International Journal for Housing Science and Its Applications
Publish August 6, 2025. Volume 46, Issue 3 Pages 4831-4840

https://doi.org/10.70517/ijhsa463413

A Study on Chinese Syntactic Structure Classification Method Based on Word Vector Modeling

Song Wang[1,*]
[1] School of Chinese Language and Literature, Xinyang College, Xinyang, Henan, 464000, China
Corresponding authors: (e-mail: andywong1985@126.com).

Abstract As a key link to improve the performance of tasks such as semantic understanding and machine translation, the study of Chinese syntactic structure classification helps to promote the rapid development of natural language processing technology. In this study, the Glove pre-trained word vector model is used to vectorize the Chinese vocabulary, and the semantic associations between words are modeled by contextual information. Then the BiLSTM model is combined to extract the global syntactic features of sentences, while the multi-head self-attention mechanism is introduced to improve the interpretability of the model. The graph convolutional network layer is further designed to obtain the syntactic structure classification probability through the softmax function. The syntactic structure classification precision, recall, and F1 score of the CSSLSTM model on the CTB5 dataset are 0.951, 0.947, and 0.949, respectively, which are much higher than the comparison methods. When the HEAD number of the model's multi-head attention mechanism is 4, the model's classification performance achieves the best results on both CTB5 and CTB7 datasets. The confusion matrix of syntactic structure classification shows that the model has an accuracy of more than 0.92 for the syntactic structures of "subject-verb", "subject-verb-object", "linked sentence", "put word sentence", "subject word sentence", "compared sentence", "existing sentence" and "concurrent sentence", and the average accuracy of syntactic structure classification in CTB5 and CTB7 datasets is 0.941 and 0.944, respectively, and the classification effect is better.

Index Terms Glove word vector model, BiLSTM model, multi-head self-attention mechanism, syntactic structure classification

I. Introduction

Natural Language Processing (NLP), an important direction in the field of Artificial Intelligence, is an interdisciplinary discipline that integrates mathematics, computer science and linguistics, and studies the communication between humans and computers to represent, understand and apply natural language in a symbolic way [1]-[3]. Linguistic modeling has long been the basis and core of symbolic representation of natural languages [4]. According to the size of language granularity, language modeling can be classified into character level, word level, sentence level and chapter level [5]. Character-level language modeling is based on words, which are considered to be the smallest processing unit [6]. Unlike most Western languages, Chinese text does not have spaces as segmentation boundaries, so Chinese is often modeled at the word level after word segmentation, and word-level modeling considers words to be the most basic unit for carrying semantics [7], [8]. Sentence-level language modeling is based on character-level or word-level modeling, and combines syntactic structure and lexical meaning to derive a formal representation that reflects the meaning of the sentence [9]-[11]. Chapter-level language modeling is based on sentence-level modeling, which formally represents the hierarchical structure and semantic relationships of words or sentence segments in a chapter. Small-grained modeling is the basis and prerequisite for large-grained modeling, and the quality of character modeling or word modeling directly affects the effectiveness of sentence modeling and chapter modeling [12].

English already has a recognized and mature syntactic structure system, but the syntactic structure of modern Chinese is not mature enough, and no recognized syntactic structure of modern Chinese has been formed yet, and the existing syntactic theories of Chinese have basically been borrowed from Western linguistic theories [13]-[15]. In view of the characteristics of the Chinese language, applying the syntactic analysis method of English to Chinese in a rigid way will inevitably lead to poor results [16]. Some scholars have done research and statistical analysis in combination with syntactic treebanks, pointing out the special difficult problems in the analysis of Chinese syntactic structures and their reasons leading to the low accuracy of syntactic analysis [17], [18]. Therefore, it is necessary to explore a personalized syntactic system and syntactic structure analysis method with characteristics suitable for Chinese.

The article proposes a method to classify Chinese syntactic structure based on word vector model, firstly, the Chinese text data are cleaned to reduce the interference such as deactivated words, and the Glove word vector model is utilized for vector coding in order to obtain high-quality global information. Second, the historical and future information of the text data is captured by BiLSTM model, which is combined with the multi-head self-attention mechanism to improve the model's ability to deal with long-distance dependencies. The multi-head self-attention score of global semantic information and the neighbor matrix with dependent syntactic structure information are fused, parsed through the dependent syntactic tree, and the information matrix is inputted into the graph convolutional network, which outputs the classification probability of syntactic structure after the update operation.

II. Key technologies

II. A.Preprocessing of Chinese Text Data

Text data preprocessing is a very critical step in the major tasks in the field of NLP [19], and the processing results can directly affect the performance of downstream work.

