Metaphor Recognition Method based on Graph Neural Network

Zhou Chuwei[1], SHI Yunmei[2]
Beijing Information Science and Technology University, Beijing Key Laboratory of Internet Culture Digital Dissemination, Beijing 100101, Beijing, China[1, 2]
School of Computer, Beijing University of Information Technology, Beijing 100101, Beijing, China[1, 2]

Abstract—**This Metaphor is a very common language phenomenon. Human language often uses metaphor to express emotion, and metaphor recognition is also an important research content in the field of NLP. Official documents are a serious style and do not usually use rhetorical sentences. This paper aims to identify rhetorical metaphorical sentences in official documents. The use of metaphors in metaphorical sentences depends on the context. Based on this linguistic feature, this paper proposes a BertGAT model, which uses Bert to extract semantic features of sentences and transform the dependency relationship between Chinese text and sentences into connected graphs. Finally, the graph attention neural network is used to learn semantic features and syntactic structure information to complete sentence metaphor recognition. The proposed model is tested on the constructed domain dataset and the sentiment public dataset respectively. Experimental results show that the method proposed in this paper can effectively improve the recognition ability of metaphorical emotional sentences.**

Keywords—Sentiment analysis; metaphor recognition; graph neural network; attention mechanism

I. Introduction

The use of metaphorical sentences can make the described things more imaginative and vivid, and can show stronger emotions. However, official documents are issued by legal organs or organizations, and their contents should be clear and specific, and their expressions should be simple and easy to understand, so they do not need to deliberately pursue rhetoric and generally do not use metaphors, lyricism and symbols. Metaphor is a common rhetorical technique in metaphorical sentences, but it is generally not used in official documents for standardization. It is a time-consuming and laborious task to manually check metaphorical sentences in official documents, and using automatic detection tools can greatly reduce the workload of document writers and increase the accuracy and efficiency of identification.

Metaphor sentence recognition is the identification of the presence or absence of metaphorical usage in a sentence, which is essentially a dichotomous classification task. There are rule-based methods, statistical machine learning methods and deep learning methods for metaphor recognition. Using rule-based approaches [1][2] can effectively address simple, common metaphor types, but requires a lot of manual effort to design rules and does not easily cover the full range of identified types. Using traditional machine learning methods [3][4] can identify more comprehensive types of metaphorical sentences with higher recognition rates, but the method is difficult to learn deep semantic features and the model generalization is more difficult. In recent years, deep learning methods have become popular, and deep learning-based methods are widely used in metaphor recognition [5][6][7][8][9][10], which are similar to text classification tasks, where text is first transformed into vector form, and then neural network models are used to learn the semantic information of sentences. A better recognition result is usually obtained given enough corpus. In the last two years, the introduction of pre-trained language models [12] has further improved metaphor recognition, but the existing studies do not use syntactic structure information or do not deeply integrate syntactic structure information into the models.

Metaphorical sentences appear in a specific contextual environment, and without the contextual relationship, there is no meaning of metaphorical sentences. Therefore, identifying metaphorical sentences requires establishing a good contextual relationship between words, so as to identify whether a sentence has words containing metaphorical relationships. In this paper, we use Bert to extract the semantic features of sentences, convert the text into word embedding information, then perform syntactic dependency analysis, and integrate semantic information with syntactic structure information when constructing sentence connectivity graphs, and finally use Graph Attention Networks (GAT) to learn the semantic and syntactic structure information of sentences, and fully extract the sentence feature information to identify metaphorical sentences.

The contributions of this paper are as follows:

The proposed BertGAT model for recognizing metaphorical sentences in Kumon is proposed, which incorporates syntactic structure information of sentences based on the external knowledge of the pre-trained model to make sentence modeling more refined.

The proposed BertGAT model can learn the sentence structure information and weight the word nodes with varying importance, which in turn enriches the text feature information.

A rhetorical implicit sentiment recognition dataset for the metaphor class in the public domain is constructed, and the proposed BertGAT model in this paper achieves accurate results not only in this dataset, but also in the public dataset ChnSentiCorp with good results.

In this paper, a metaphor recognition model based on graph neural network and pre-training model bert is proposed.

Therefore, the graph neural network and pre-training model are introduced respectively. Finally, the validity of the model in domain data set and public data set is verified by experiments.

II. Related Studies

A. Metaphorical Sentence Recognition

Recognition of metaphorical sentences is a common and important problem in Natural Language Processing (NLP) and has been studied by researchers for a long time. The earliest metaphor recognition is the rule-based template approach proposed by Krishnakumaran [1], which relies on WordNet, an a priori knowledge base, to identify "A is B" noun metaphors, "verb+noun" verb metaphors, and "adjective+noun" metaphors by using its contextual relations and setting frequency thresholds “noun" verb metaphor and "adj+noun" adjective metaphor. The drawback of this method is that it relies too much on the a priori knowledge base WordNet and the identified metaphor sentences are simple. Noun metaphors are the most common expressions in lines, followed by verb metaphors, so researchers generally focus on the identification of these two types of metaphorical sentences [2][3][5][6][7][8][9][10].

