



Full length article

Social network sentiment classification method combined Chinese text syntax with graph convolutional neural network



Xiaoyang Liu*, Ting Tang, Nan Ding

School of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China

ARTICLE INFO

Article history:

Received 12 July 2020

Revised 1 March 2021

Accepted 19 April 2021

Available online 30 April 2021

Keywords:

Sentiment classification

Graph convolutional neural network

Text syntax

Word embedding

ABSTRACT

In view of most current studies on text sentiment classification focus on the deep learning model to obtain the sentimental characteristics of English text. Chinese text sentiment analysis is rarely involved, and only the context information of the statement is considered, but the syntax information of the statement is rarely considered. In this paper, a novel sentiment classification model is proposed (Dependency Tree Graph Convolutional Network, DTGCN) combined Chinese syntactically dependent tree with graph convolution. Firstly, the Bi-GRU (Bi-directional Gated Recurrent Unit) model is used to learn the contextual feature representation of a given text. Secondly, the syntax-dependent tree structure of a given text is constructed, then obtain its adjacency matrix according to the syntax-dependent tree, with the initial features extracted from the bidirectional gate control network, input into the graph convolutional neural network (GCN) to extract the sentimental features of the text; the obtained sentimental characteristics are then input into the classifier SoftMax for text sentimental polarity classification. Finally, the data set is compared with the mainstream neural network model. The experimental results show that the accuracy of the proposed DTGCN model proposed on the data set is 90.51% and the recall rate is 90.34%. Compared with the benchmark models (LSTM, CNN, TextCNN and Bi-GRU), the proposed DTGCN model shows a 4.45% advantage in accuracy. It shows that the proposed DTGCN model can effectively use the grammatical information of Chinese text to mine the hidden relationship in statements, it can improve the accuracy of Chinese text sentiment classification. In addition, the proposed DTGCN model not only improves the performance of sentiment classification in the essay, it also provides a new research method for social network public opinion identification.

© 2022 THE AUTHORS. Published by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

1.1. Background

Sentiment Analysis (SA) has been a hot topic in the field of Natural Language Processing (NLP) as well as a research hotspot in recent years. Sentiment analysis is a process of automatically analyzing the text and explaining the emotions behind it. Through

machine learning and text analysis algorithm, text can be classified into positive, negative and neutral categories according to sentence emotion. With the popularization of mobile Internet, network users have been accustomed to express their opinions and Suggestions on the Internet, such as the evaluation of commodities on e-commerce websites, social media reviews of brands, products, policies, etc. How web users perceive and feel the real world, any choice they make is influenced by how other people perceive the real world. Therefore, it will excavate and analyze the opinions and feelings expressed by netizens on the things they are interested in, and apply the research results to public opinion analysis, market research, customer experience analysis and other fields, this is the research significance of emotion analysis.

Text sentiment analysis technology mainly studies how to automatically analyze subjective information such as opinions, feelings, positions and attitudes expressed in texts. Identify people's views on a product or event from massive texts to improve the efficiency of emotional analysis of texts. According to the granularity of text

* Corresponding author.

E-mail address: lx3103@163.com (X. Liu).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.



Production and hosting by Elsevier

processing, emotion analysis can be divided into three levels: word emotion analysis, sentence emotion analysis and document emotion analysis.

The object of word emotion analysis is the words and phrases that appear in a particular sentence. Words that express emotions are mostly nouns, verbs, the emotional tendency of adverbs and adjectives can be divided into three categories: positive, negative and neutral. Emotional intensity and contextual patterns are analyzed. The object of sentence emotion analysis is the sentence appearing in the specific context. Its purpose is to analyze various subjective information in the sentence. Each element related to emotional tendency is extracted to realize the judgment of sentence emotional tendency. The object of emotional analysis of a document is a complete article, analyzing the emotional tendency of an article as a whole. Since document emotion analysis is a text classification problem, machine learning is usually used to solve the problem of text emotion analysis. This paper is mainly aimed at the short text of social network platform, so the work of this paper belongs to sentence emotion analysis. The difficulty in the study of sentence emotion analysis lies in the fact that any artificially generated sentence contains many layers of meaning. People express their opinions in complex ways, and rhetoric such as sarcasm, sarcasm, and implied meaning can influence the end result of emotional analysis.

Most of the existing sentence emotion analysis methods are based on deep learning and encode the short text as a whole. Ignoring the syntactic structure of the sentence and using only the semantic information of adjacent words, a context-aware representation of a specific word without the ability to determine the emotion described by non-adjacent words. In recent years, studies have shown that [1–3] can effectively capture emotional features through syntactic information in syntactically dependent tree dependent sentences. For example, Luo H [4] proposed a bidirectional dependency tree network, explicit merging of two representations obtained separately from bottom-up and top-down propagation solves the aspect extraction problem. Ahmed M [5] designed a general attention framework for syntactically dependent trees and phrase structure trees. Zhang J [6] established an interaction model along the dependent arc through the graph neural network. Using grammar-based self-attention, the syntax diagram can be manipulated directly to obtain structural information. In addition to emotion analysis, syntactically dependent trees also make neural networks greatly improved in NLP tasks such as natural language generation [7] and Dialogue Understanding [8].

Because of the special structure of the tree, in this paper, Graph Convolutional Networks (GCN) is introduced to capture the high-level semantic information. Graph convolution was first proposed to solve non-Euclidean structural data in social networks, knowledge graphs and information networks. At present, it is widely used in image detection [9], user recommendation [10] and image classification [11]. Wan S [12] proposed that GCN network is used for text classification. The graph is constructed on the whole corpus set, the words and the number of articles are taken as nodes in the graph, and the edge in the graph is constructed by using co-occurrence information. Then the text classification problem is regarded as the node classification problem, and good results have been achieved [13]. In order to integrate grammatical information and make full use of syntactic structures in sentences, a syntactical dependence tree based GCN model is proposed to detect emotional polarity. First, it covers a Bi-GRU for feature extraction of each word; then, the word features and the adjacency matrix dependent on the analytic tree are fused into GCN, and the features are fused through the maximum pooling layer. Finally, the experiment results can be obtained through SoftMax function.

1.2. Our main contributions

The main contributions of this paper are as follows:

- (1) A syntactically dependent tree is constructed for a given Chinese text, and the syntactic dependent tree is used to analyze the grammatical relations among words in the text of the data set, hidden relationships between words are mined.
- (2) A novel graph convolutional network sentiment classification model (DTGRU) based on Chinese syntax-dependent tree is proposed. This model combines the preliminary features extracted by Bi-GRU with the syntactic tree of the sentence, and then convolves with GCN. To obtain the comprehensive characteristics of the text, so as to realize the emotional classification of Chinese text.
- (3) Proposed DTGRU model is compared with six baseline models (CNN, LSTM, CNN + LSTM, TextCNN, Bi-GRU + GCN). *Accuracy* (ACC), precision *P*, macro average *F1* value of (*Macro_F1*) and recall rate (*Recall*) are used as experimental evaluation indexes. Through comparison experiments, the accuracy of DTGRU model and Macro_F1 value are improved by 4.45% and 3.71% compared with the benchmark model. It verifies the importance of using grammatical information and long-term word dependence, and proves the validity of DTGRU model in sentiment classification.

2. Related work

2.1. Deep learning for sentiment classification

At present, there are three kinds of analytical techniques in emotion analysis: rule-based method, machine-learning method and deep learning method. The rule-based method uses a set of artificial rules to help identify sentence emotional polarity. For example, define the word list of positive polarization and negative polarization and calculate the number of polarity in a given text. If the number of positively polarized words is greater than the number of negatively polarized words, the algorithm will return the judgment result of positive polarization, and vice versa. If the number of occurrences of the two polarities is the same, a neutral judgment is returned. The principle of this method is relatively simple, however, the process is tedious and the combination mode in the word sentence sequence is not considered, and the update of part of speech list is complex, which needs constant maintenance.

In machine learn-based approaches, the sentiment analysis task is typically modeled as a classification in Fig. 1.