First of all, the use of spaces and punctuation for segmentation, Chinese text segmentation generally requires the use of complex segmentation algorithms. Therefore, the Chinese dataset participle tool used in the experiments of this paper is Jieba participle (Jieba). Then, the regular expression of Python language is used directly to remove non-text parts and deactivate the word list to remove some pronouns, dummy words or words that do not express a clear meaning so as to clean the text. After the above two processes, you can get a standardized text dataset.

II. B.Text representation based on Glove word vector modeling

Since computers can only recognize binary data, when a large amount of textual data needs to be improved with the help of computers, it is necessary to construct a variety of textual representations to encode each word in the textual data into a vector or matrix form, which enables computers to recognize and process them.

Glove (Global Vectors for Word Representation) [20], compared with Word2Vec, not only considers the local information between contexts, but also applies co-occurrence matrix to introduce higher quality global information. Its operation process is mainly divided into three steps:

First, a co-occurrence matrix X is created from the given corpus, where each vector $X_{m,n}$ denotes the number of times the words m and n co-occur. Second, the approximate relationship between the two is obtained from the given co-occurrence matrix and word vectors as:

$$\log(X_{mn}) \approx u_m^T u_n + b_m + b_n \tag{1}$$

where u_m and u_n denote the word vectors of words m and n, respectively, and b_m and b_n denote the bias vectors; and finally, the corresponding loss function is constructed as:

$$L = \sum_{m,n=1}^{V} f(X_{mn})(\log(X_{mn}) - u_m^T u_n + b_m + b_n)^2 \tag{2}$$

where $\log(X_{mn})$ denotes the true value and $u_m^T u_n + b_m + b_n$ denotes the predicted value. $f(X_{mn})$ is the weight function, which is non-decreasing and cannot be increased infinitely when the word frequency is too high, so the following segmentation function is chosen as the weight function $f(x)$:

$$f(x) = \begin{cases} 1 & \text{if } x \geq x_{\max} \\ \left(\dfrac{x}{x_{\max}}\right)^{\alpha} & \text{otherwise} \end{cases} \tag{3}$$

II. C.Bidirectional long- and short-term memory networks

The gates in LSTM [21] contain sigmoid neural network layer and dot product operation. The key is the cell state C_t, where the input signal is transformed by the sigmoid layer into a value between 0 and 1, which represents the weight of the passed information. The LSTM implements the control by means of three kinds of gates: the input gate i, the forgetting gate f and the output gate o. By reading h_{t-1} and x_t and undergoing the σ transformation, the information f_t to be discarded and the information i_t to be retained in the C_{t-1} is derived and the cell state is updated, and the g_t represents the control parameters. The new cell state C_t is computed based on the previous parameters. Finally, x_i and h_{t-1} do σ transform to get the information to be output o_t and multiply it with the tanh transform of C_t to get the final result. The formula is as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{4}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5}$$

$$g_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{6}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{7}$$

$$C_t = f_t \square\ C_{t-1} + i_t \square\ g_t \tag{8}$$

$$h_t = o_t \square\ \tanh(C_t) \tag{9}$$

Since a single LSTM can only learn long term dependencies in a single direction, for this reason researchers have proposed to feed the input sequence from both forward and reverse directions into two separate LSTMs for processing to get the feature vectors in both directions, and then splice the two feature vectors as the final feature vector. This new model, called Bidirectional LSTM model (BiLSTM).The structure of BiLSTM is shown in Fig. 1.The parameters of forward and reverse LSTMs in BiLSTM model are independent of each other, and they share a list of word vectors (input). The model enables the feature data captured at a given moment to contain not only past information but also future information.