Su Chang [2] proposed four algorithms to identify noun metaphors from different perspectives and recognized rich types of noun metaphor sentences with high accuracy, but the method relied entirely on WordNet. Shutova [6] proposed a clustering-based approach to identify verb metaphor sentences by taking verb subject groups and verb object groups that have been labeled as metaphorical expressions as seed sets and continuously expanding the seed sets by clustering. When these phrases appear in a sentence, it is judged as a metaphorical sentence, and the performance of this method depends on the quality and diversity of the seed words. Su Chang et al [3] mined multi-level features based on multi-level semantic networks, including full-text features, attribute features, perceptual features, etc., but for some complex metaphorical sentences, some attributes of the target domain are difficult to extract.

In recent years, deep learning has been developed and applied more and more widely in the field of NLP. The classical CNN model is not only simple but also powerful, Kim Y et al [4] applied CNN to text classification and achieved good results, while metaphor recognition is essentially text classification, some researchers use CNN models [7][9] in metaphor recognition to capture locally relevant features of text. Different ranges of semantic features can be captured by convolutional kernels of different sizes. Kui-Lin Su [9] used a pre-trained model Bert with external semantic knowledge and fused it with a CNN model for noun metaphor recognition.

BiLSTM (Bi-directional Long Short-Term Memory) is formed by combining forward LSTM with backward LSTM, which has stronger bi-directional semantic dependency capture capability, and also effectively prevents the gradient explosion and gradient disappearance problems of long text sequences by using gate mechanism and memory units. Jiaying Zhu [7] combined CNN and BiLSTM for metaphor recognition, in which CNNs with different windows were used to obtain semantic features in different ranges, and BiLSTM was used to concatenate local semantic features extracted by CNNs and obtain full-text features from them, and the model could recognize verb metaphors, noun metaphors, preposition metaphors, etc. Chuandong Su et al [8] based on Bert used five features for metaphor recognition to identify whether the sentence and the query words are metaphorically related.

Shenglong Zhang [10] used pre-trained language models to extract sentence semantic features, used Graph Convolutional Network (GCN) to extract syntactic features from syntactic structure information, and finally obtained classification results of sentence metaphors by splicing sentence semantic features and syntactic features classification, which achieved better recognition results on verb metaphors and noun metaphors.

Existing deep learning-based studies make full use of sentence sequence information or incorporate external semantic knowledge from pre-trained models, but do not make full use of the supporting role of sentence structure, and some studies have shown that word dependencies are subject-verb structures or verb-direct object structures, which are more likely to be expressed metaphorically [11]. This shows that syntactic structure is also an important feature for identifying metaphors.

In this paper, we propose a pre-trained language model as a word embedding layer for the recognition of metaphorical types of rhetoric in the public domain, and construct a connected graph using dependency syntactic analysis with nodes embedded in the corresponding word vectors, which enables the graph attention neural network to learn sentence semantic and syntactic structure information more fully and improve the effectiveness of metaphor recognition.

B. Bert Model

Bert [12] is a pre-trained model proposed by Google AI Research Institute in October 2018.Bert stacks multiple bi-directional Transformer structures and uses residual connections to address the limitations of traditional models in one direction and the problem of long-term dependencies. The use of the pre-trained language model Bert is divided into two phases: model pre-training and model fine-tuning.

In the model pre-training phase, there are two training tasks, namely word masking task and next sentence prediction task, through which the Bert model learns the relationships between words and lays the foundation for downstream tasks.

Bert's excellent performance and strong generalization performance make it widely used in the field of NLP, and many researchers have developed on the Bert model and then proposed improved models based on Bert, which has become one of the most commonly used techniques in the field of NLP.

The Bert model can replace word vectors, and in addition to transforming text into a computer-processable vector, it can further extract the deeper information embedded in the text. The Bert model introduces the relative position information between words, so it can effectively extract the correlation and difference between ontology and metaphor in metaphor [13].

C. Graphical Neural Networks

Graph neural network is an algorithm that fuses deep learning and graph structure, and its main idea is to use graph propagation mechanism and then update the features of nodes using deep learning. With stronger interpretability and good at processing graph structured data, graph neural networks are widely used in social systems, transportation networks, knowledge graphs and other fields.