During the training, the model learns from the test samples used in the training, associating specific inputs (that is, text) with corresponding outputs (labels). An eigenvector converts the input text into an eigenvector. Pairs of eigenvectors and markers, input machine learning algorithms (e.g., Naive Bayes [1], SVM [14] (Support Vector Machines) and LR [14] (Logistic Regression)) are used to generate models. During the prediction process, feature extractors convert text inputs into feature vectors, which are then input into the model, generate a prediction label (such as positive, negative, or neutral).

The approach based on deep learning is divided into two steps: firstly, critique corpora that need to be classified are represented as semantic Word vector; secondly, different semantic synthesis methods are used to obtain the corresponding characteristic expressions of sentences or documents, finally, it is classified by deep neural network [14]. Currently, the deep learning models commonly used in emotion analysis include Convolutional Neural Networks (CNN) [15,16], Recurrent Neural Networks (RNN) [17], Long-term Memory neural network (LSTM) [18].

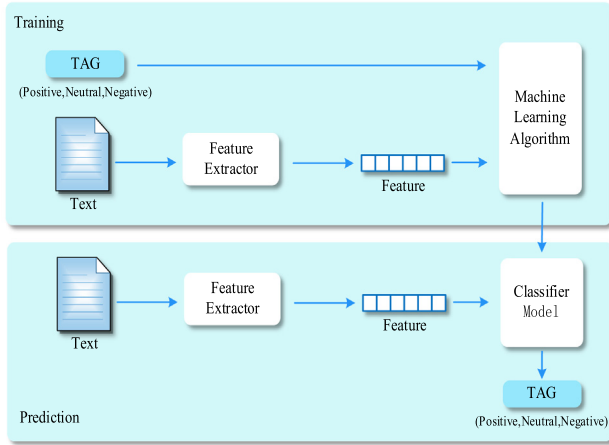


Fig. 1. Machine learning classifier.

Kim [19] used convolutional neural network (CNN) to conduct a series of experiments on sentence level text classification with pre-trained word vectors. Thus, it shows that a simple CNN can achieve excellent results in multiple benchmarks with a small amount of super parameter adjustment and static vector. Makoto [20] proposed the emotional classification of customer comments by combining space pyramid pool with Max Pooling and using gated CNN. Meng [21] proposed a transfer learning method based on multi-layer convolutional neural network. Extracting features from the source domain, the weights are Shared in the convolutional layer and pooling layer between the source domain and target domain samples.

In addition to using CNN single, some researchers also combine CNN with RNN to achieve better classification effect. Jiang [22] combined LSTM and CNN, and utilized the ability of LSTM to deal with remote dependency and CNN to recognize local characteristics. Features extracted by the LSTM will be filtered again through convolution and pooling operations to find important local features. Luo [23] proposed that CNN with gated cycle unit could be used as classifier. According to the input characteristic matrix, GRU-CNN enhances word-to-word, text-to-text relationships to achieve high-precision classification of emotions. Abid [24] takes word embedding as the input of deep neural structure, three cyclic neural networks, namely Bi-LSTM, GRU and Bi-GRU, which are variants of RNN model, are used to capture long-term dependence.

The above work does not use dependency resolution trees to train the deep learning network and ignores the syntactic information of the text. The dependency analytic tree can reveal the synonym structure of sentence more accurately and clearly. Therefore, the combination of Bi-GRU model and graphic convolutional neural network in this paper can enhance the emotional understanding of the short text.

2.2. GRU

In order to alleviate the long-distance dependence problem of traditional RNN and its gradient disappearance and gradient explosion in the back propagation process, based on RNN, LSTM adds input gate, forgetting gate and output gate to control input value, memory value and output value. In this way, the network can selectively discard or retain historical information. GRU is a further improvement on LSTM. GRU replaces forgetting gate and input gate in LSTM with updating gate. Due to fewer GRU parameters, the code is easier to modify and maintain, and the calculation amount in the process of training network parameters is greatly

reduced, so the calculation efficiency is higher. Shorter training time is required.

In a one-way neural network structure, states are always output from front to back. The internal structure of a single time step of the GRU model is shown in Fig. 2.

In Fig. 2, r is to reset gate and determine whether to ignore the hidden state h^{t-1} of the previous time step. If the activation weight goes down, the current time step captures more dependency information of the short time span. When the reset door activation approaches 0, the input x^t of the current time step will affect the output of the implicit state h' to a greater extent, thus forgetting the result of the dependence of the long-time span. Update gate z selects whether to update the output hidden state h^t with the new candidate hidden state h' . Each time step of GRU model has reset gate r and update gate z respectively. The hidden nodes of each time step will learn to capture the characteristics of different dependent information under different time spans. In the classification of textual emotions, if the output of the current moment can be related to the state of the previous moment and the state of the next moment, just like a fill-in-the-blank, infer the word in the blank from the context of the blank. In this case, Bi-GRU is needed to establish this connection.

Bi-GRU is a neural network model consisting of a unidirectional, opposite direction GRU whose output is jointly determined by the two GRU states. At each moment, the input provides two opposite GRUs, and the output is determined by both one-way GRUs. Table 1 shows the research comparison of GRU research papers in the literature survey.

2.3. Gcn

The basic idea of graph convolution network is to generate node embedding based on the information of neighboring nodes, and each node and neighboring nodes generate one to compute the graph. The graph-based structure can generate any multi-layer, where the information of the node can be passed in with the feature of the node itself.

Fig. 3 shows a simple graph convolution example, x_i represents the node to be studied and x_j represents a node in the neighbor of x_i , and A_{ij} represents the connection between nodes. We can study the characteristics of x_i by studying the neighboring nodes of x_i , as shown in Equation (1):

$$\text{summation}(x_i) = \sum_{j \in \text{Neighbor}} A_{ij} x_j \quad (1)$$

At the same time, we have to consider the characteristics of the node itself, so we can add back its own characteristics by adding a self-loop to get the Equation (2)

$$\text{summation}(x_i) = \sum_{j \in \text{Neighbor}} A_{ij} x_j + x_i \quad (2)$$

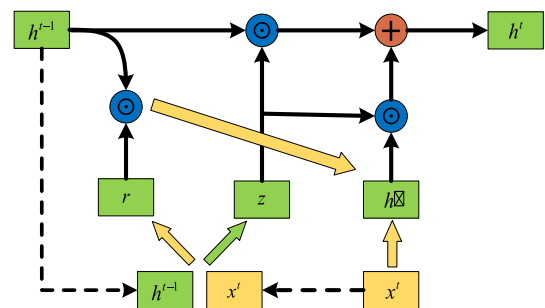


Fig. 2. Internal structure of GRU.

Table 1
Comparison of GRU research status.

No.	Year	Study	Research	Method	Target
[24]	2018	Minh et al.	By studying the influence of hidden emotion of financial news on investors, the language data set of finance specialty was established, and finally the stock price trend was predicted	Lexicon, StockTwits, Bi-GRU	Aspect-based sentiment analysis
[25]	2018	Hongo et al.	Improved the weight of words that have a particular effect on the sentence by modeling	Bi-GRU, novel attention pooling	Sentiment classification, opinion extraction, fine-grained sentiment analysis
[26]	2018	Bleiweiss et al.	Categorize sentences that describe the emotions and emotions usually expressed by flowers, thus implicitly predicting the type of bouquet	GRU and CNN	Extract features from user verbal information
[27]	2019	Chen et al.	Use GRN to model emotional relationships such as negation and reinforcement in a recursive architecture	Lexicon, LSTM and GRU	Aim to improve accuracy
[28]	2019	Lee et al.	Use GRU to get a fine-grained classification of Chinese product reviews	GRU	Enhance the effectiveness and robustness of the GRU
[29]	2019	Yu et al.	The in-depth exploration of online text emotional information is used for online opinion discovery	Attention, Bi-GRU	To extract the polarity from the data
[30]	2020	Sachin et al.	Sentiment-/semantic-based approaches for sentiment detection	LSTM, GRU	Combining sentiment and semantic features
[31]	2020	Fagui et al.	Long text modeling and exploring the semantics between sentences and the dependencies between elements	Attention, 2DCNN, Bi-GRU	Studying the possibility of leveraging the contribution of different filter lengths and grasping their potential in the final polarity of the sentence

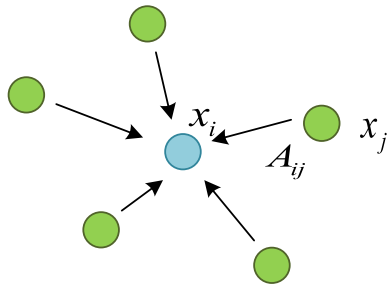


Fig. 3. An example of GCN.

