Computer Science and Information Systems 21(4):1583–1605 https://doi.org/10.2298/CSIS240314049L

GCN-LSTM: Multi-label educational emotion prediction based on graph Convolutional network and long and short term memory network fusion label correlation in online social networks

Zhiguang Liu[1,⋆], Fengshuai Li[2], Guoyin Hao[3], Xiaoqing He[1], and Yuanheng Zhang[1]

[1] School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology
Zhengzhou, 450064 China
snowycry@qq.com

[2] College of Civil and Architectural Engineering, Zhengzhou University of Science and Technology
Zhengzhou, 450064 China
lifsfsfs@163.com

[3] School of Music and Dance, Zhengzhou University of Science and Technology
Zhengzhou, 450064 China
hexiaoqingvip@163.com

Abstract. Although there are a lot of methods for multi-label classification in the past research, there are still many problems. For example, in the real world, labels are not necessarily independent of each other, and there may be some connection between labels. Therefore, exploring and utilizing the interdependence between labels is a key issue in current research. For example, in the photo category, a picture that contains blue sky often also contains white clouds, and in the text category, a political story is less likely to be entertainment news. Therefore, the key to improve the accuracy of multi-label classification is to effectively learn the possible correlations between each label. Therefore, we propose a novel multi-label educational emotion prediction based on graph convolutional network and long and short term memory network fusion label correlation in online social networks. This model uses Word2Vec method to train word vectors and combines graph convolutional neural network (GCN) with long and short term memory network (LSTM). The GCN is used to dig deeper word features of text, the LSTM layer is used to learn the long-term dependence relationship between words, and the multi-granularity attention mechanism is used to assign higher weight to the affective word features. At the same time, label correlation matrix is used to complete the label feature vector and text features as the input of the classifier, and the correlation between labels is investigated. The experimental results on the open data set show that the proposed model has a good classification effect compared with other advanced methods. The research results promote the combination of deep learning and affective computing, and can promote the research of network user behavior analysis and prediction, which can be used in personalized recommendation, targeted advertising and other fields, and has wide academic significance and application prospects.

Keywords: multi-label educational emotion prediction, GCN, LSTM, multi-granularity attention mechanism.

⋆ Corresponding Author

1. Introduction

With the rapid development of Internet technology, data such as pictures and text in education have also shown rapid growth, and the development of data has provided a cornerstone for machine learning. Therefore, multi-label classification learning has become a hot spot in the field of machine learning research, which is mainly reflected in emotion classification, image and video semantic annotation [1,2], gene function analysis, etc. The traditional single-label classification problem, in which an object belongs to only one specific category, however, in most classification tasks, an instance may contain multiple categories at the same time. For example, in a daily text classification task, it may belong to both the entertainment and economic categories; In the image classification task, an image may contain cats, dogs, people, and the sun at the same time. For this kind of multi-label data, the classification task changes from the traditional prediction of one category to the prediction of multiple categories [3,4].

In traditional image classification and recognition, the information contained in the image (such as edge detection, contour detection, local binary mode (LPB), directional gradient histogram (HOG), Haar feature detection) is usually extracted manually to achieve image classification [5,6]. However, these extraction methods not only cost a lot of manpower, but also have many serious problems. For example, the image is affected by external environmental factors, and the change of spatial information such as illumination, rotation and translation of objects leads to the change of object characteristics, resulting in the decline of classification accuracy.

The task of affective prediction is to determine an attitude towards a goal or subject, which can be either a polarity (positive or negative) or an emotional state such as joy, anger, or sadness [7]. Most previous studies have judged the positive or negative attitudes contained in social media content, such as whether customers are satisfied with a product in product reviews, or whether viewers view a movie as exciting or boring in movie reviews. Most of the work on judging emotional states only focus on single-label classification, dividing emotional texts into one emotional category in multiple categories, while ignoring the situation that multiple emotional categories may coexist in a text instance, which is obviously inconsistent with reality [8]. For example, an old man who has three sons but no one to take care of him in his old age may cause the reader to feel sad, angry, or other different emotions.