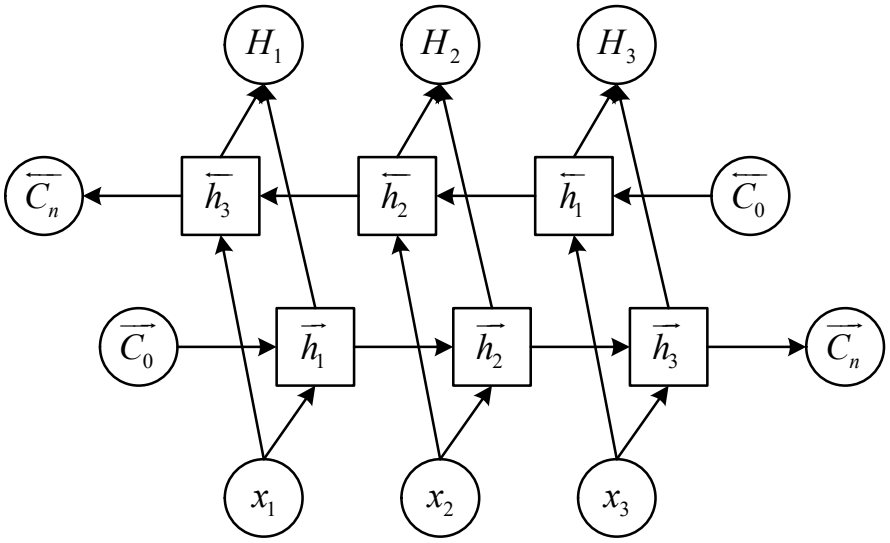

Figure 1: BiLSTM Network Structure

II. D.Dependency Syntax Analysis

The main objective of Dependency Syntactic Analysis (DP) [22] is to obtain the structure of a sentence at the syntactic level, and it achieves this by analyzing the dependencies of the words in the sentence. Dependency denotes a special dominance relationship between two words with directional pointing. The core of syntactic analysis is the dependency relation, a dependency relation connecting two words (nodes): the dominant and the subordinate. Studies of dependency syntactic analysis generally fall into two categories: transfer-based, and graph-based.

Transfer-based methods are performed based on the state of the sentence, and thus require first analyzing the sentence state to characterize the relationship between words in terms of a sequence of states. Each state transfer represents an analysis. The transfer-based approach generates the sequence into a tree structure usually requires 2 actions: shift, reduce.Currently, the Stack-LSTM model is more effective, which is implemented using 3 LSTM layers in modeling the state, the sequence to be input and the sequence of actions, respectively.

The graph-based approach finds the one with the highest probability as the final graph among all possible generated dependency graphs for a given sentence and model parameters. A widely used and effective model is Biaffine. Its dependency probabilities are obtained by relying on neural network models to obtain fully connected graphs. The edges in the graph refer to the probability of each node pointing to another node. Finally the graph is transformed into the form of a tree using the Maximum Spanning Tree (MST) method. The model is simple in principle, similar to the self-Attention model, and has achieved good results in several experiments.

III. Structural classification models incorporating semantic and syntactic information

In this section, a long and short-term memory network model (CSSLSTM) that fuses semantic information with syntactic structural information is proposed, which, unlike the models proposed in previous studies, enhances the traditional LSTM model by representing the semantic information captured by the multi-head attention mechanism as an attention score matrix and fusing it with a neighboring matrix with sentence dependencies and structural information. Finally, aspect-specific features for aspect word sentiment classification are obtained using multilayer graph convolution operations.

III. A. CSSLSTM model architecture

The overall architecture of CSSLSTM is shown in Fig. 2, and the model is mainly composed of five parts: word vector embedding layer, Bi-LSTM layer, attention module, dependency parsing module, and graph convolutional network layer. The word vector embedding layer vectorizes the input text words, and the hidden state obtained through the Bi-LSTM layer is used as the input, which is fed into the attention module and the dependency parsing module to get the deep semantic information and the syntactic structure of the text, and then fed into the graph convolutional layer for node updating, and then the classification result is obtained through the classification function sotfmax.

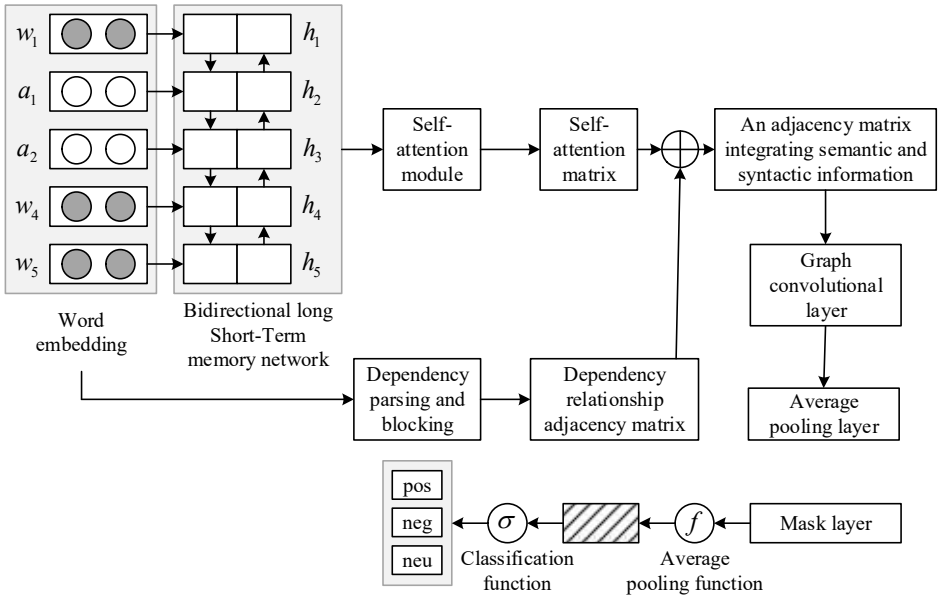

Figure 2: Overall Architecture of the CSSLSTM model

III. B. Word Vector Embedding Layer

Given a text sequence $s=\{s_1,s_2,...,s_n\}$, and aspect words of length m $a=\{a_1,a_2,...,a_m\}$, and also a subsequence of the sentence s. One of the aspect words can be a word or a word phrase.