To sum up, although there is a growing body of research on ways to identify positive and negative emotions about a particular topic from online texts. However, most of these studies focus on sentiment analysis of English texts, while the field of Chinese sentiment analysis is still in its infancy. Therefore, in order to learn more about the emotional characteristics of Chinese sentences and the hidden information in sentence grammar, a graph convolution network based on syntactic dependency tree is proposed. On the one hand, syntactic dependency trees are used to aggregate syntactic information into the representation of context and aspect words. On the other hand, the preliminary features of sentences are extracted through the Bi-GRU network and embedded into the syntactic dependency tree. Finally, the syntax-dependent tree input graph is convolved with the network to obtain the final emotional characteristics.

3. The proposed method

In social networks, most short texts have a variety of emotional tendencies, while some texts express no emotions at all. In practical applications such as psychological research or user emotional portraits, emotional analysis of social networks is necessary. The proposed DTGCN network model designed in this paper consists of five layers. ① Input layer: mainly responsible for some pretreatment of the sentence; ② Bi-GRU: Use bidirectional GRU layer to extract hidden semantics of text; ③ Grammar layer: by analyzing the grammatical relations in the sentence, gets the syntax-dependent tree; ④ Graph Convolutional layer: embed the feature

vectors extracted from the coding layer into the nodes of the syntax-dependent tree. The graph convolution network operates directly on the syntax-dependent tree of text to obtain the emotional characteristics of a given text; ⑤ Output layer: the Softmax layer is used to predict the corresponding emotional polarity according to the output result of graph convolution, and the output will be classified at last.

Get the final result.

The structure of sentiment classification model DTGCN is convolved with the graph of syntax-dependent tree, as shown in Fig. 4.

3.1. Input layer

a) Normalization

Most of the text in social networks is network text, in which there are a lot of useless value information, such as “~@#¥%. . . . &””, special characters. Meaningful information will not be added to the Chinese emotion analysis model, the addition of these special characters will complicate the result and is not conducive to sentence segmentation, so regular expressions can be used to remove invalid characters.

Since the length of the network text is not fixed, in order to avoid inconsistent representation in the dot product similarity calculation, the variation dimension of emotion clause is reconstructed into the characteristic vector of uniform size, that is, the sentence length is less than the specified value. By default, special symbols are automatically filled in the back (fill 0 for this article); if the length of the sentence is greater than the specified value, the part greater than the specified value is truncated. The sentence length distribution of the data set in this paper is shown in Fig. 5. As can be seen from Fig. 5, very few sentences are longer than 170. Few sentences are longer than 200, and most of them are within 150, which is in line with the shorter Chinese text on the social network platform. Among them, 140 or so sentences appear most frequently, so the rated value of this paper is 140, that is, when the sentence length is greater than 140, The excess will be eliminated from the model training.

b) Tokenization

Tokenization of word segmentation or raw text is a standard preprocessing step for many NLP tasks. For English, particles

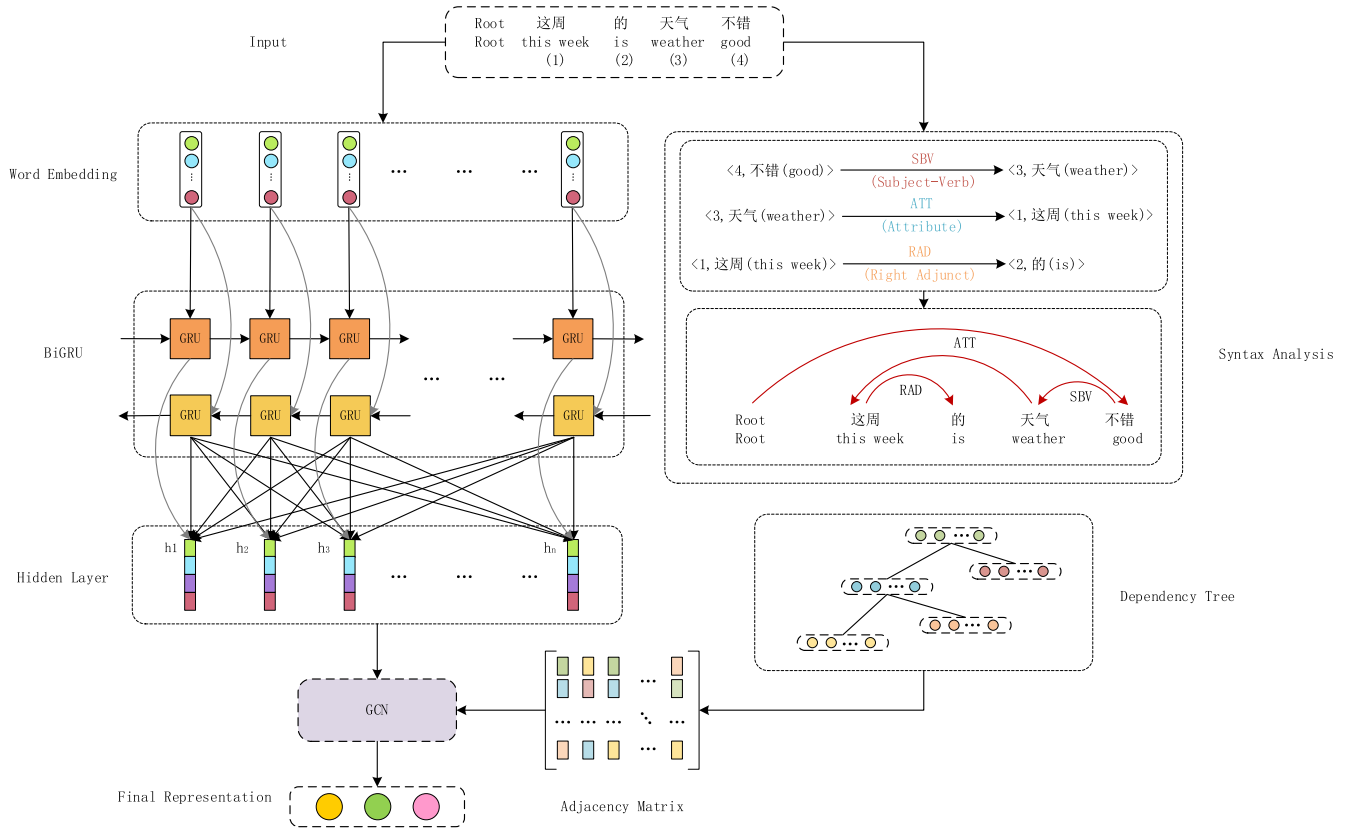


Fig. 4. Proposed DTGCN model.

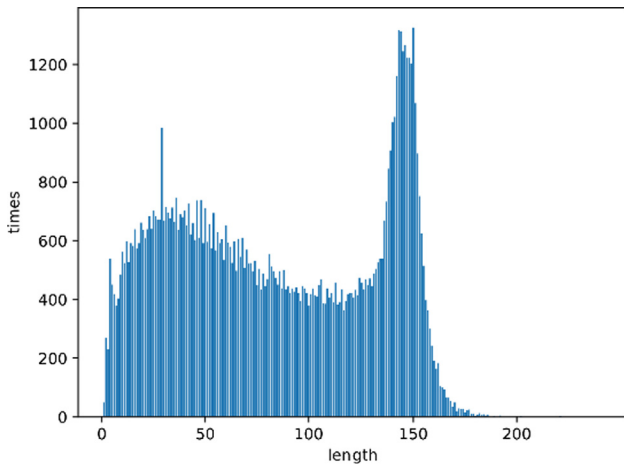


Fig. 5. Sentence length distribution.

can usually be broken up with Spaces. For Chinese, Chinese texts have some characteristics that are different from English texts.