In recent years, graph neural networks have received increasing attention, and the field of NLP has likewise introduced the study of graph neural networks. When graph neural networks are used for text processing, serialized text needs to be modeled first so as to introduce text structure information and then combined with downstream tasks to process text data using deep learning-based graph propagation algorithms. Graph neural networks use words as nodes and can introduce additional information to enrich features, such as dependency relationships [14], co-occurrence information between words [15], etc.

In the field of NLP, there are two common graph neural networks, GCN and GAT, in which the neighboring nodes of a node have the same weight, but the association between nodes usually has different importance, and GAT improves this by using the attention mechanism. The use of multi-headed attention also prevents overfitting the multi-headed attention mechanism and enhances the expressive power of the network [16].

The graph neural network model for metaphor recognition proposed in this paper introduces the dependent syntactic structure information between words in order to fully integrate the sentence structure information, and makes use of the powerful language structure modeling capability and structured feature extraction capability of GAT in order to learn the dependent syntactic information from the sentence dependent syntactic tree [17].

III. Metaphor Recognition Model BertGAT

The metaphor recognition model BertGAT proposed in this paper is divided into 4 layers, which are input layer, coding layer, graph attention layer and output layer, as shown in Fig. 1. First, the text is input to Bert and converted into word embedding information; then the sentence connectivity graph is constructed using LTP, feature extraction is performed using GAT, and finally the features are fed into the classifier to get the probability of each category. The coding layer and the graph attention layer are the core parts of the model.

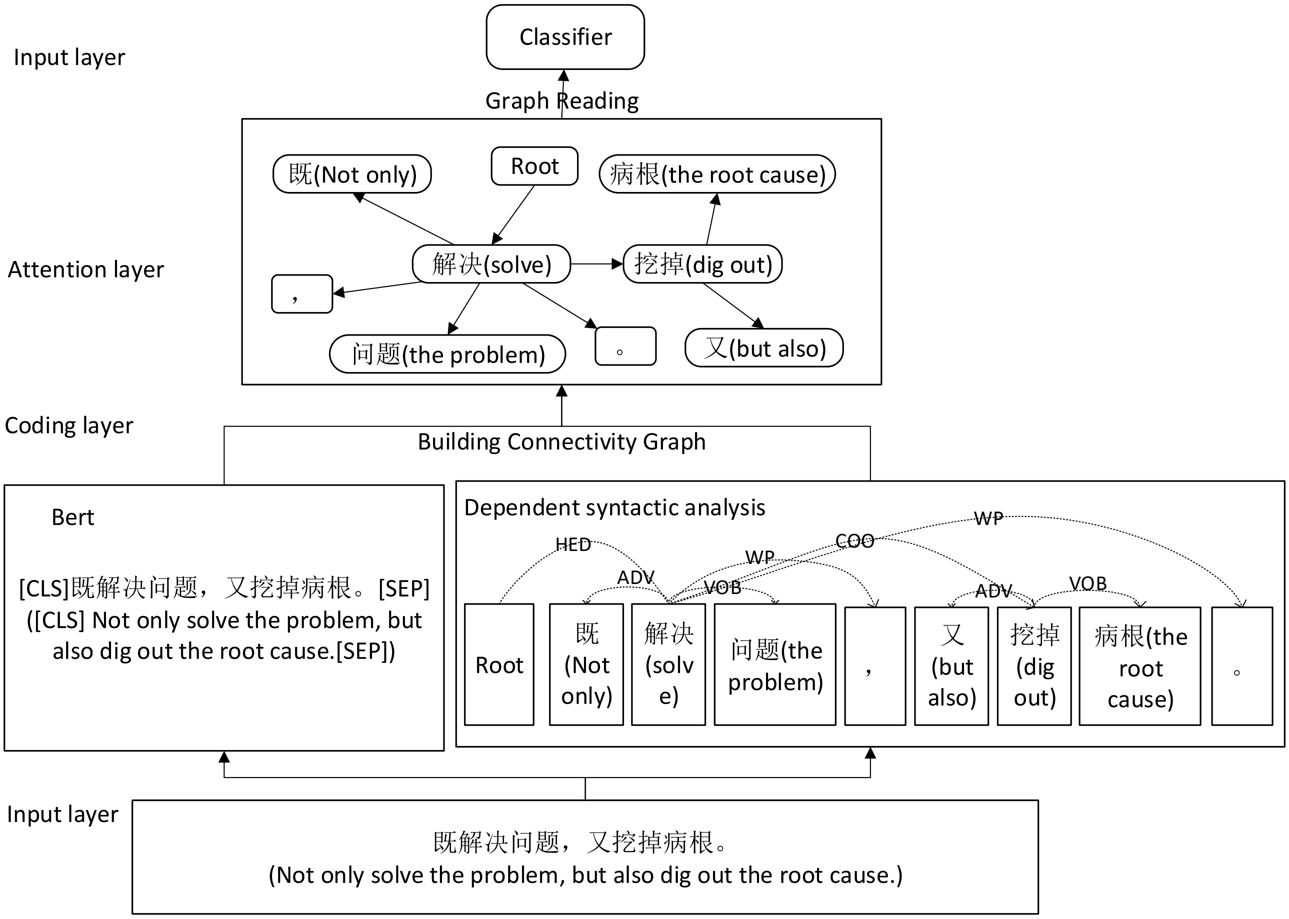

Fig. 1. Structure of BertGAT model.

