model inputs in the above two sets of experiments. When the word or character features are used as model inputs, the character and word cross-attention networks are replaced with self-attention networks for experiments in this paper.

Table 3 shows that when only word features were used as inputs, the F1-score of the CAMFP model was 69.15%. When character and word features were used as inputs, the F1 score of the CAMFP model improved by 2.42%. From Table 4, when character features were used as inputs alone, the model's performance was better than using word features alone. This can be attributed to word segmentation errors affecting the model's

Table 4. Effects of different input features on CAMFP (without GCN) performance

Input feature	Model	P	R	F1-score
Character vectors	CAMFP (without GCN)	69.29%	68.80%	69.04%
Word vectors		67.91%	69.72%	68.80%
Character and word vectors		67.43%	72.33%	**69.79%**

Table 5. The effect of cross-attention network on model performance

Model	P	R	F1-score
CAMFP(none CA)	65.82%	69.40%	67.56%
SAMFP	66.85%	72.72%	68.80%
CAMFP	70.97%	72.18%	**71.57%**

understanding of contextual information in the sentences. The performance was optimal when character and word features were used as inputs to the model. The F1-score improved by 0.75% and 0.99% compared to using character and word features alone. It shows that the different semantic information contained in character features and word features can complement each other and enrich the contextual semantic information of sentences. It also highlights the significance of combining character and word features for understanding the contextual semantics of sentences.

4.4.2. *Effectiveness analysis of character and word cross-attention networks.* Section 3.2 introduces the proposed character-word cross-fusion network structure, which utilizes the character-word cross-fusion network to aggregate character and word information to enhance the semantic expression capability of sentences. To investigate whether the character-word cross-attention mechanism can positively impact the model's performance, experiments were conducted on the Chinese SanWen dataset for comparison. The experimental results are shown in Table 5, include the CAMFP (none CA) model, which directly concatenates character and word information after removing the cross-attention mechanism, and the SAMFP model, which replaces the cross-attention mechanism with two independent self-attention mechanisms for character and word processing.

Table 5 shows that the CAMFP model with the cross-attention mechanism achieves the highest F1-score among the three character-word fusion strategies. The SAMFP model shows a 2.1% improvement in F1-score compared to the CAMFP (none CA) model, as the SAMFP model incorporates a self-attention mechanism prior to character-word fusion, which captures the correlation features between characters and words within the local context. The proposed CAMFP model achieves a 1.91% improvement in F1-score compared to the SAMFP model. This is because the character-word cross-fusion network in the CAMFP model utilizes the attention interaction between characters and words by exchanging the query feature vectors of characters and words. It effectively leverages characters' and words' different contextual semantic information to select important feature information, thereby enhancing the model's feature extraction capability.

4.4.3. *Effectiveness analysis of the DSC network module and the GCN network module.* To further verify the effectiveness of different perceptual modules in feature-aware networks, the relation extraction methods of CAMFP, CAMFP-DSC, CAMFP-GCN, and CAMFP-DSC-GCN are correlated on the Chinese SanWen dataset in this paper. The experimental results are shown in Figure 6 and Table 6, where the CAMFP-DSC is a separate removal

TABLE 6. The effect of cross-attention network on model performance

Model	P	R	F1-score
CAMFP	70.97%	72.18%	**71.57%**
CAMFP-DSC	69.21%	70.15%	69.68%
CAMFP-GCN	67.43%	72.33%	69.79%
CAMFP-DSC-GCN	65.16%	71.88%	68.50%

of the DSC network module on top of CAMFP. The CAMFP-GCN is a separate removal of the GCN network module on top of the CAMFP. CAMFP-DSC-GCN represents the removal of both the DSC network module and the GCN network module on top of the CAMFP, and the output of the character and word cross-fusion network is delivered to the prediction output network after passing through the sentence-level attention mechanism.

Table 6 shows that when any network module in the feature-aware network or the entire feature-aware network is removed from CAMFP, the model's performance decreases. Among them, when DSC is removed, the F1-sorces of the model decreases by 1.89%, indicating that DSC in this paper can effectively extract local information of different granularities in the sentence to enhance the semantic representation ability of the sentence. When GCN is removed, the F1-sorces of the model decreases by 1.78%, indicating that GCN in this paper can capture the syntactic structure information and dependency relationships between words in the sentence to enhance the model's understanding of global semantic information. When both DSC and GCN are removed, the F1-sorces of the model decreases by 3.07%, indicating that these two modules can independently perceive the sentence's local information and syntactic structure information, avoiding the mutual influence between the perception of local information and syntactic structure information. The combined effect of these two types of information can more comprehensively represent the sentence's semantic information and improve the model's relationship extraction performance.