That is, the writing style is continuous, with no spaces. Therefore, according to the word segmentation standard, the steps of dividing Chinese text into a series of words are defined. This article selects Jieba, a Python package that deals with Chinese particples. It works by first tagging individual characters, then connect the tags with Spaces before returning the complete sentence.

c) Embedding

Sentences belong to high-level cognitive abstract entities produced in human cognitive process. Most machine learning models,

including neural networks, cannot process text in its original form and require numbers as input. Therefore, to convert the characters in the dictionary into continuous vectors, word embedding must be carried out, namely: embed a higher-dimensional space (tens of thousands of words, hundreds of thousands of words) whose dimension is the number of all words into a much lower dimensional continuous vector space. Each word or phrase is mapped to a vector in the real field. In this paper, the method of random initialization is used to represent each word with a random vector, and the vector is updated during the network training.

After the above three steps, the input data becomes a word matrix that corresponds to the word vector in terms of the index.

3.2. Bi-GRU layer

The main task of the coding layer is to use Bi-GRU to learn the context information of sentences and attribute clauses respectively, and obtain corresponding feature representation. The structure of Bi-GRU is shown in Fig. 6.

For the forward propagation algorithm of Bi-GRU; firstly, the input x_t of the current time step and the hidden state output \vec{h}_{t-1} of the last time step can be used to calculate the original pre-feedback output \hat{z}_t :

$$\hat{z}_t = W_{zx}x_t + W_{zh}h_{t-1} + b_z \quad (3)$$

Then the output result is divided into two parts, namely $\hat{z}_{t,H1}$ and $\hat{z}_{t,H2}$, which are activated respectively:

$$r_t = \sigma(\hat{z}_{t,H1})(\text{reset gate}) \quad (4)$$

$$z_t = \sigma(\hat{z}_{t,H2})(\text{update gate}) \quad (5)$$

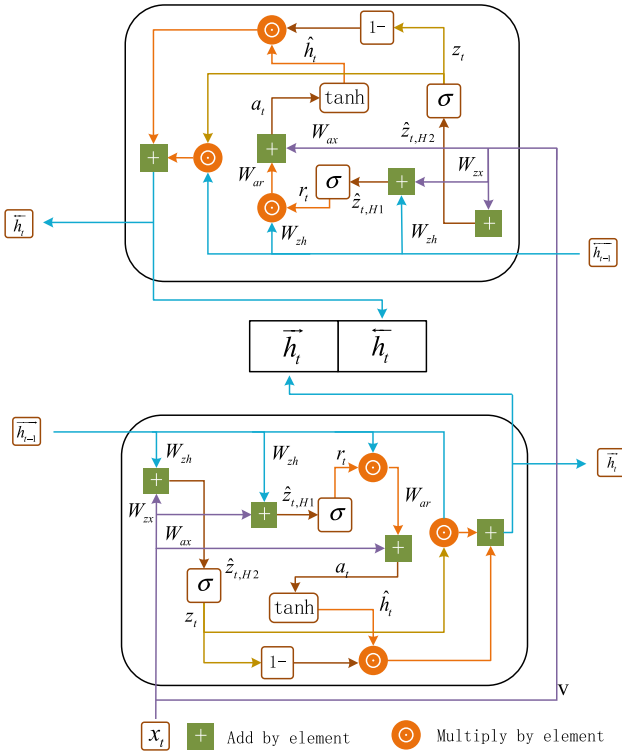


Fig. 6. Internal structure of Bi-GRU.

Then use the reset gate output r_t of the current time step together with the original input x_t to calculate another intermediate feedforward output:

$$a_t = W_{ax}x_t + W_{ar}(\vec{h}_{t-1} \otimes r_t) + b_a \quad (6)$$

Hyperbolic tangent activation of the intermediate feedforward output a_t can be expressed as:

$$\hat{h}_t = \tanh(a_t) \quad (7)$$

Take the output of the update gate as the switch, the hidden state output of the current time step can be obtained:

$$\vec{h}_t = \vec{h}_{t-1} \otimes (1 - z) + z_t \otimes \hat{h}_t \quad (8)$$

3.3. The syntax layer

There are two forms of dependency. One is to mark the dependency arrow and grammatical information directly on the sentence. The other is to make the grammatical relations of the sentence into a tree structure. This paper uses dependency arrows to indicate the grammatical relationships of sentences.

Specifically, the semantic tree of sentences is generated by obtaining the dependency relationship and grammar information of a given text. The semantic tree of the sentence is stored as an adjacency matrix, given the vertex set N of all words in the text, the adjacency matrix of the text is represented as $\text{Adj} \in \mathbb{R}^{N \times N}$. If two words i and j are connected by arrows, then $\text{Adj}[i, j] = 1$, $\text{Adj}[j, i] = 1$. According to Kipf [32], it is assumed that the set of edges contains a self-iterating $\text{Adj} = \text{Adj} + \mathbf{I}$, that is, $\text{Adj}[v, v] = 1$ for any vertex v .

3.4. Graph convolution layer

The task of the graph convolutional layer is to use the graph convolutional network to generate the sentiment feature representation of a given text based on the dependency tree embedded with feature vectors.

tation of a given text based on the dependency tree embedded with feature vectors.

For a graph $G(V, E)$, input X is a matrix of $N \times D$, N represents the number of nodes, D represents the number of input features, the values in the matrix are the characteristics of each node and the adjacency matrix A of the graph. Output an $N \times F$ characteristic matrix Z , which represents the characteristic representation of each node learned, where F is the dimension of categories desired. Corresponding to the proposed DTGCN model in this paper, the input matrix X is the feature matrix of each text trained from the Bi-GRU layer, matrix A is the adjacency matrix Adj of the syntax-dependent tree obtained from the grammar layer, and the feature matrix Z is the feature of the text extracted by the graph convolution.

Equation (9) is the initial state of graph convolution. The single-layer forward propagation of graph convolution is shown in Equation (10). Each neural network layer can be written as a nonlinear function in Equation (11).

$$H^{(0)} = X \quad (9)$$

$$f(H^{(l)}, A) = \text{RELU}(\hat{D}^{-\frac{1}{2}} A \hat{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \quad (10)$$

$$H^{(l+1)} = f(H^{(l)}, A) \quad (11)$$

where Relu is the activation function, \hat{D} denotes the degree matrix of the dependent tree, expresses the weight matrix by learning and training.

3.5. Output layer

3.5.1. MaxPooling

Pooling is also known as sub-sampling. It only reduces the size of the matrix and does not change the depth of the three-dimensional tensor. Pooling layer can reduce the number of nodes in the full connection layer and alleviate the risk of overfitting by reducing the number of parameters in the whole neural network. In this paper, the maximum pool is used to extract some feature values from the output of the graph convolutional neural network. Only the value with the maximum score is taken as the pooling layer reserve value, and all other feature values are discarded.

3.5.2. Softmax

Softmax classifier is used to construct the conditional probability distribution of each emotion tag and output the final emotion tag of the microblog text. The mathematical expressions of the predicted values of the output tags and emotions are shown in Equations (12) and (13).

$$y_j = \text{softmax}(wH + b) \quad (12)$$

$$p_j = \frac{\exp(y_j)}{\sum_{i=1}^J \exp(y_i)} \quad (13)$$

where y_j is the output tag, w and b are the weights and biases learned by training respectively, J denotes the number of emotion categories, and p_j expresses the predicted value of j emotion.

3.5.3. L_2 regularization

Generalization ability refers to the ability to accurately predict data other than training data. In order to increase the generalization ability of proposed DTGCN model. In this paper, a constraint L_2 normal form is added to the loss function.

$$Loss = - \sum_{j=1}^J y_j p_j + \lambda \| \theta \|^2 \quad (14)$$

where $-\sum_{j=1}^J y_j s_{p_j}$ is the initial loss function of the proposed DTGCN model, λ represents regularization coefficient of L_2 , θ shows all trainable parameters.

4. Experiments and analysis

The experimental environment is Ubuntu16.04LTS operating system, CPU is CoReI5-8300h, 64 GB memory, 2 TB hard disk, GPU is Nvidia GeForce GTX 1060. The experiment is based on deep learning framework Pytorch, and the development language used in the experiment is Python.

4.1. Datasets

In order to verify the validity of the proposed DTGCN model, this paper crawled 99,300 valid microblog data. They are labeled as positive, negative and neutral. The training set and test set were divided in 8:2 ratios. For model training and for model testing. The specific distribution of emotional polarity in the data set is shown in [Table 2](#). The emotion category of data in the experiment is represented by one-hot vector, for example, the Positive one-hot is [1.0,0.0,0.0].