The coding layer consists of Bert and the dependent syntactic analysis module. In the Bert pre-trained language model, sentences S in the text are sliced into word or character sequences, and after Bert processing, the output is a sequence of word vectors. It has been demonstrated that the dependency syntactic tree helps to improve the recognition rate of sentiment analysis [18][19] because it provides dependency connections between words, which leads the model to better learn the dependency information between long-distance words. In order for the model to fully learn the dependency information between words, the open source LTP tool of Harbin Institute of Technology is used for dependency syntactic analysis to obtain sentence dependencies and their clauses, and the word vectors belonging to the same clause are summed to construct a sentence connectivity graph by combining the dependencies, and the dependencies on each edge are embedded in low-dimensional vectors, and the graph nodes are embedded in word or character vectors.

The graph attention layer processes the connected graph of sentences to fully obtain the syntactic structure and semantic information of sentences, and finally outputs graph representation vectors as sentence features and input them into the classifier.

A. Input Layer

The role of the input layer is to perform data pre-processing in preparation for input to the encoding layer.

Before input to the Bert model, the sentence is cut into a sequence of words or characters, as shown in Equation (1).

$$S1=\{[CLS] , c_1, c_2, c_3,\cdots, c_n ,[SEP]\} \quad (1)$$

[CLS] and [SEP] are special identifiers for Bert, where [CLS] is added at the beginning of the sentence and [SEP] is added at the end of the sentence, and c_i is the character or word after the sentence is cut.

Chinese word separation is performed before input to the syntactic analysis module.

$$S2=\{ w_1, w_2, w_3,\cdots, w_n\} \quad (2)$$

w_i s the word after sentence segmentation.

B. Coding Layer

The encoding layer consists of the Bert module and the dependent syntactic analysis module. The pre-trained language model used in the Bert module is the original bert-base-chinese released by Google, containing 12 Transformer coding blocks, which are fine-tuned and trained together with the parameters of the attention layer to achieve metaphorical sentence recognition in Kumon.

Fig. 2 shows the model structure of Bert, which transforms the text sequence [CLS] , c_1, c_2, c_3, ..., c_n [SEP] into a word or character vector e_([CLS]), e_1, e_2, ..., e_n,e_[SEP] , and the final output is x_([CLS]) , x_1, x_2, x_3, ..., x_n , x_([SEP]), which is the vector generated by the encoder capturing the semantics of the multilayer Transformer structure.

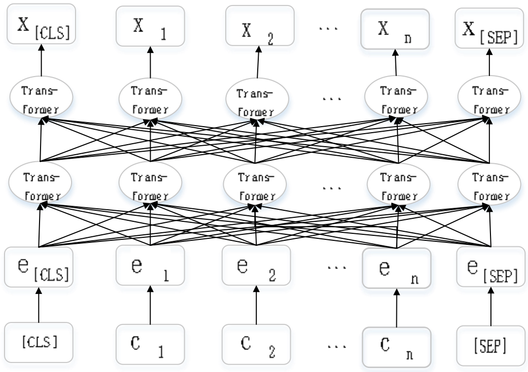

Fig. 2. Bert model structure.

This sequence S1 is input to the Bert model, and the Bert layer converts the text sequence S1 into the form of a vector and captures the semantics of each word or character using the Transform encoder to generate an embedded sequence X containing contextual semantic information, as shown in equation (3).

X={x_([CLS]) , x_1, x_2, x_3, ..., x_n , x_([SEP])}(3)

The output after the Bert module is a character-level vector embedding, while the generation of word-level vector embeddings is combined with LTP dependent syntactic analysis.

The LTP tool transforms the sentence sequence S2 into a triplet sequence T={t_1, t_2, t_3, ...t_n}. The triad t_j is shown as follows.

t_j=(g,q,d) (4)

where t_j is the jth set of dependency information, g is the dominant word, q is the modifier, and d is the dependency.

Finally, Bert is used to generate word embeddings, which are combined with LTP tool dependent syntactic analysis to generate a connected graph representation of the sentences.

The character-level vectors generated by the Bert pre-training model are represented as word vectors by summing the word vectors belonging to the same word.