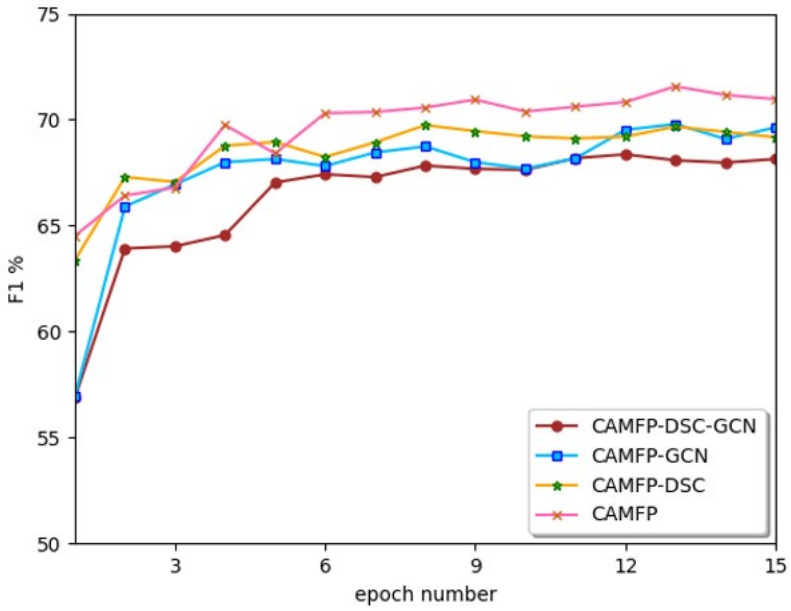

FIGURE 6. F1-score of four models on the Chinese SanWen dataset

From the F1-sorces curves of the four models in Figure 6, it can be seen that the F1-sorces of CAMFP, CAMFP-DSC, CAMFP-GCN, and CAMFP-DSC-GCN rapidly increases with the increase of model iterations in the initial stage of model training, and then stabilizes with certain fluctuations. Among them, the F1-sorce curves of CAMFP, CAMFP-DSC, and CAMFP-GCN are superior to the F1-sorce curve of CAMFP-DSC-GCN. The experimental results show that the introduced DSC module and GCN module can improve the performance of the relationship extraction model, and the performance improvement is most significant when the DSC and GCN modules work together in the model.

5. **Conclusion.** Currently, Chinese relation extraction models that rely solely on characters face challenges in obtaining rich semantic information and cannot fully leverage text's structural information during neural network feature extraction. This paper proposes a relation extraction method based on CAMFP to address this issue. The proposed method first utilizes BERT to encode sentences and construct character and word vectors. It then employs character-word cross-attention to select important feature information from characters and words, resulting in a fused sequence of characters and words. Next, a feature extraction network is constructed, which uses GCN and DSC to capture syntactic features between words and different granularity local features in the sentence sequence. An attention mechanism is applied to form sentence features for classification. Finally, softmax classifier is employed for relation extraction. To validate the force of the proposed method, extensive experiments are conducted on the Chinese SanWen dataset. The experimental results demonstrate that the proposed method outperforms existing RNN, CNN, and BERT methods, achieving an F1-score of 71.57%. In addition, related experiments were performed on the Chinese SanWen dataset, which demonstrates that each network module in the proposed model can improve the performance of the relationship extraction model. In future work, we will study how to optimize the sentence graph structure to decrease the influence of extraneous information on relationship extraction, such as using pruning strategies to remove redundant edges in the sentence structure graph based on this paper.

REFERENCES

[1] R. Lu, C. Fei, C. Wang, S. Gao, H. Qiu, S. Zhang, and C. Cao, "HAPE: A programmable big knowledge graph platform," *Information Sciences*, vol. 509, pp. 87–103, 2020.

[2] Y. Zhou, Y. Jun, C. Xiang, J. Fan, and D. Tao, "Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 12, pp. 5947–5959, 2018.

[3] B. Shi, and T. Weninger, "Discriminative predicate path mining for fact checking in knowledge graphs," *Knowledge-Based Systems*, vol. 104, pp. 123–-133, 2016.

[4] C. N. Santos, B. Xiang, and B. Zhou, "Classifying relations by ranking with convolutional neural networks," *Computer Science*, vol. 86, no. 86, pp. 132–-131, 2015.

[5] Y. Xu, L. Mou, L. Ge, Y. Chen, and J. Zhi, "Classifying relations via long short term memory networks along shortest dependency paths," *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing(EMNLP 2015)*, Association for Computational Linguistics, 2015, pp. 1785–1794.

[6] M. Xiao, and C. Liu, "Semantic relation classification via hierarchical recurrent neural network with attention," *In Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers (COLING 2016)*, Association for Computational Linguistics, 2016, pp. 1254–1263.

[7] T. Zhang, H. Lin, M. M. Tadesse, Y. Ren, X. Duan, and B. Xu, "Chinese medical relation extraction based on multi-hop self-attention mechanism," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 2, pp. 355–-363, 2021.