In order to observe the distribution of terms in the data set more intuitively, a word cloud of the first 300 keywords is constructed (as shown in Fig. 7).

In Fig. 7, the font size of words in the word cloud is in direct proportion to the number of words in the data set. The more frequently the number appears, the larger the font size of the words in Fig. 7. It can be seen from the word cloud that “Pneumonia”, “COVID-19” and “Coronavirus” dominate the data set.

4.2. Performance measure

Obfuscation matrix, also known as error matrix, is a standard format for accuracy evaluation. Each column of the matrix represents the predicted value. Each row represents the actual category. The classical confusion matrix is shown in [Table 3](#).

For the evaluation of proposed DTGCN model, evaluation indexes such as *Accuracy* (*Acc*), Precision rate *P*, macro average *F1* value (*Macro_F1*) and micro average *F1* value (*Micro_F1*) are

Table 2
Datasets.

	Positive	Neutral	Negative
Train	20,001	46,764	12,675
Test	5371	10,265	4224



Fig. 7. Data set word cloud.

Table 3
Confusion Matrix.

Confusion Matrix		Predict class	
		Positive	Negative
Actual class	Positive	TP	FP
	Negative	FN	TN

adopted. The *Macro_F1* value refers to the calculation of precision rate and recall rate respectively on the obfuscating matrix, and then the average value. *Micro_F1* first averages the corresponding elements of each confusion matrix to obtain the average values of *TP*, *FP*, *TN*, *FN*. Recall and precision are calculated based on these averages. For a single category, let *TP* be the correct predicted sample, *FP* is the sample whose other category is judged as the current category, *FN* is the sample whose current category is wrongly judged as another category, then the exact rate *P*, *Recall* rate and *F1* value can be calculated as follows:

$$P = \frac{TP}{TP + FN} \quad (15)$$

$$Recall = \frac{TP}{TP + FP} \quad (16)$$

$$F1 = \frac{2 \times P \times Recall}{P + Recall} \quad (17)$$

Accuracy Acc and $Macro_F1$, $Micro_F1$ are respectively defined as:

$$Acc = \frac{\sum_{i=1}^m TP_i}{\sum_{i=1}^m (TP_i + FP_i + FN_i)} \quad (18)$$

$$Macro_F1 = \frac{1}{m} \sum_{i=1}^m F1 \quad (19)$$

$$Micro_P = \frac{\sum_{i=1}^m TP_i}{\sum_{i=1}^m TP_i + \sum_{i=1}^m FP_i} \quad (20)$$

$$Micro_P = \frac{\sum_{i=1}^m TP_i}{\sum_{i=1}^m TP_i + \sum_{i=1}^m FN_i} \quad (21)$$

$$Micro_F1 = \frac{2 \times Micro_P \times Micro_R}{Micro_P + Micro_R} \quad (22)$$

4.3. Hyper-parameter setting

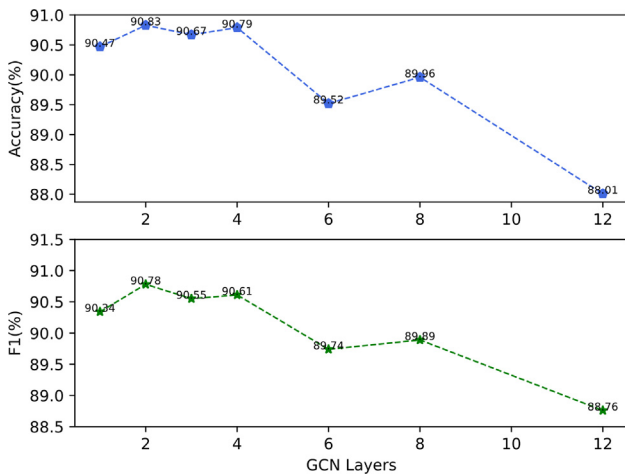
Usually, deep learning algorithm needs to obtain the optimal value of parameters through optimization (model adjustment) or change the preprocessing of selected model data to improve the model to reach the optimal verification accuracy, and adjust and evaluate the performance of training model in the iterative process. This usually requires repeated experimentation to change the model's hyper-parameters, or to include/exclude specific preprocessing steps. Then, the performance of the verification set is evaluated. According to the accuracy of the experiment, the loss rate is adjusted until the parameter achieves the highest validation performance. After several iterations, the experimental super parameters are shown in [Table 4](#).

In order to study the effect of the number of GCN layers on the final performance of DTGCN. Set the number of GCN layers to L , take the values in the set $\{1, 2, 3, 4, 6, 8, 12\}$ in turn, and check the corresponding accuracy of DTGCN of the data set and the macro average F1. The result is shown in the Fig. 8.

Table 4

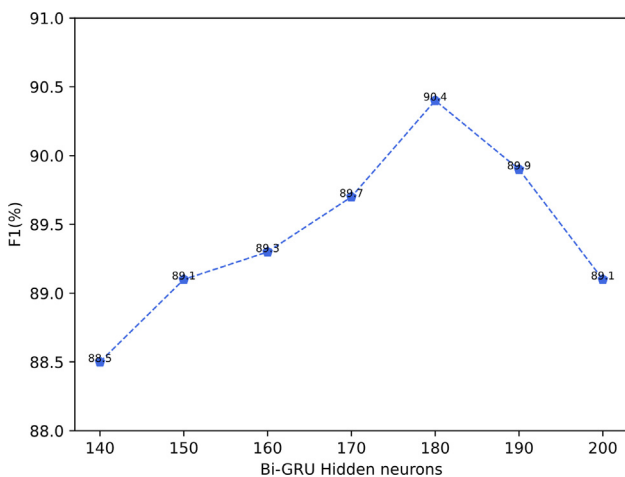
Parameter Settings.

Parameters	Value
Embedding size	300
Bi-GRU Hidden neurons	180
Bi-GRU Hidden layers	2
GCN Network size	[360,3]
GCN Activation Function	Relu
Learning rate	0.001
Optimizer	Adam

**Fig. 8.** Influence of GCN Layers.

It can be seen from the Fig. 8 that an increase in the number of layers of the Fig. 8 does not necessarily improve the accuracy of the model. On the contrary, when the number of layers is greater than 4, the accuracy will decrease with the increase of the number of layers. This is because when the network structure of the model is more complex, the training score is very high, but the test score is very low, there is so-called overfitting. In the Fig. 8, when $L = 2$, that is, when the number of layers of the graph convolutional network is 2, the accuracy of the model is 0.2% higher than the accuracy of $L = 1$, but this article only uses a single layer because of the efficiency of model training Graph convolution.

The Fig. 9 shows the effect of Bi-GRU's hidden neurons on model F1. It can be seen that when the Bi-GRU's hidden neurons are 180, the effect of the model is better.

**Fig. 9.** Influence of hidden neurons.

Due to the large number of parameters in model learning, it is easy to over-fit. Therefore, in order to solve the problem of overfitting, add a dropout layer in the input layer, that is, randomly select the nodes to be discarded according to a given probability. In the process of forward propagation, the contribution of these neglected nodes to downstream nodes is temporarily lost, and in back propagation, these nodes do not have any weight updates. In order to find the dropout value suitable for the model, an experiment was conducted on the data set. The experimental results are shown in Fig. 10.

From Fig. 10(a) to Fig. 10(d), it can be seen that when the dropout value is 0.7, the accuracy of the red line increases rapidly. However, the numerical value increases slowly and fluctuates greatly after 5 iterations, so the best result of the model cannot be obtained. When the dropout value is 0.3, although the fluctuation of pink line is relatively stable, the optimal value of model evaluation index is not obtained. When the dropout value is 0.5, the blue line fluctuates less during training than other lines. Moreover, the blue line can obtain a higher average value of indexes compared with other line models, that is, the best performance of the model. Therefore, the dropout rate of the experiment in this paper is set as 0.5.

In order to verify the influence of the dependency parser on the model, this article uses the following three parsers to analyze the same sentence:

LTP: "Language Cloud" is based on the "Language Technology Platform (LTP)" developed by the Social Computing and Information Retrieval Research Center of Harbin Institute of Technology to provide users with efficient and accurate Chinese natural language processing cloud services.