This experiment uses Glove word vectors to map each word in the input utterance to a low-dimensional real-valued vector to obtain the embedding matrix $E\in R^{|V|\times d_{emb}}$, where $|V|$ is the size of the vocabulary in the lexicon, and d_{emb} denotes the size of the dimension of each word vector. Thus the word embedding $x=\{x_1,x_2,\ldots,x_n\}$ corresponding to the sentence s.

III. C. Bi-LSTM layer

Bi-LSTM at each time point, the hidden states of forward and reverse layers are combined to represent the complete information at that time point. The specific calculation formula is shown in:

$$\vec{h}_t=f(U^{(1)}h_{t-1}^1+W^{(1)}x_t+b^{(1)}) \tag{10}$$

$$\overleftarrow{h}_t=f(U^{(2)}h_{t-1}^2+W^{(2)}x_t+b^{(2)}) \tag{11}$$

$$h_t=\vec{h}_t\oplus\overleftarrow{h}_t \tag{12}$$

where $\vec{h}_t$ denotes the hidden layer vector of the forward LSTM output, $\overleftarrow{h}_t$ denotes the hidden state vector of the backward LSTM output, and $\oplus$ denotes the splice of the vector.

The forward LSTM representation of sentence s is: $\overrightarrow{H^F} = \{\vec{h}_1, \vec{h}_2, ..., \vec{h}_n\}$, the inverse LSTM representation of sentence s is: $\overleftarrow{H^B} = \{\overleftarrow{h}_1, \overleftarrow{h}_2, ..., \overleftarrow{h}_n\}$, the corresponding vectors of $\overrightarrow{H^F}, \overleftarrow{H^B}$ are concatenated to the final hidden state vectors $H = \{h_1, h_2, ..., h_n\}$, where $h_i \in R^{2d}$, and H contains the subsequence of the corresponding vocabulary $h_a = \{h_{a_1}, h_{a_2}, ..., h_{a_m}\}$, using H as the initial input to the model.

III. D. Multi-pronged self-attention mechanisms

The advantage of the self-attention mechanism is that self-attention allows the model to focus directly on information at any position in the sequence, improving the ability to handle long-distance dependencies. By analyzing the attention weight matrix, it is possible to visualize how the model allocates its attention between different parts of the sequence when making decisions, improving the interpretability of the model. In this paper A_{self}, t heads can also be used to construct the self-attention mechanism, which captures the semantic information between any two words in a single sentence, and the hidden state vector output from the coding layer provides a query index for the attention computation:

$$A_{self}^{i} = \frac{QW^{Q} \times (KW^{K})^{T}}{\sqrt{d}_{k}} \tag{13}$$

where Q and K are derived from the hidden vectors H produced by the coding layer, $W^{Q} \in R^{d \times d}$ and $W^{K} \in R^{d \times d}$ are the weights that can be learned, and d_k is the vector of the KW^{K} computed dimension.

As the neural network gets deeper, the more expressive the model is, but with the deepening of the network will bring a series of problems, such as the problem of gradient disappearance or gradient explosion, the emergence of the residual network greatly alleviates these problems and simplifies the learning process of the model and enhances the model's generalization ability. So the A_{self}^{i} obtained in this paper carries out the operation of residual connection, i.e., the output of the self-attention mechanism is added to the input. As in Eq:

$$A_{self}^{i} = H + A_{self}^{i} \tag{14}$$

III. E. Dependency Resolution Module

A dependency syntax tree can be interpreted as a graph G with n nodes, where nodes denote words in a sentence and edges denote syntactic dependency paths between words in the graph. The dependency tree G of any sentence can be represented as an adjacency matrix D of $n \times n$, where $D_{ij} = 1$ if node i is connected to node j through a single dependency path in G, and $D_{ij} = 0$ otherwise, as in Eq:

$$D_{ij} = \begin{cases} 1 & \text{Nodes } i, j \text{ have a dependency relationship} \\ 0 & \text{Nodes } i, j \text{ have no dependency relationship} \end{cases} \tag{15}$$

In the attention layer, t head attentions yield t attention matrices, so the matrix D is copied t times so that the number of dependent syntactic adjacency matrices is the same as the number of attention matrices, as in Eq:

$$D = \{D^{1}, ..., D^{t}\} \tag{16}$$

The fusion of the multi-head self-attention score focusing on global semantic information and the adjacency matrix with dependent syntactic structural information yields a matrix A^{i} rich in semantic and syntactic structural information, as in Eq:

$$A^{i} = softmax(A_{aspect}^{i} + A_{self}^{i} + D^{i}) \tag{17}$$

A matrix $A^{i} \in R^{n \times n}$ based on fusing semantic information syntactic structural information.