[8] Q. Zhang, M. Chen, and L. Liu, "An effective gated recurrent unit network model for Chinese relation extraction," *DEStech Transactions on Computer Science and Engineering*, pp. 262–-267, 2018.

[9] S. Rönnqvist, N. Schenk, and C. Chiarcos, "A recurrent neural model with attention for the recognition of chinese implicit discourse relations," *arXiv preprint*, arXiv:1704.08092, 2017.

[10] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidi-rectional transformers for language understanding," *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computa-tional Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2019, pp. 4171–4186.

[11] A. Yates, M. Banko, M. Broadhead, M. Cafarella and S. Soderland, "Textrunner: open information extraction on the web. In Proceedings of Human Language Technologies ," *The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2007)*, Association for Computational Linguistics, 2007, pp. 25–26.

[12] A. Culotta, and J. Sorensen, "Dependency tree kernels for relation extraction," *In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 2004, pp. 423.

[13] S. Xu, S. Sun, Z. Zhang, F. Xu, and J. Liu, "BERT gated multi-window attention network for relation extraction," *Neurocomputing*, vol. 492, pp. 516–-529, 2022.

[14] K. Sun, R. Zhang, Y. Mao, S. Mensah, and X. Liu, "Relation extraction with convolutional network over learnable syntax-transport graph," *In Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 5, pp. 8928–8935, 2020.

[15] Z. Quan, W. Zeng, X. Li, Y. Liu, Y. Yu, and W. Yang, "Recurrent neural networks with external addressable long-term and working memory for learning long-term dependences," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 3, pp. 813–-826, 2019.

[16] E. K. Wang, X. Zhang, F. Wang, G. Ding, T. Y. Wu and C. M. Chen, "Multilayer dense attention model for image caption," *IEEE Access*, vol. 7, pp. 66358–-66368, 2019.

[17] S. Zhang, X. Su, X. Jiang, M. Chen, and T. Y. Wu, "A traffic prediction method of bicycle-sharing based on long and short term memory network," *Journal of Network Intelligence*, vol. 4, no. 2, pp. 17–-29, 2019.

[18] S. Kumar, A. Damaraju, A. Kumar, S. Kumari, and C. M. Chen, "LSTM Network for Transportation Mode Detection," *Journal of Internet Technology*, vol. 22, no. 4, pp. 891–-902, 2021.

[19] J. Li, M. T. Luong, D. Jurafsky, and E. Hovy, "When are tree structures necessary for deep learning of representations?," *arXiv preprint*, arXiv:1503.00185, 2015.

[20] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao and B. Xu, "Attention-based bidirectional long short-term memory networks for relation classification," *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (volume 2: Short papers)*, Association for Computational Linguistics, 2016, pp. 207–212.

[21] Z. Li, Y. Sun, J. Zhu, S. Tang, and H. Ma, "Improve relation extraction with dual attention-guided graph convolutional networks," *Neural Computing and Applications*, vol. 33, no. 6, pp. 1773–-1784, 2021.

[22] Q. Zhao, T. Gao, and N. Guo, "A novel chinese relation extraction method using polysemy rethinking mechanism," *Applied Intelligence*, vol. 53, no. 7, pp. 7665–7676, 2023.

[23] J. Zhang, K. Hao, X. Tang, X. Cai, Y. Xiao, and T. Wang, "A multi-feature fusion model for Chinese relation extractionwith entity sense," *Knowledge-Based Systems*, vol. 206, pp. 106348, 2020.

[24] Z. Li, N. Ding, H. Zhen, and Y. Shen, "Chinese relation extraction with multi-grained information and external linguistic knowledge," *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2019, pp. 4377–4386.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, and L. Kaiser, "Attention is all you need," *In Proceedings of the 31st International Conference on Neural Information Processing Systems(NIPS 2017)*, Curran Associates Inc., 2017, pp. 6000–-6010.

[26] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," *arXiv preprint*, arXiv:1809.10185, 2018.

[27] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2017, pp. 1251–1258.

[28] J. Xu, J. Wen, X. Sun and Q. Su, "A discourse-level named entity recognition and relation extraction dataset for chinese literature text," *arXiv preprint*, arXiv:1711.07010, 2017.

[29] D. Lin, and X. Wu, "Phrase clustering for discriminative learning," *In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Association for Computational Linguistics, 2009, pp. 1030–1038.

[30] R. Cai, X. Zang and H. Wang, "Phrase clustering for discriminative learning," *Bidirectional recurrent convolutional neural network for relation classification*, Association for Computational Linguistics, 2016, pp. 756–765.

[31] S. Zhao, M. Hu, Z. Cai, Z. Zhang, T. Zhou, and F. Liu, "Enhancing chinese character representation with lat-tice-aligned attention," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 7, pp. 3727–3736, 2021.