Stanford Parser: Stanford Parser is one of a series of tools provided by the Stanford NLP team to complete the task of grammatical analysis. Stanford Parser can find out the dependency related information between words in the sentence, and output it in Stanford Dependency format, including directed graphs and trees.

Berkeley Parser: Berkeley Parser is an open-source syntactic analyzer developed by the Natural Language Processing Group of the University of Berkeley. Currently supported languages mainly include English, Chinese, German, Arabic, Bulgarian, French, etc.

Among the four results, Fig. 11(a) and Fig. 11(b) are the dependent grammatical analysis of the same sentence by LTP and Stanford Parser, and Fig. 11(c) and Fig. 11(d) are the sentence component analysis of Stanford Parser and Berkeley Parser. By comparing Fig. 11(a) and Fig. 11(b), it can be found that because different classifiers have different judgment rules of the dependency relationship between sentence components, the dependency between sentence components is the same; although Berkeley Parser can only analyze sentence component types, by comparing the results of sentence component analysis Fig. 11(c) and Fig. 11(d) with Stanford Parser, it can also show that different parsers are similar to sentence component analysis.

In order to quantify the impact of the parsing tools on the model, the data processed by the three parsers were subjected to experiments with the following results: By comparing the last three rows of data in the table, it can be seen that although there is a slight difference between the three results, the overall effect of the dependent analysis tool on the model is limited.

Therefore, it can be concluded that the proposed DTGCN model is not sensitive to the performance of dependency analysis.

4.4. Performance of proposed DTGRU model

The preprocessing steps of data cleansing were applied to collate the data set, and then the model was loaded to process the data. The model trains each sentence level to assign an emotional label (positive, negative, neutral). Model performance is evaluated

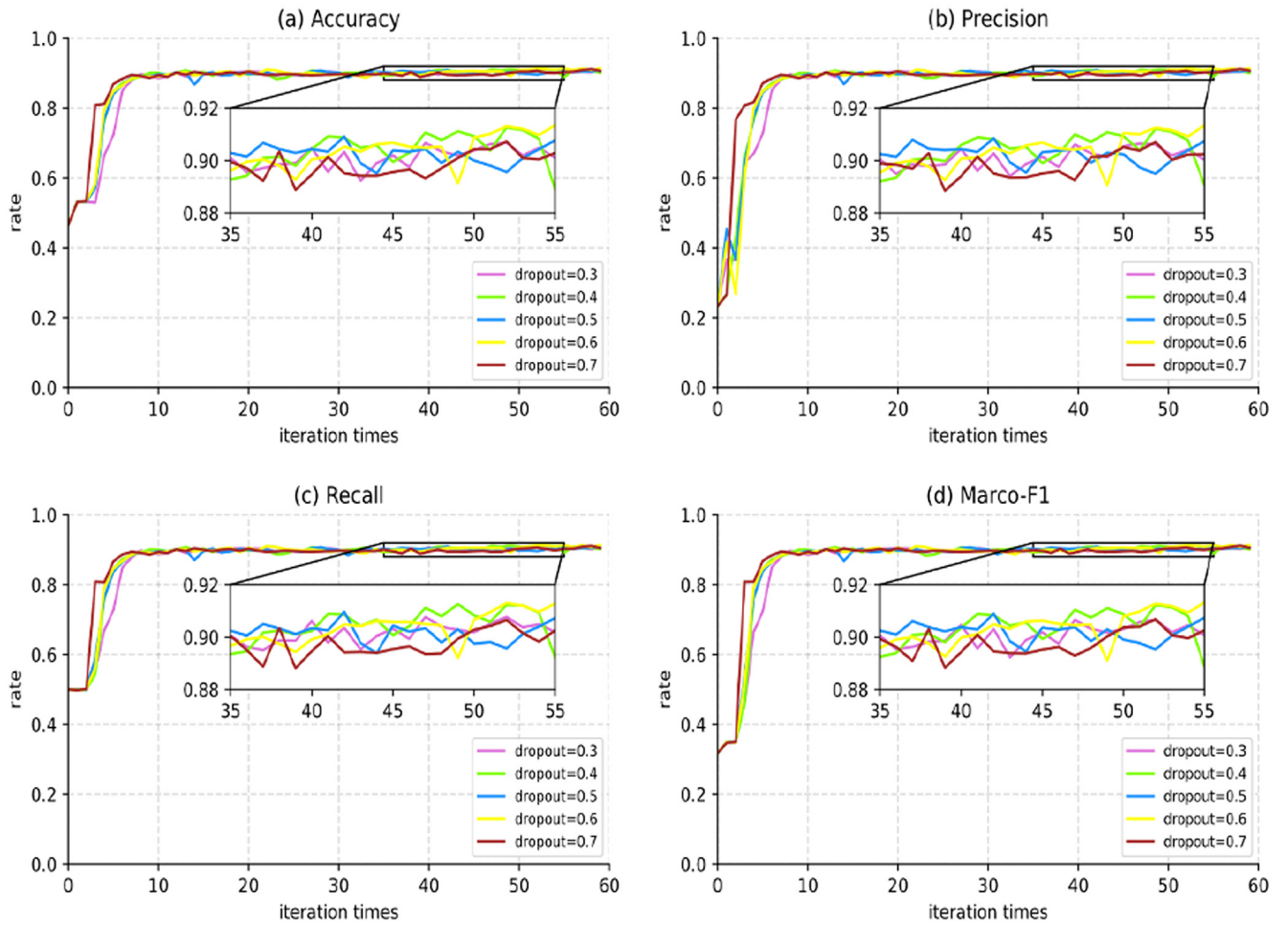


Fig. 10. Influence of dropout on model results.

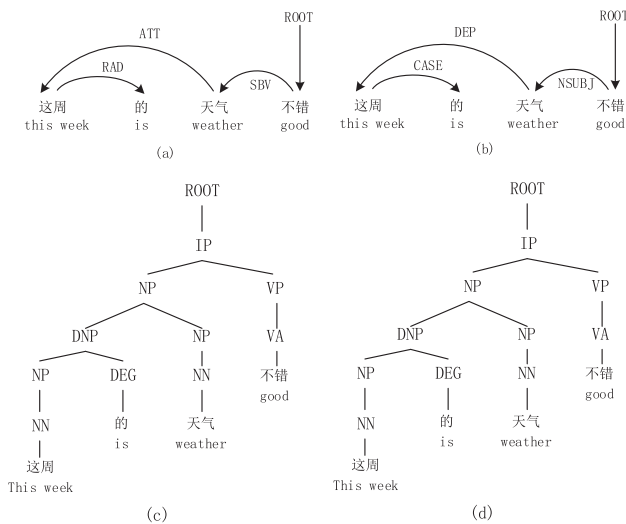


Fig. 11. Influence of the dependency parser.

according to accuracy rate (*Acc*), precision rate *P*, *Recall* and macro average *F1* value (*Macro.F1*). As can be seen in Table 6.

Specific numerical images are shown in Fig. 12.

As can be seen from Table 5 and Fig. 12, the model can achieve high accuracy and *Macro.F1* with macro-mean value. It proves that proposed DTGRU, model can better extract the emotional features

of the Chinese short text. It can more accurately analyze the emotions expressed by netizens in social networks.

4.5. Comparative analysis of different models

In order to better verify the validity of the proposed DTGCN model, several widely used emotion classification algorithms are selected as the baseline. Including the traditional deep learning method and the most advanced neural network structure, the model results are shown in Table 7.

Model 1: LSTM, which only uses a single-layer LSTM network to model sentences. The average of all hidden states is treated as the final sentence representation.

Model 2: CNN, the convolutional neural network is a feedforward neural network with deep structure and contains convolution computation. It is also a classic model in NLP tasks.

Model 3: CNN + LSTM. This structure is similar to the model in this paper, but LSTM is used instead of GRU to extract text features. CNN instead of GCN is used to extract comprehensive features.

Model 4: TextCNN, which is usually considered as a work for computer vision direction, Yoon Kim [33] made some deformation for the input layer of CNN and proposed text classification model TextCNN to deal with NLP.

Model 5: Bi-GRU + GCN. This model does not introduce the syntax information of the Chinese short text, but directly adopts the Bi-GRU for modeling. Input text features into GCN to achieve emotional analysis.