III. F. Graph Convolutional Network Layer

In the dependent syntax layer t different matrices rich in semantic and syntactic information, $A \in R^{t \times n \times n}$, are generated, and thus t graph convolution operations need to be performed in each layer of the graph convolution network. If h^{l-1} is taken as the input state, then h^{l} represents the output state of the l th layer of graph convolution. The h^{0} is denoted as the output of the initial text after the sentence encoding layer (Bi-LSTM). Each

node in the l th layer graph convolution neural network is updated according to the hidden representation of its neighborhood, which is formulated as:

$$h_i^l = \sigma(\sum_{j=1}^{n} A_{ij} W^l h_j^{l-1} + b^l) \tag{18}$$

where h_i^l denotes the hidden state representation of node i in the l th layer graph convolutional neural network, b^l denotes the bias of the l th layer graph convolutional neural network, W^l is the linearly varying weights, which is the learnable parameter matrix, and σ denotes the nonlinear activation function, such as ReLU. Using A_{ij}, which is rich in semantic syntactic information, as the adjacency matrix of the graph convolutional neural input, the final output of the graph convolutional neural network at the l th layer is denoted as H^l, as in Eq:

$$H^l = \{h_1^l, h_2^l, ..., h_n^l\} \tag{19}$$

The final feature representation of the word is obtained after aggregating the node information in each layer of the l -layer graph convolutional neural network. The non-aspect word representations in the feature representations output by the GCN are masked to obtain the aspect word feature representations, and the aspect word feature representations h_a^l retain most of the aspect information through average pooling:

$$h_a^l = f(h_{a_1}^l, h_{a_2}^l, ..., h_{a_m}^l) \tag{20}$$

where $f(\cdot)$ denotes the average pooling function used to augment the aspect representation through the graph convolutional neural network layer.

The representation of aspect words h_a^l is fed into the linear layer and then passed through the softmax function in order to obtain the syntactic structure classification probability:

$$c(a) = softmax(W_c h_a^l + b_c) \tag{21}$$

where W_c and b_c are both learnable parameters.

Finally, the cross-entropy loss function is used as the loss function in this paper:

$$L(\theta) = -\sum_{(s,a)\in D} \sum_{c\in C} \log c(a) \tag{22}$$

IV. Experimentation and analysis

IV. A. Experimental data

In this section, experiments are conducted on two public Chinese datasets, CTB5 and CTB7, both of which have annotation results for participle, lexical and dependent syntax. For CTB5 and CTB7, the data division follows the existing work without any modification. In order to test the adaptability on other languages, Japanese and Vietnamese, which do not have participle boundaries, are selected from the UD Treebank v2.6 for testing, and the evaluation metrics use the F1 value of the composite performance index.

IV. B. Hyperparameter settings

To facilitate comparison with existing studies, the hyperparameter settings for BiLSTM in existing studies are followed. The dimensionality of the BiLSTM is set to 500, the number of layers of all MLPs is set to 1, the activation function is relu, and the output dimension of the perceptron used to score the dependency arcs and the second-order subtree d is set to 300. A dropout layer is added to the BiLSTM and all the MLPs, and the dropout rate is set to 0.30. The weighting parameter in the loss function formulation is set to 0.5, and the Batch size of the model is 128 for each round of training, and the joint model uses the Adam optimizer to optimize the model. is 0.5, the Batch size of the model is 128 for each round of training, and the joint model is trained for 100 rounds using the Adam optimizer for gradient computation as well as parameter updating, and the learning rate is set to 0.0001. The models are validated on the development set and the model with the best dependent syntactic F1 value on the development set is retained for final evaluation on the test set.

IV. C. Experimental environment

The hardware configuration parameters used in this experiment are as follows:

Programming language is python, version python 3.6.5, development tool is VSCode. Web server used is Apache Tomcat 7.0.6 with MySQL 8.1 database.

CPU is Inter(R)core(TM) i5-5200U CPU @ 2.2GHz, RAM is 4GB, GPU is NVIDIA GEFORCE GTX 1050ti, and OS is Window10 64-bit.

IV. D. Analysis of the effect of categorizing Chinese syntactic structures

IV. D. 1) Performance Comparison for Classifying Syntactic Structures

This section first compares the classification results of different methods on the Chinese text grammar structure classification dataset to verify the effectiveness of this paper's method. The method of this paper (CSSLSTM) is compared with the following methods:

(1) DSR (Dictionary based sparse representation for domain adaptation) method.