In recent years, the aspect-level emotion classification model based on deep learning [9] performs well. These models are mainly based on Long Short-Term memory network (LSTM) and Bi-directional LSTM (Bi-LSTM), and introduce attention mechanism and position weight to improve the classification performance. Multi-label classification has been widely used in many fields, such as text classification, image annotation, bioinformatics [10] and so on. At present, the research on multi-label sentiment classification of microblog texts is still in its infancy. The Second Conference on Natural Language Processing and Chinese Computing presented a fine-grained sentiment prediction task for Chinese microblogs for the first time, requiring multiple sentiment tags to be selected from a set of associated tags for each instance of a microblog. In 2019, Parwez et al. [11] used a convolution neural network (CNN) model to synthesize word vectors in microblog sentences into sentence vectors. These sentence vectors were used as features to train a multi-label classifier to complete the multi-label sentiment classification of microblogging on the NLP&CC2013 dataset. This method only extracted the local features of the

text without considering the global semantic information of the text. In 2020, Wang et al. [12] proposed a long short term memory networks (LSTM) model based on hierarchical attention. It used an attention mechanism to represent the text, and then used LSTM for multi-label classification, but this method ignored the local features of the text. In 2020, Acheampong et al. [13] used the FastText model to make emotion prediction on the crawled news text dataset. The experimental results showed that the accuracy rate of the FastText model was higher than that of traditional machine learning methods such as support vector machine and logistic regression, and the speed advantage was more obvious than that of the neural network model. In 20220, Reza et al. [14] uses Transformer multi-head attention mechanism to simultaneously acquire global features and partial information related to specific aspects of the text, which to a certain extent solved the problems that it was difficult for CNN to obtain global semantic information, the RNN training speed was too slow, and the dependence degree between words decreased with the increase of distance. Fine-grained sentiment analysis was performed on the Amazon food review dataset and Booking Hotel review dataset, but the method failed to take into account the relevance of labels.

In terms of LSTM basic network, Kumar et al. [15] adopted two LSTM networks to encode statements from the left and right sides of aspect words respectively. Fadel et al. [16] used multi-layer Bi-LSTM network to model the concatenation of statements and aspect words. Li et al. [17] proposed Target-Specific Transformation Networks-Adaptive Scaling (TNET-AS). The context conversion unit was built based on Bi-LSTM, and the features were extracted from the output by Convolutional Neural Networks (CNNs). However, because LSTM was time-dependent, it could not be able to accurately identify the potential association between distant descriptive words and aspect words when dealing with complex long sentences. Moreover, when multiple aspect words with different emotional polarity appear at the same time, LSTM could mistakenly identify some adjectives or phrases with obvious emotional color as the modifiers of aspect words, resulting in semantic confusion.

Graph Convolutional Networks (GCN) [18] can be seen as an improvement over traditional CNN-encoded unstructured data. Although CNN can handle semantically rich text structures and are easy to parallelize, convolution operation will treat the features of multiple words as continuous words and cannot accurately identify the emotions described by multiple discontinuous words. GCN can solve the problems of LSTM and CNN in processing text data to a certain extent. By convoluting the input statements and their syntactic dependence graphs, GCN can gather the distant descriptive words and aspect words into a smaller range, better combine the syntax information of the sentences, and find the complex dependence relationship between long-term multi-words. Therefore, it has certain advantages in the problem of aspect level emotion classification. Zhu et al. [19] used GCN to build aspect-specific GCN (ASGCN), used Bi-LSTM to capture context information, and multi-layer graph convolution to extract aspect word features, and fed aspect word features back to the hidden layer of Bi-LSTM.

In order to make the model focus on the semantic information related to aspect words, attention mechanism has been introduced in most models in recent years. Jang et al. [20] used word vectors and Bi-LSTM hidden vectors respectively to compute attention iteratively. Kamyab et al. [21] used both word vector and hidden vector to calculate attention in segmented decoding. Li et al. [22] used multi-head attention mechanism to improve

the memory network and alleviate the selective preference of the model. The introduction of attention mechanism can further improve the classification performance. However, the above attention mechanism does not pay attention to the bidirectional influence relationship between a particular aspect and its context, that is, the semantics of a particular aspect are based on the context, and the emotion expression of the context is also based on a specific aspect. In a statement, some words can express semantics and emotions independently (fine-grained), while others need to form phrases with neighboring words to have full semantics (coarse-grained). Based on this situation, Liu et al. [23] proposed Interactive Attention Networks (IAN) to calculate the interaction between words and sentences from a coarse-grained perspective. Puthige et al. [24] proposed an Attention-over-Attention (AOA) neural network to calculate the interaction between words and sentences from a fine-grained perspective. Sun et al. [25] proposed that Multi-grained Attention Network (MGAN). The multi-granularity attention mechanism used two sets of symmetric structures to train the attention weight for each word. It could capture the interaction between aspect words and context at the word level and phrase level, respectively, and used aspect alignment loss to describe the influence relationship between aspect words with the same context.