C. Attention Layer

The role of this layer is to mine the sentiment association between word nodes and construct a connectivity graph based on a sequence of triples, where the nodes are word vectors or character vectors and the edges are dependencies represented by onehot. Based on the constructed sentence connectivity graph, the features between nodes are captured using graph attention network learning.

The following shows the node i at layer l. The similarity coefficients between it and its neighbor nodes are calculated one by one.

$$a_{ij}^l = \frac{\exp(f(a_l^T[W^l h_i^l||W^l h_j^l||d_{ij}]))}{\sum_{j\in n[i]} \exp(f(a_l^T[W^l h_i^l||W^l h_j^l||d_{ij}]))} \quad (5)$$

The above $\mathrm{a}_{\mathrm{ij}}^{\mathrm{l}}$ denotes the layer l, the weights of node i to neighbor node j. W^{l} is the weight matrix of the model trained at layer l. h_i^l is the word vector representation of node i at layerl. $n[i]$ is the set of neighbor nodes of node i. d_{ij} is the dependency between the i node and the j node.

The weights between the word nodes and the n neighbors can be calculated by equation (5).

$$\mathrm{a}_{\mathrm{i}}^{\mathrm{l}} = \{\mathrm{a}_{\mathrm{i1}}^{\mathrm{l}},\ \mathrm{a}_{\mathrm{i2}}^{\mathrm{l}},\ \ldots,\ \mathrm{a}_{\mathrm{in}}^{\mathrm{l}}\} \tag{6}$$

$\mathrm{a}_{\mathrm{i}}^{\mathrm{l}}$ identifies the importance level between word node i and its neighbor nodes, and combined with the node vector of the previous layer, the word node word vector of layer l+1 can be updated.

$$h_i^{l+1} = \sigma(\sum_{j\in n[i]} a_{ij}^l\, w^l h_j^l) \tag{7}$$

h_i^{l+1} is the word vector representation of point i at layer l+1, and σ is the ELU activation function. In a sentence, there are usually key words that determine the sentiment of the sentence, so the attention mechanism can be used to identify these key words and give them a greater weight. In the sentence "to solve the problem but also to dig out the root of the disease", the word "root of the disease" has a more obvious emotional tendency, so the weight value will be higher and the influence on neighboring nodes will be greater.

A study [11] pointed out that syntactic structure helps in metaphor recognition, so the graph constructed based on dependent syntactic analysis with dependencies can add syntactic structure information to enable the model to mine possible metaphorical expressions, in addition to directing the model to focus on important words.

The use of the multi-headed attention mechanism enables the model to mine information from multiple dimensions and prevents the model from overfitting, and the output after the introduction of the multi-headed attention mechanism is shown in the following equation.

$$\mathrm{h}_{\mathrm{i}'}^{\mathrm{l}+1} = \sigma(\frac{1}{\mathrm{K}}\sum_{\mathrm{k}=1}^{\mathrm{K}} \sum_{\mathrm{j}\in\mathrm{n[i]}} \mathrm{a}_{\mathrm{kij}}^{\mathrm{l}}\, \mathrm{w}_{\mathrm{k}}^{\mathrm{l}} \mathrm{h}_{\mathrm{j}}^{\mathrm{l}}) \tag{8}$$

$h_{i'}^{l+1}$is the feature vector of node i in this layer after feature extraction by the graph neural network with multi-head attention mechanism. K indicates that K attention heads are used; a_{kij}^l is the similarity coefficient between node i and node j after calculation by the attention mechanism of the k head; w_k^l is the weight matrix of the linear transformation of the input vector at the k head.

D. *Output Layer*

After the above feature extraction of graph attention neural network, the graph representation vector is obtained by reading out all graph node features using Sum. After the calculation of the multi-headed attention mechanism, the obtained graph representation vector is $R(H)$as in the following equation.

$$\mathrm{H}=\{\ \hbar_1, \hbar_2, \hbar_3, \ldots, \hbar_n\} \tag{9}$$

$$R(H) = \sum_{i=1}^{n} \hbar_i \tag{10}$$

H is the whole graph and $\hbar_i$ is the i node feature vector of the graph. $R(H)$ is the readout of the whole graph H using node Sum.

After obtaining the graph representation vector representing the sentence feature information, a fully connected softmax function is used to classify the sentiment, where the category with the highest probability is the sentiment category of the predicted text, as shown in the following formula:

$$p = softmax(wR(H) + b) \tag{11}$$

w is the weight matrix of the model training, b is the bias matrix, p is the prediction category of the model.