Table 5

Comparison of different dependency parsing tools on the NLP&CC2013 and NLP&CC2014 dataset.

Model	NLP&CC2013 ¹				NLP&CC2014 ²			
	Acc	F1	P	Recall	Acc	F1	P	Recall
DTGCN (LTP)	89.90	89.85	89.89	89.81	88.54	88.51	88.47	88.56
DTGCN (Stanford Parser)	88.54	88.53	88.61	88.77	86.26	86.08	86.67	85.88
DTGCN (Berkeley Parser)	88.61	88.43	88.63	88.42	86.06	85.89	86.39	85.72

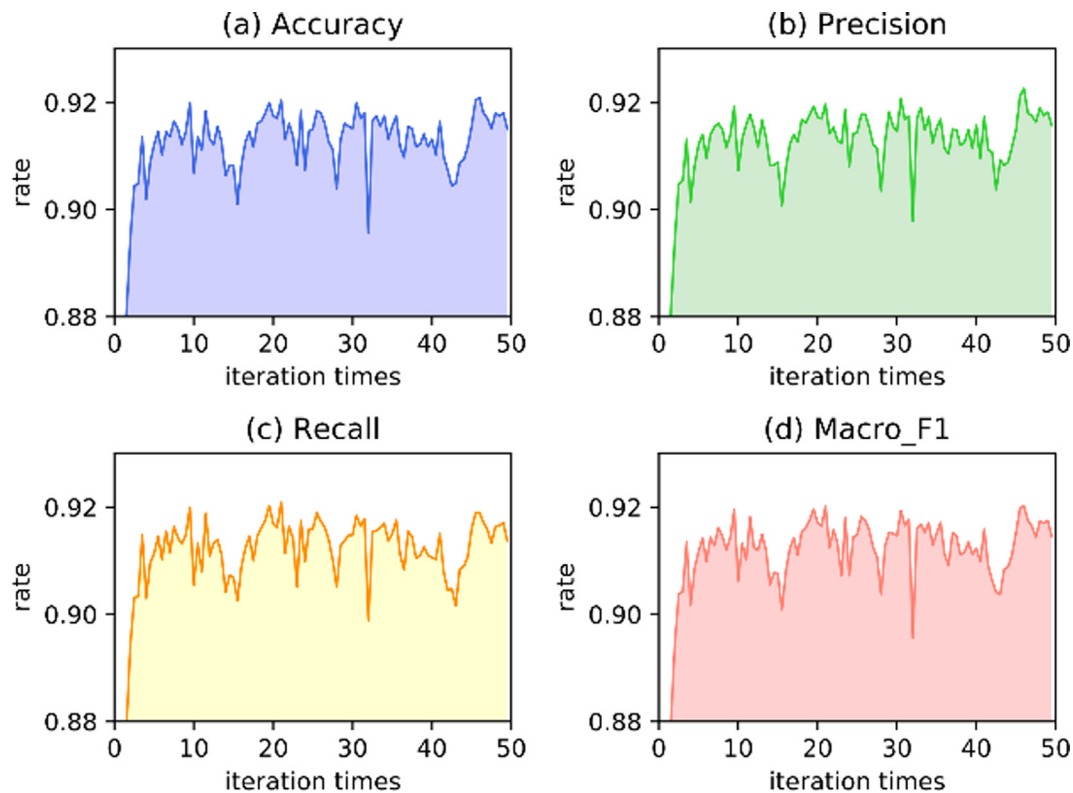
Table 6

Model evaluation indexes.

Acc	P	Recall	Macro_F1	Micro_F1
90.51	90.75	90.34	90.47	90.24

The model performance by each tag category in Table 7 is based on accuracy rate (Acc), macro average F1 value (Macro_F1), the precision rate P is compared with the Recall rate. As can be seen from Table 7, syntactic-based graph convolutional neural network (DTGCN) model performs well in the affective classification of each label. In macro-average Macro_F1, the DTGCN model performs

better than other comparison models. The experiments of model 1 and Model 2 show that a single neural model (such as LSTM and CNN in the experiment). Due to the limitations of its own network results, it cannot better learn the emotional characteristics of the text. Therefore, it is difficult to improve the accuracy of emotion classification only by adjusting network parameters. The experiment of Model 3 is to add LSTM model on the basis of CNN model and input LSTM for semantic feature extraction. The output of LSTM is taken as the input of CNN for further feature extraction. Finally, the classification results are obtained. The experimental results show that although Model 1 and Model 2 are improved, the accuracy of model 3 is still low among other model results.

**Fig. 12.** DTGCN model performance diagram.**Table 7**

Comparison of polarity classification results.

Model	Positive				Neutral				Negative			
	Acc	F1	P	Recall	Acc	F1	P	Recall	Acc	F1	P	Recall
SVM	63.93	65.76	61.12	71.15	76.76	76.94	79.38	74.64	54.55	55.12	54.88	53.36
LSTM	73.93	67.83	62.66	73.93	77.80	78.76	77.24	73.94	54.56	54.41	47.56	63.56
CNN	72.15	71.71	71.29	70.14	76.94	78.41	77.93	73.56	64.88	65.16	64.91	63.24
CNN + LSTM	79.61	79.80	79.43	78.73	80.48	80.99	80.02	79.89	78.13	78.74	77.48	77.53
TextCNN	81.17	80.47	80.56	80.48	81.76	82.42	80.64	81.09	79.28	79.30	79.41	78.02
Bi-GRU + GCN	86.24	86.98	85.76	85.64	87.22	87.71	88.29	85.94	84.71	85.50	85.38	84.95
DTGCN	90.41	90.33	89.51	90.23	91.21	91.22	90.81	90.87	89.93	89.87	89.23	89.92

Table 8

Comparison of different models on the NLP&CC2013 and NLP&CC2014 testing dataset.

Model	NLP&CC2013				NLP&CC2014			
	Acc	F1	P	Recall	Acc	F1	P	Recall
SVM	53.20	46.01	54.53	55.61	52.96	44.82	46.64	49.87
LSTM	66.29	67.18	66.45	69.33	65.21	66.10	65.43	64.15
CNN	63.40	63.31	64.61	65.13	62.17	63.88	62.47	63.08
CNN + LSTM	72.48	70.81	72.67	72.05	71.56	69.92	70.17	71.73
TextCNN	77.43	76.72	76.45	78.33	76.71	76.21	74.67	76.47
BiGRU + GCN	84.86	84.19	85.44	84.78	83.29	82.76	83.10	82.21
DTGCN(LTP)	89.90	89.85	89.89	89.81	88.54	88.51	88.47	88.56

Table 9

Comparison of different uses of dependencies.

Method	ACC	F1
Dependency Embedding + DTGCN	78.87	72.01
Dependency Embedding + BiGRU	81.73	80.94
DTGCN	90.51	90.47

The Bi-GRU + GCN model of Model 6 is a reduced version of the DTGCN model. The model in Experiment 6 lacks the syntactic information of the input sample, that is, all elements in the adjacency matrix in GCN are assigned a value of 1. The accuracy of the proposed DTGCN model is 4.45% higher than that of Bi-GRU + GCN through experimental verification. Thus, the importance of syntactic information in affective classification is highlighted. In the proposed DTGCN model, the accuracy of “neutral” and *Macro_F1* are the highest, which is 91.21% and 91.22%, respectively. The imbalance of data is part of the reason for this result. As the number of neutral samples is relatively large, the characteristics obtained from neutral samples after training will be more abundant and thus have higher accuracy and precision.

In order to verify the generalization ability of this model, two datasets, NLP&CC2013 and NLP&CC2014, are selected in this paper, and the performance of each model on these two datasets is plotted as Table 8 without changing the model parameters. According to Table 8, we can see that the classification effect of DTGCN model decreases on different datasets, but compared with other models both This proves that the model has good transferability. Meanwhile, this paper also introduces a comparison experiment using the same sentence dependency-based Word Embeddings [34], and the experimental results are shown in Table 9. Comparing the data in the first row of the table with the data in the third row, we found that the DTGCN model with Dependency-based word embeddings did not increase the accuracy of the model but decreased it, so we considered whether the model was too redundant and thus add a set of comparison tests Dependency Embeddings + BiGRU, which can be seen from the data in the second row of the table that both accuracy and Macro-averaged F1 have improved, but do not perform better than the model in this paper.