Bidirectional Encoder Representation from Transformers (BERT) [26] is a pre-trained language model based on the coding part of multi-layer bidirectional Transformer, which has obvious advantages in the semantic expression of text. Liu et al. [27] proposed an attention encoder network based on pre-trained BERT model. Mewada et al. [28] used BERT model to encode statements, and used the pooled aspect word vector to classify the sentiment of statements. The fusion of aspect information made BERT pre-training model more suitable for the fine-grained sentiment classification task of text.

In the task of multi-label emotion prediction and classification, not only the local features and global semantic information of text should be considered, but also the correlation between labels should be fully modeled. In 2023, Ahanin et al. [29] proposed a multi-label sentiment classification method that combined context features and text features. In this method, each sentence in the text was initially classified by emotion, and then the emotion label correlation between each sentence was examined by the emotion transfer relationship between the sentences. Finally, the experiment was carried out on the NLP&CC2013 dataset. But this method neglected the local features of the sentence. Huang et al. [30] proposed to learn the multi-label classification of label features by calculating the cosine similarity of labels, and to examine the correlation of labels by judging whether different labels can share the features between labels. In 2000, Yang Kwon et al. [31] performed K-nearest neighbor retrieval on the word vector space of all labels, and took the first k labels with the nearest cosine distance to the network output vector as the multi-label prediction, verifying the feasibility of the proposed method in label semantic expansibility. Therefore, using the correlation between different labels could effectively improve the performance of multi-label learning.

Some existing multi-label sentiment classification methods fail to extract both local features and global semantic information, and some fail to fully consider the correlation between labels. To solve the two problems, a novel multi-label educational emotion prediction based on graph convolutional network and long and short term memory network fusion label correlation in online social networks model is proposed in this paper. GCN is used to extract the local features of words in text and combine them with word vec-

tors as the input of LSTM. As a special RNN model, LSTM can better solve the long dependency problem [32]. After extracting the long-term dependence between word local features and word vectors, the attention mechanism generates text feature vectors at different time steps according to the contribution of each text fragment and combines them into a complete text feature expression. At the same time, in order to solve the correlation between labels, the label correlation matrix is used to complete the label matrix and concatenate the text feature expression. Finally, the text emotion classification is carried out in the output layer, and the extracted text features and the corresponding label matrix are used as the input of the classifier to enhance the accuracy of the classification.

This paper is structured as follows. In section 2, we detailed introduce the related works. Multi-label educational emotion prediction model is proposed in section 3. Experiments are conducted in section 4. There is a conclusion in section 5.

2. Related Works

2.1. Affective Computing

The research process of emotion computing can be summarized into three aspects: how to describe emotion, namely emotion quantification; How to describe the research object, that is, the characteristics; How to build emotional models. About the quantification of emotion, the current research ideas are generally divided into ”category view” and ”dimension view”. Category view divides, for example, the positive, negative and neutral of text classification and Ekman’s six basic emotions. Dimensional view is divided as PAD model [33].

The feature and emotion computing models are determined according to the object of study. The content of social network mainly includes text and image, and text emotion classification is one of the important research contents in natural language processing. In the early image emotion calculation, a single feature of the image, such as color or texture, was used to establish its mapping relationship with emotion. Then the comprehensive features of the image are used to establish an emotional model. With the development of social networks, the emotional model of social networks has become a research hotspot. The following characteristics of social networks should be considered when constructing the user emotion model of social networks: (1) Dynamic, the network structure and content change dynamically with time; (2) Heterogeneity, including text, images, network structure and other data sources.