(2) DTL (Deep transfer learning) method.

(3) MSTL (Multisource migration learning) method.

(4) SST (Syntactic structure migration) method.

The comparison results of precision rate, recall rate and F1 score of different Chinese text syntactic structure classification methods are shown in Table 1. The following conclusions can be drawn from the table:

(1) Compared with the comparison methods DSR, DTL, MSTL and SST methods, the average precision rate of this paper's method on the two Chinese text datasets, CTB5 and CTB7, is improved. For example, on the CTB5 dataset, this paper's method improves 0.208, 0.028, 0.185, and 0.202, respectively, compared with the above methods.Meanwhile, although the DTL method achieves a high accuracy rate on the CTB5 dataset, the accuracy rate decreases more on the CTB7 dataset. The method in this paper achieved the highest accuracy rate on both datasets.

(2) The recall and F1 score of this paper's method on both datasets achieve excellent results, which are significantly higher than those of the comparison methods.The recall and F1 score on the CTB5 dataset are 0.947 and 0.949, respectively, and on the CTB7 dataset they are 0.932 and 0.948, respectively.This indicates that this paper's word-vector syntactic structure classification method fused with deep learning has a high syntactic structure recognition efficiency.

Table 1: Comparison of different Chinese text syntax structure classification methods

Date set	Index	Ours	DSR	DTL	MSTL	SST
CTB5	Precision	0.951	0.743	0.923	0.766	0.749
	Recall	0.947	0.756	0.908	0.789	0.754
	F1	0.949	0.749	0.915	0.777	0.751
CTB7	Precision	0.964	0.758	0.845	0.799	0.746
	Recall	0.932	0.736	0.827	0.783	0.768
	F1	0.948	0.747	0.836	0.791	0.757

IV. D. 2) Analysis of the effectiveness of attention mechanisms

The multi-head self-attention mechanism is usually applied in the field of machine translation to capture the internal associations and features of sentences. The main purpose of this part of the experiment is to explore the improvement of the multi-head attention mechanism on the performance of the model in this paper and its applicability to the task of classifying Chinese syntactic structures. Since the final representation of features in multi-head attention is related to the number of heads, this paper explores the effect of the number of heads on the model performance when performing multi-head attention.

The performance of the multi-head attention mechanism of this paper's model is tested on the CTB5 and CTB7 datasets with the multi-head parameter head={1,2,3,4,5,6}, respectively. Since the dimensionality reduction of the multi-head attention when performing head computation is averaged, it is important to make sure that the number of heads selected is able to be integrally divisible by the dimensionality of the hidden layer. The evaluation metrics for the experiment are Precision, Recall, and F1 score.

Figure 3 shows the performance test results of this paper's attention mechanism for syntactic structure categorization under different HEAD numbers. It can be observed from the test data in the figure: when the number of heads is 4, the performance of this paper's multi-head attention mechanism is the best on 2 datasets. The Precision, Recall, and F1 scores in the CTB5 dataset are 0.949, 0.938, 0.943, respectively, and in the CTB7 dataset, they are 0.9445, 0.932, 0.938, respectively. when the head number is small (1, 2, or 1), the performance is poor, indicating that the model's ability to mine deep features is insufficient at this time, and the HEAD number is too small resulting in the contextual representation not being able to contain important syntactic structural features. When the number of HEAD is large, the model does not necessarily achieve better performance, such as at HEAD of 6, the

performance of the multi-head attention mechanism model is instead reduced. It may be because as the number of HEAD increases, the model parameters also increase, making the model generalization ability weaker. So on the whole, the multi-head attention model in this paper has the best performance when the number of heads is 4.

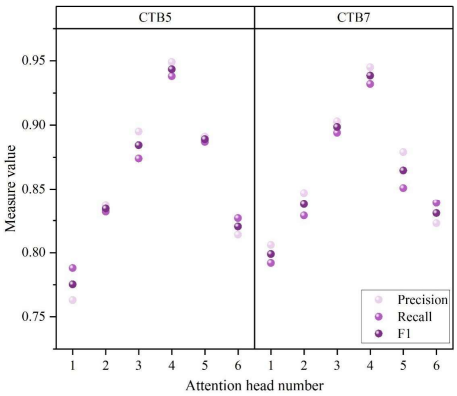

Figure 3: The performance test results of the syntax structure of different heads

IV. D. 3) Recognizing Syntactic Structures in Chinese Texts

In this section, the syntactic structure recognition experiment is carried out in the above dataset, and the confusion matrix is used to evaluate the classification performance of the Chinese syntactic structure of the proposed model. The Chinese syntactic structures in the two datasets are marked as "subject-verb", "subject-verb-object", "linked sentence", "put word sentence", "subject word sentence", "comparative sentence", "existing sentence", and "concurrent sentence", which are recorded as syntactic structures 1~8 in turn. After data cleaning and screening, the total amount of data used in the experiments of Chinese syntactic structure recognition in the two datasets was 8000 and 10000, respectively, and the division ratio of the training and test datasets was 8:2.