IV. Experiment

A. *Authors and Affiliations Data sets and Assessment Indicators*

This experiment is oriented to the recognition of sentences containing implicit sentiment in government official documents, for which there is no data available, so we need to construct a domain dataset. We search for metaphorical sentences commonly used in official documents in the Internet, among which are mainly noun metaphor sentences, and combine the verb metaphor and noun metaphor sentences in the Chinese metaphor recognition dataset released in the CCL 2018 Chinese Metaphor Recognition and Sentiment Analysis Task, a total of about 5000 sentences as positive examples; we use a crawler to crawl 30,000 official documents on government official document websites, and randomly select about 5000 sentences as counterexamples. Several sentences in the dataset are listed in Table I.

TABLE I. Example Table of Metaphorical Sentences

Text	**Category**	**Source**
切忌眉毛胡子一把抓，防止芝麻西瓜一起抱。 (Do not browbeat, to prevent the sesame seeds and watermelons together to hold.)	Noun Metaphor	Internet
经济转型中，政府必须起到催化剂的作用。 (The government must act as a catalyst in economic transformation.)	Noun Metaphor	CCL 2018
栽培社会所需要的集体主义精神。 (Cultivate the spirit of collectivism that society needs.)	Verb Metaphor	CCL 2018
中国青年是有深厚家国情怀的青年。 (Chinese youth are young people with a deep sense of family and country.)	Literary	Internet

In the first sentence of Table I, eyebrows and beards, sesame seeds and watermelons are metaphors for priorities and sub-priorities in work, which are noun metaphors. These sentences are often used in speeches of leaders to promote their policies, ideas and work, but not in strictly official documents. The second sentence is a noun metaphor, comparing the government to a catalyst. The third sentence is a verb metaphor, where the verb "cultivate" refers to planting and nurturing, but the sentence refers to cultivating.

To further validate the model for sentiment recognition, the public dataset ChnSentiCorp [20] was used to further validate the model effect.

The training set, validation set and test set were assigned in the ratio of 8:1:1.

The result evaluation metrics used recall, precision, accuracy and F1 value formulas as shown in (12) to (15) below.

$$\text{recall} = \frac{\text{TP}}{\text{TP+FN}} \times 100\% \tag{12}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP+FP}} \times 100\% \tag{13}$$

$$\text{F1} = 2 \times \frac{\text{precision}\times\text{recall}}{(\text{precision}+\text{recall})} \times 100\% \tag{14}$$

$$\text{accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \times 100\% \tag{15}$$

B. Parameter Setting and Experimental Environment

The experimental parameters of the BertGAT model proposed in this paper are set as follows:

1) Experimental environment: Using deep learning framework pytorch, operating system ubuntuServer 18.04.5, GPU is Tesla P100, memory 60G, programming language python3.8.

2) Parameter setting in Bert: Bert uses the original Bert-base-Chinese released by Google, the dropout is set to 0.3, the optimizer uses AdamW, the maximum number of individual text words is 300, the initial learning rate is set to 5e-5, the number of iterations is 40, and the Batch Size is set to 16.

3) Parameter settings in the GAT: In order to avoid the potential problem of over-smoothing the nodes of the graph neural network, the experiment of GAT layer setting is conducted in this paper. From the experimental results, we can know that the model effect gradually decreases when the number of GAT layers is greater than two, and the model achieves the optimal effect at two layers, see Fig. 3.

4) Graph reading methods in GAT: There are three methods to calculate the graph representation, Sum, Mean and Max respectively, where Sum is to sum the nodes, Mean is to average the nodes and Max is to take the maximum value of the nodes.

In order to determine the optimal graph readout method and training learning rate of the model, the following experiments are set up in this paper, from the experimental results, we can see that the graph readout method is optimal using node Sum, and the learning rate is optimal using 5e-5 for fine-tuning, see Table II and Table III.

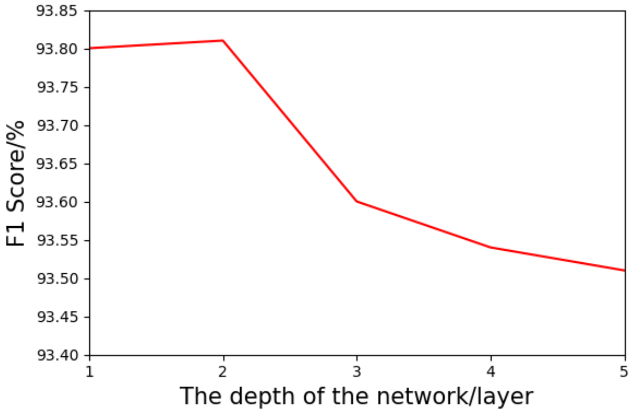

Fig. 3. Experimental results of graph attention neural networks with different layers.