2.2. Social Network User Emotion

Studies have been conducted to analyze users’ emotional states based on their behaviors in social networks, such as microblogs, geographical locations, and phone records. Predict user personality based on social network structure and user behavior. A study based on data from Facebook shows that the emotions of users on the social network are closely related to their social activities and interconnection. Sociological research shows that Instinctive Empathy makes emotions collective, that is, how people feel depends on what they are exposed to. Some researchers have established influence models and influence propagation models based on user annotations, microblogs and published articles on the Internet [35].

The emotion based on probability graph model has too many assumptions (artificial empirical formula) about the change of emotion, if the assumption is biased on the data set, it is difficult to achieve good results, and the modeling process often only deals with the binary emotion problem each time, which will be worse in the multi-classification problem. At the same time, these studies do not provide a comprehensive and systematic quantitative model of their own psychology, emotions, external influences and behaviors.

2.3. Multi-label Problem Definition

Suppose the training set has m samples, which is denoted as $X = x_1, x_2, \cdots, x_m$. q labels are denoted as $L = l_1, l_2, \cdots, l_q$. $Y = y1, y_2, \cdots, y_m$ denotes label space. Then multi-label classification is the function $f : X \to Y$ learned by training set $(x_i, y_i)|1 \leq i \leq m$. Where $x_i \in X$ represents a sample, $y_i \in X$ represents the class label to which sample x_i belongs, and y_i is a subset of the label L. By inputting the predicted data into the trained model, the label classification results that are close to the reality can be obtained.

The traditional method is to train a classifier independently for each label, and train each classifier separately with all the data. However, this independent method does not take into account the reciprocity of labels, and the classification effect is often difficult to be satisfactory. The convolutional neural network multi-label classification algorithm based on label correlation proposed in this paper solves this problem by adding the correlation between different labels.

2.4. GCN Method

The co-occurrence relationship between labels is usually represented in the form of a graph. The formula $G =< V, E >$ is used to represent the dependency co-occurrence relationship of the class label node. V indicates all label nodes, and E indicates the dependency between two label nodes. As an effective feature extractor based on graph structure, graph convolutional neural network (GCN) has been widely used in node classification, graph classification, link prediction and so on. The input to GCN contains the node characteristics and the correlation matrix between these nodes. GCN updates the characteristics of a node by aggregating the characteristics of its neighbors and itself into the node. The process of graph propagation can be expressed as follows:

$$X^{l+1} = \alpha^l(\bar{A}X^lW^l). \tag{1}$$

Where, X^l, W^l, and α^l respectively represent the input features, weight parameters, and nonlinear activation functions of the convolution layer of the $l-th$ graph. $\bar{A}$ represents the normalized form of the correlation matrix:

$$\bar{A} = \bar{D}^{-0.5}\bar{A}\bar{D}^{-0.5}. \tag{2}$$

Where $\bar{A} = A + I$. I is the identity matrix. $\bar{D}$ is a diagonal matrix and $\bar{D} = \sum_j \bar{A_{ij}}$.

GCN represents the word vector to be trained as the correlation matrix constructed by the features of the label nodes in the graph and the statistics of the sample labels as the input of the graph convolutional neural network, learns semantic label embedding through the two-layer graph convolutional neural network, and uses it as an interdependent object

classifier in the prediction stage. At the same time, the feature maps extracted by convolutional neural network of sample images are globally maximized, and the resulting image features and label co-occurrence features extracted by convolutional network are fused in the form of matrix multiplication to obtain the final classification result.

3. Proposed Multi-label Educational Emotion Prediction Model

GCN-LSTM is mainly composed of four parts: statement feature vector generation module, aspect feature vector generation module, multi-granularity attention computing module, and emotion classification module integrating aspect level features. The specific block diagram is shown in Figure 1. The statement feature vector generation module in Figure 1 uses the LSTM network to obtain the statement feature vector containing word order information. Aspect feature vector generation module uses the GCN layer constructed in this paper to obtain the aspect feature vector integrating context information by word masking. The multi-granularity attention computing module uses the output of the above two modules as input, and cross-calculates the fine-grained attention and the coarse-grained attention based on double branches by vector product and pooling methods. The multi-granularity attention weight is obtained by concatenating and normalizing the calculation results of the two kinds of granularity attention. The sentence emotion classification module uses the feature vector and multi-granularity attention weight to calculate the final emotion expression of the sentence, and uses the full connection layer and softmax classifier to get the aspect level emotion classification results.