Figures 4 and 5 are the syntactic structure classification and identification confusion matrices of the model on the CTB5 and CTB7 datasets, respectively. The results of the confusion matrix show that the proposed model can better realize the recognition of "subject-verb", "subject-verb-object", "linking sentence", "put word sentence", "being word sentence", "comparison sentence", "existing sentence" and "concurrent sentence" on the two datasets. The classification accuracy of syntactic structure on the CTB5 dataset was between 0.925~0.952, and the average accuracy was 0.941, among which the recognition accuracy of the "word and sentence" structure was the lowest, and there was a possibility that 2.1% of the structure could be divided into "concurrent sentence" structure. On the CTB7 dataset, the average accuracy is improved to 0.944, but the model still has a certain error in the recognition of the "conjunctive sentence" structure, and there is a 2.2% probability that it is predicted as a "subject-verb" structure. BiLSTM can effectively extract the relationship between the text before and after, which is helpful for the collection of syntactic structure information, while the Glove word vector model can effectively solve the problem of long-distance attenuation and have a stronger ability to extract semantic information.

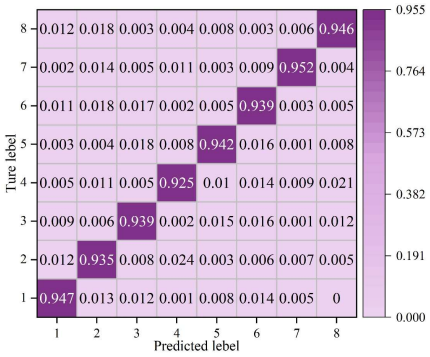

Figure 4: The identification confusion matrix on the CTB5 data set

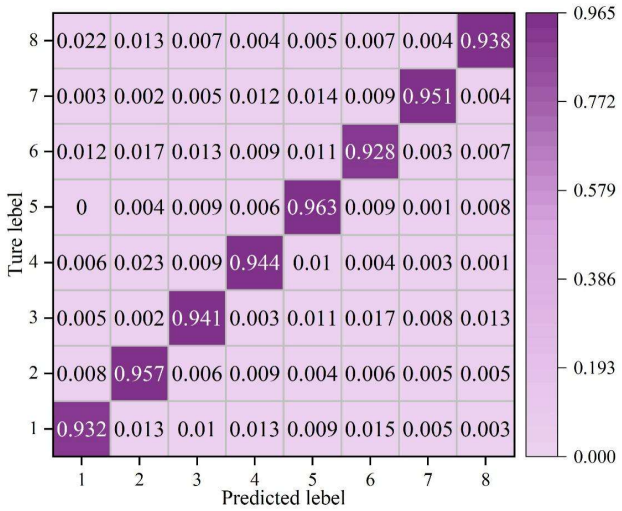

Figure 5: The identification confusion matrix on the CTB7 data set

V. Conclusion

In this paper, we design a syntactic structure classification model that integrates semantic and syntactic information based on the Glove word vector model, bi-directional long and short-term memory network and other techniques. The model maps the embedding matrix of each word by Glove word vector, bi-directional long and short-term memory network is used to represent the temporal information of the vector sequences, and the multi-head self-attention mechanism is introduced to obtain the semantic information between two words in a sentence. The syntactic structure information is fused with the semantic information, and the syntactic structure classification results are output through the graph convolutional network layer. In order to verify the effectiveness of this paper's method, several classification effect experiments are designed, and the experimental results show that the average classification precision, recall and F1 score of this paper's method on CTB5 and CTB7 datasets are higher than those of the comparison methods. Too small or too large head number in the multi-head attention mechanism affects the classification performance of the model, and the head number of 4 is the best, and the precision rate, recall rate and F1 score on CTB5 dataset are 0.949, 0.938 and 0.943 respectively. The accuracy of the model in identifying the structures of "subject-verb", "subject-verb-object", "linked sentence", "put word sentence", "subject word sentence", "comparative sentence", "existing sentence" and "conjunctive sentence" in Chinese is more than 90%, and the syntactic structure recognition rate in CTB5 and CTB7 datasets is the lowest "put word sentence" and "conjunctive sentence" structure, respectively. In conclusion, this study provides an extensible word vector model solution for Chinese syntactic analysis.

References

[1] Ali, A. A. S., & Shandilya, V. K. (2021). AI-Natural Language Processing (NLP). International Journal for Research in Applied Science and Engineering Technology, 9, 135-140.