5. Discussion

This section discusses the value achieved in this article. First of all, this paper established 99,200 large-scale Chinese emotion analysis corpora through crawling microblog data for Chinese short texts in social networks emotional analysis; secondly, this paper combines the efficient double-layer gated neural network with the graph convolutional neural network to build the DTGCN model. In the data set, the DTGCN model achieved 90.51% accuracy and 90.34% recall rate. Compared with other most advanced deep learning technologies, LSTM and CNN, this result is better improved, because in previous studies, a model like Bi-LSTM only successfully captures context information. In this paper, GCN is

convolved in synter-dependent trees to optimize Bi-GRU embedding and obtain sentence structure and context information. Therefore, the DTGCN model is better than the more complex and up-to-date model when dealing with the same problem. Finally, the proposed DTGCN model can not only process the emotional analysis of Chinese text. If the corresponding language corpus is obtained and appropriate tags are added to the data set, it can better analyze the emotional polarity of other languages, such as English. This can provide a more detailed, deeper emotional analysis.

6. Conclusion

Attributive emotion classification of sentences has been a research focus in the field of natural language processing in recent years. However, the research on the classification of Chinese text emotions pales in comparison with English documents. This paper aims at the phenomenon that only the context information of attributes is not combined with syntactic dependency tree in the current related researches. A syntactic-based image convolution emotion classification model DTGCN is proposed. It provides a new direction for the research of emotion analysis of Chinese short texts in social network. Finally, in order to verify the validity of the proposed DTGCN model, in the experiment on the data set, its accuracy and *Macro_F1* are 90.51% and 90.47%, respectively. In addition, the accuracy of DTGCN and *Macro_F1* exceeded that of the benchmark model by 4.45% and 3.71% respectively when comparing multiple benchmark model experiments. The experimental results show that proposed DTGCN model can obtain the emotional characteristics of Chinese samples by using both syntactic information and long-distance word dependence. The whole performance of emotion classification is improved.

Next, this study will be further improved in the following aspects. Firstly, increase the granularity of sentences like good, joy, sorrow, anger, fear, evil, etc. Fine-grained partitioning helps provide data researchers with additional insights into existing data; secondly, the DTGCN model will be applied to other types of emotional tasks, such as user intention mining and false opinion detection. Evaluate the performance of the DTGCN model against data from other domains.

Acknowledgements

This work was supported in part by National Social Science Fund of China (17XXW004).

Conflicts of interest

The authors declare no conflict of interest.

References

- [1] Li J, Luong MT, Jurafsky D, et al. When are tree structures necessary for deep learning of representations. *Comput Sci* 2015;2304–14.
- [2] Mou L, Li G, Zhang L, et al. Convolutional neural networks over tree structures for programming language processing. In: *National Conference on Artificial Intelligence*. p. 1287–93.
- [3] Wan Y, Zhao Z, Yang M, et al. Improving automatic source code summarization via deep reinforcement learning. *Software Eng* 2018;7:397–407.
- [4] Luo H, Li T, Liu B, et al. Improving aspect term extraction with bidirectional dependency tree representation. *IEEE Trans Audio Speech Lang Process* 2019;27(7):1201–12.
- [5] Ahmed M, Samee MR, Mercer RE, et al. Improving tree-LSTM with tree attention. *Comput Language* 2019;1:247–54.
- [6] Zhang J, Xu G, Wang X, et al. Syntax-aware representation for aspect term extraction. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. p. 123–34.
- [7] Park Y, Kang S. Natural language generation using dependency tree decoding for spoken dialog systems. *IEEE Access* 2019;7:7250–8.
- [8] Shi Z, Huang M. A deep sequential model for discourse parsing on multi-party dialogues. *Comput Language* 2019;33(1):7007–14.
- [9] Giovanna C, Ciro C, Corrado M, Gennaro V. Crowd detection in aerial images using spatial graphs and fully-convolutional neural networks. *IEEE Access* 2020;8:64534–44.
- [10] Chen L, Xie Y, Zheng Z, et al. Friend recommendation based on multi-social graph convolutional network. *IEEE Access* 2020;8:43618–29.
- [11] Wan S, Gong C, Zhong P, et al. Multiscale dynamic graph convolutional network for hyperspectral image classification. *IEEE Trans Geosci Remote Sens* 2019;58:3162–77.
- [12] Yao L, Mao C, Luo Y, et al. Graph Convolutional networks for text classification. *National Conference on Artificial Intelligence*, 2019, 33(1):7370–7377.
- [13] Maron ME. Automatic indexing: an experimental inquiry. *J ACM* 1961;8(3):404–17.
- [14] Cortes C, Vapnik V. Support-vector networks. *Machine Learn* 1995;20(3):273–97.
- [15] Hu Y, Chen K. Predicting hotel review helpfulness. *Int J Inf Manage* 2016;36(6):929–44.
- [16] Zhang Y, Zhang Z, Miao D, et al. Three-way enhanced convolutional neural networks for sentence-level sentiment classification. *Inf Sci* 2018;477:55–64.
- [17] Lu Z, Cao L, Zhang Y, et al. Speech sentiment analysis via pre-trained features from end-to-end ASR. *Models* 2019;11:68–82.
- [18] Wen S, Wei H, Yang Y, et al. Memristive LSTM network for sentiment analysis. *IEEE Trans Syst Man Cybernetics* 2019;4:1–11.
- [19] Kim Y. Convolutional neural networks for sentence classification. In: *Conference on Empirical Methods in Natural Language Processing*. p. 1746–51.
- [20] Okada M, Yanagimoto H, Hashimoto K, et al. Sentiment classification with gated CNN and spatial pyramid pooling. In: *International Conference on Advanced Applied Informatics*. p. 133–8.
- [21] Jiana M, Yingchun L, Yuhai Y, et al. Cross-domain text sentiment analysis based on CNN. *Information* 2019;10(5):162–75.
- [22] Jiang M, Zhang W, Zhang M, et al. An LSTM-CNN attention approach for aspect-level sentiment classification. *J Comput Methods Sci Eng* 2019;19(4):859–68.
- [23] Luo L. Network text sentiment analysis method combining LDA text representation and GRU-CNN. *Pers Ubiquit Comput* 2019;23(3):405–12.
- [24] Minh DL, Sadeghiniaraki A, Huy HD, et al. Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. *IEEE Access* 2018;5:5392–404.
- [25] Hong M, Wang M, Luo L, et al. Combining gated recurrent unit and attention pooling for sentimental classification. In: *The 2018 2nd International Conference*. p. 99–104.
- [26] Bleiweiss A. Machine florigraphy: sentiment-inspired flower predictions over gated recurrent neural networks. In: *International Conference on Agents and Artificial Intelligence*. p. 413–21.
- [27] Chen C, Zhuo R, Ren J, et al. Gated recurrent neural network with sentimental relations for sentiment classification. *Inf Sci* 2019;268–78.
- [28] Lee JS, Zuba D, Pang Y, et al. Sentiment analysis of chinese product reviews using gated recurrent unit. In: *International Conference on Big Data*. p. 173–81.
- [29] Yu Q, Zhao H, Wang Z, et al. Attention-based bidirectional gated recurrent unit neural networks for sentiment analysis. In: *International Conference on Artificial Intelligence*. p. 116–9.
- [30] Sachin S, Tripathi A, Mahajan N, et al. Sentiment analysis using gated recurrent neural networks. *SN Comput Sci* 2020;74(1):78–96.
- [31] Fagui L, Jingzhong Z, et al. Combining attention-based bidirectional gated recurrent neural network and two-dimensional convolutional neural network for document-level sentiment classification. *Neurocomputing* 2020;372:39–50.
- [32] Kipf T, Welling M. Semi-supervised classification with graph convolutional networks. In: *International Conference on Learning Representations*. p. 342–54.
- [33] Kim Y. Convolutional neural networks for sentence classification. In: *Empirical Methods in Natural Language Processing*. p. 1746–51.
- [34] Levy O, Goldberg Y. Dependency-based word embeddings. In: *Meeting of the Association for Computational Linguistics*. p. 324–31.