TABLE II. WAYS OF READING GRAPH

Graph reading mode	**Indicators**			
	Acc	***Precision***	***Recall***	***F1***
Max	0.9379	0.9377	0.9387	0.9379
Sum	**0.9383**	**0.9380**	**0.9384**	**0.9381**
Mean	0.9352	0.9349	0.9352	0.9351

TABLE III. LEARNING RATE SETTINGS

Learning Rate	**Indicators**			
	Acc	***Precision***	***Recall***	***F1***
5e-4	0.9366	0.9362	0.9368	0.9364
5e-5	**0.9383**	**0.9380**	**0.9384**	**0.9381**
5e-6	0.9355	0.9354	0.9365	0.9355

C. Experimental Results

CNN, BiLSTM and BertGCN are chosen as baseline models. CNN, BiLSTM and other models use word2vec trained from wiki Chinese corpus as word vector, the maximum number of single text words is 300, the dimension of word vector is 200, and different sizes of convolutional kernels are set in the convolutional layer to extract different features from different dimensions. Three sizes of convolutional kernels are set with the size of 3* 200, 4* 200 and 5* 200 respectively.

The baseline experimental results are shown in Tables IV and V below, and it can be seen that the proposed method in this paper achieves the best performance in both the constructed public domain dataset and the public sentiment dataset. From the experimental results, it can be seen that the performance of the model with the introduction of graph neural network in this paper has been improved, where the biggest difference between GCN and GAT is that the way of aggregating neighbor nodes is different, and GAT utilizes attention coefficient, so the correlation between nodes calculated by GAT will be stronger to some extent. The experimental results show that the comparison of the experimental results processed using GAT is more advantageous than GCN, which indicates that GAT is better than GCN in learning the structural information of the sentences.

TABLE IV. METAPHOR RECOGNITION RESULTS OF DIFFERENT MODELS IN DOMAIN DATASETS

Model	**Indicators**			
	Acc	***Precision***	***Recall***	***F1***
CNN	0.8839	0. 8845	0.8853	0.8838
BiLSTM	0.9111	0.9109	0.9118	0.9110
Bert	0.9349	0.9346	0.9355	0.9348
BertGCN	0.9366	0.9367	0.9361	0.9364
BertGAT	**0.9383**	**0.9380**	**0.9384**	**0.9381**

The results of the experiments on the dataset ChnSentiCorp are shown in Table V. In this paper, ChnSentiCorp-6000 is selected for the experiments, and there are 3000 positive and negative sentiment sentences each in this dataset. In this public dataset experiment, model CNN, BiLSTM, AttBiLSTM, DCNN, ADCNN, and experimental result data are used from Zhu Ye et al [21], through the experiment, it can be seen that the sentiment recognition effect of BertGAT model is better than the baseline model in Precision, Recall and F1, and comparing Bert has a significant improvement, which indicates that the model proposed in this paper is advanced in emotion recognition.

TABLE V. EMOTION RECOGNITION EFFECT OF DIFFERENT MODELS IN THE PUBLIC DATASET CHNSENTICORP-6000

Model	**Indicators**			
	Acc	***Precision***	***Recall***	***F1***
CNN	0.8550	0.8557	0.8550	0.8553
BiLSTM	0.8790	0.8796	0.8790	0.8793
AttBiLSTM	0.8810	0.8827	0.8816	0.8821
DCNN	0.8865	0.8865	0.8865	0.8865
ADCNN	0.8897	0.8898	0.8895	0.8896
Bert	0.8893	0.8891	0.8897	0.8893
BertGCN	0.8850	0.8898	0.8827	0.8840
BertGAT	**0.9035**	**0.9045**	**0.9030**	**0.9033**

The main reason why the metaphor recognition model BertGAT proposed in this paper is ahead of other comparative models is the introduction of syntactic structure information and GAT graph attention network, which is based on GCN graph convolutional network and introduces the attention idea. GAT can avoid a large number of matrix operations compared with GCN, and it can also obtain the node feature information with stronger dependency.

V. CONCLUSION

In this paper, we propose an implicit sentiment classification model BertGAT for rhetorical types. To fully capture the semantic and syntactic structural information in text sequences, the model uses the Bert pre-trained language model and the GAT graph attention network. To verify the effectiveness of the model, experiments were conducted on both domain datasets and public datasets, and the experimental results outperformed the baseline model, which demonstrated the effectiveness of the model.

It is difficult to satisfy all downstream tasks by simple metaphor recognition classification alone. Metaphorical sentences contain mappings from ontology to metaphor, and our further research hopes to extract ontology, mapping words and their metaphors with the help of advanced Chinese named entity recognition techniques [22]. Besides, the different cognitive perspectives and cultural value orientations of English and Chinese can lead to different semantics of the same metaphorical word, and the syntactic structures of English and Chinese, two different languages, are also different [23].

we're actively engaged in follow-up this work. The next step will be to identify the similarities and differences between English metaphorical sentences and Chinese metaphorical sentences from several perspectives in order to improve the model to make it more generalizable.