[2] Priyadarshini, S. B. B., Bagjadab, A. B., & Mishra, B. K. (2020). A brief overview of natural language processing and artificial intelligence. Natural language processing in artificial intelligence, 211-224.

[3] Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., & Chanona-Hernández, L. (2014). Syntactic n-grams as machine learning features for natural language processing. Expert Systems with Applications, 41(3), 853-860.

[4] Pavlick, E. (2023). Symbols and grounding in large language models. Philosophical Transactions of the Royal Society A, 381(2251), 20220041.

[5] Lai, Y., Liu, Y., Feng, Y., Huang, S., & Zhao, D. (2021). Lattice-BERT: leveraging multi-granularity representations in Chinese pre-trained language models. arXiv preprint arXiv:2104.07204.

[6] Al-Rfou, R., Choe, D., Constant, N., Guo, M., & Jones, L. (2019, July). Character-level language modeling with deeper self-attention. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 3159-3166).

[7] Gerz, D., Vulić, I., Ponti, E., Naradowsky, J., Reichart, R., & Korhonen, A. (2018). Language modeling for morphologically rich languages: Character-aware modeling for word-level prediction. Transactions of the Association for Computational Linguistics, 6, 451-465.

[8] Shuang, K., Li, R., Gu, M., Loo, J., & Su, S. (2019). Major–minor long short-term memory for word-level language model. IEEE Transactions on Neural Networks and Learning Systems, 31(10), 3932-3946.

[9] Gu, K., Kabir, E., Ramsurrun, N., Vosoughi, S., & Mehnaz, S. (2023). Towards sentence level inference attack against pre-trained language models. Proceedings on Privacy Enhancing Technologies.

[10] Zhang, Y., Kamigaito, H., & Okumura, M. (2021, November). A language model-based generative classifier for sentence-level discourse parsing. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (pp. 2432-2446).

[11] Guan, J., Mao, X., Fan, C., Liu, Z., Ding, W., & Huang, M. (2021). Long text generation by modeling sentence-level and discourse-level coherence. arXiv preprint arXiv:2105.08963.
[12] Wu, Y. C., Yin, F., & Liu, C. L. (2017). Improving handwritten Chinese text recognition using neural network language models and convolutional neural network shape models. Pattern Recognition, 65, 251-264.
[13] Xu, Z. (2008). Analysis of syntactic features of Chinese English. Asian Englishes, 11(2), 4-31.
[14] Pan, V. J., & Paul, W. (2018). The syntax of complex sentences in Mandarin Chinese: A comprehensive overview with analyses. Linguistic Analysis, 42(1-2).
[15] Liao, W. W. R., & Lin, T. H. J. (2019). Syntactic structures of Mandarin purposives. Linguistics, 57(1), 87-126.
[16] Ming, C. (2023). Comparison and Analysis of English and Chinese Syntactic Features. Academic Journal of Humanities & Social Sciences, 6(16), 146-150.
[17] Yang, S., Cai, Y., Xie, W., & Jiang, M. (2021). Semantic and syntactic processing during comprehension: ERP evidence from Chinese QING structure. Frontiers in Human Neuroscience, 15, 701923.
[18] Liu, Z., & Lin, C. J. C. (2025). Grammar in Syntactic Adaptations of Chinese: The State of the Art. Handbook of Chinese Language Learning and Technology, 251-279.
[19] Umesh Gupta,Shubham Kandpal,Hayam Alamro,Mashael M. Asiri,Meshari H. Alanazi,Ali M. Al Sharafi & Shaymaa Sorour. (2025). Efficient malware detection using NLP and deep learning model. Alexandria Engineering Journal,124,550-564.
[20] Sai Srinivas Vellela,Roja D,NagaMalleswara Rao Purimetla,SyamsundaraRao Thalakola,Lakshma Reddy Vuyyuru & Ramesh Vatambeti. (2025). Cyber threat detection in industry 4.0: Leveraging Glove and self-attention mechanisms in BiLSTM for enhanced intrusion detection. Computers and Electrical Engineering,124(PA),110368-110368.
[21] Imen Jarraya,Safa Ben Atitallah,Fatimah Alahmed,Mohamed Abdelkader,Maha Driss,Fatma Abdelhadi & Anis Koubaa. (2025). SOH-KLSTM: A hybrid Kolmogorov-Arnold Network and LSTM model for enhanced Lithium-ion battery Health Monitoring. Journal of Energy Storage,122,116541-116541.
[22] Yuanxi Li,Haiyan Wang & Dong Zhang. (2025). An Entity-Relation Extraction Method Based on the Mixture-of-Experts Model and Dependency Parsing. Applied Sciences,15(4),2119-2119.