REFERENCES

[1] Krishnakumaran S, Zhu X. Hunting elusive metaphors using lexical resources[C]//Proceedings of the Workshop on Computational approaches to Figurative Language. 2007: 13-20.

[2] Su Chang, Fu Ze, Zheng Fakui, et al. Metaphor Recognition Method Based on Dynamic Classification [J]. Journal of Software, 2019, 30(11): 3340-3354.

[3] Su Chang, Wang Xiaomei, Huang Shuman, Chen Yijiang. An Approach to Metaphor Comprehension Based on Relevance Constraints [J]. Journal of Software,2017,28(12):3167-3182.

[4] Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014.

[5] Li Bin, YU Lili, Shi Min, et al. Simile Calculation of Image [J]. Journal of Chinese Information Processing, 2008, 22(6): 27-32.

[6] Shutova E, Sun L, Korhonen A. Metaphor identification using verb and noun clustering[C]//Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010). 2010: 1002-1010.

[7] Zhu Jiaying, Wang Rongbo, Huang Xiaoxi, et al. Multilevel Metaphor Recognition Method Based on Bi-LSTM [J].. Journal of Dalian University of Technology/Dalian Ligong Daxue Xuebao, 2020, 60(2):209-215.

[8] Su C, Fukumoto F, Huang X, et al. DeepMet: A reading comprehension paradigm for token-level metaphor detection[C]//Proceedings of the second workshop on figurative language processing. 2020: 30-39.

[9] Su Kuilin, Zhang Kai, Lv Xueqiang, Zhang Le. Noun Metaphor Recognition Based on Fusion Model [J]. Computer Technology and Development,2022,32(06):192-197.

[10] Zhang S, Liu Y, Ma Y. SaGE: Syntax-aware GCN with ELECTRA for Chinese Metaphor Detection[C]//Proceedings of the 20th Chinese National Conference on Computational Linguistics. 2021: 667-677.

[11] EKATERINA S，SIMONE T，ANNA K.Statistical metaphor processing[J].Computational Lingus，2013，39（2）：301-353.

[12] Kenton J D M W C, Toutanova L K. Bert: Pre-training of deep bidirectional transformers for language understanding[C]//Proceedings of naacL-HLT. 2019: 4171-4186.

[13] Zhang Dongyu, Cui Zijuan, Li Yingxia, Zhang Wei, Lin Hongfei. Noun Metaphor Recognition Based on Transformer and BERT [J]. Data Analysis and Knowledge Discovery,2020,4(04):100-108.

[14] Tian Y， Chen G， Song Y，et al. Dependency-driven Relation Extraction with Attentive Graph Convolutional Networks[C]// 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). Online: Association for Computational Linguistics. 2021: 4458–4471.

[15] Caren Han S, Yuan Z, Wang K, et al. Understanding Graph Convolutional Networks for Text Classification[J]. arXiv e-prints, 2022: arXiv: 2203.16060.

[16] Wu Bo, Liang Xun, Zhang Shusen, et al. Advances and Applications of Graph Neural Networks [J]. Chinese Journal of Computing, 2022, 45(1).

[17] Ji T, Wu Y, Lan M. Graph-based dependency parsing with graph neural networks[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 2475-2485.

[18] Tian Y, Chen G, Song Y. Enhancing aspect-level sentiment analysis with word dependencies[C]//Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume. 2021: 3726-3739.

[19] Fu Chaoyan, Huang Xianying, Liu Hankai, Qi Songzhe. Syntactic feature fusion enhancement aspect level sentiment analysis [J/OL]. Journal of Chinese Mini-Micro Computer Systems:1-10[2022-10-05].http://kns.cnki.net/kcms/detail/21.1106.TP.20220215.1046.021.html

[20] TAN S, ZHANG J. An empirical study of sentiment analysis for Chinese documents[J]. Expert Systems with Applications, 2008, 34(4): 2622-2629.

[21] Zhu Ye, Chen Shiping. A Novel approach to Emotion Analysis Based on Convolutional Neural Networks [J]. Journal of Chinese Mini-Micro Computer Systems,2020,41(03):551-557.

[22] Deng Yiyi, Wu Changxing, Wei Yongfeng, Wan Zhongbao, Huang Zhaohua. A Review of named Entity Recognition Based on Deep Learning [J]. Journal of Chinese Information Processing,2021,35(09):30-45.

[23] Liu Xiaoyu, Liu Yongbing. A Comparative Study of "Big/small" Spatial Metaphors in English and Chinese from the Perspective of Cognition [J]. Journal of PLA University of Foreign Languages ,2022,45(02):17-24.