3.1. Semantic Information Expression Based on GCN

In this paper, two word vector generation methods are adopted: static word vector generated based on Stanford University pre-trained Global Vectors for Word Representation (GloVe) and dynamic word vector generated based on Google pre-trained BERT language model. In this paper, we sum the last four hidden layer vectors of the BERT model to get the sub-word vector, and use the embedding vector of the average sub-word to generate an approximate vector for the original word.

We use $x = [x_1, \cdots, x_{\tau+1}, \cdots, x_{\tau+M}, \cdots, x_N]$ denotes a statement of length N mapped word by word to a fixed real-valued vector in a low-dimensional space. The subsequence $[x_{\tau+1}, \cdots, x_{\tau+M}]$ of x represents the aspect word of length M. $x_i \in R^d$ represents the word vector of the word i, and d represents the dimension of the word vector.

Graph convolution describes dependencies between words using adjacencies between nodes in a graph structure. A node in the undirected graph G represents a word, the edge between the nodes represents the dependency between the two words, and the adjacency matrix A of G is the syntactic dependency graph. If the input statement is "World peace leads to economic development.", the dependency diagram for the statement is shown in Figure 2. The values on the subdiagonal of A are all 1. If there is a direct dependence between the two words described by node i and node j, indicating that the two nodes have edges in the graph G, then $A_{ij} = A_{ji} = 1$; Otherwise, $A_{ij} = A_{ji} = 0$.

GCN simply uses aspect word features that extract and fuse contextual information. However, because the mean and variance of the feature vector containing the semantic

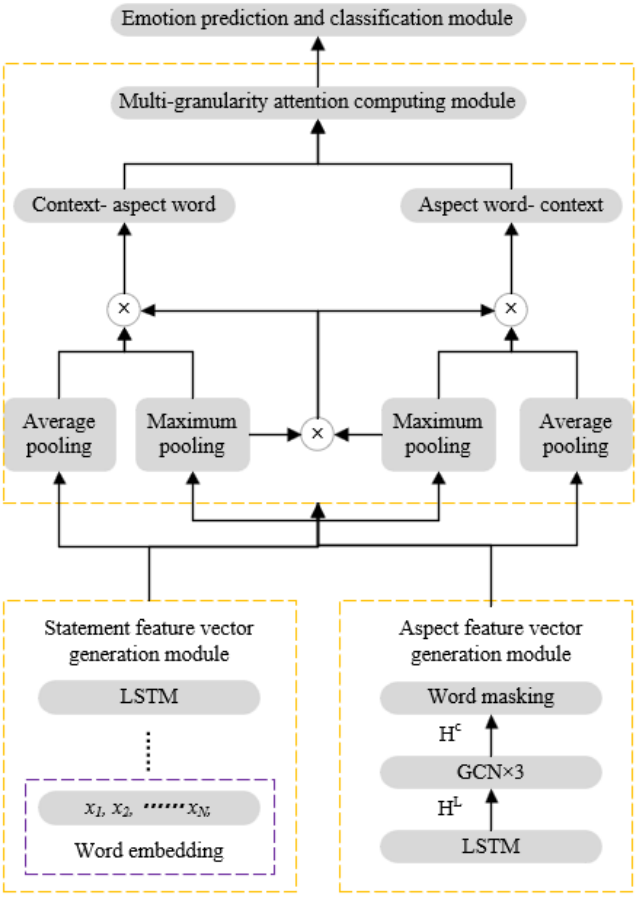

Fig. 1. Proposed model block diagram

.	0	0	0	0	0	0	1
development	0	0	1	0	0	1	0
economic	0	0	1	0	1	0	0
to	0	0	1	1	0	0	0
leads	1	1	1	1	1	1	0
peace	0	1	1	0	0	0	0
World	1	0	1	0	0	0	0
	World	peace	leads	to	economic	development	.

Fig. 2. Dependency graph of the sentence sample

and syntactic information of the statement will change after the nonlinear transformation of the graph convolution, the context information is lost during the forward propagation of GCN. In order to further reduce the lost input information in the output of each layer of graph convolution, this paper uses adaptive scaling strategy [35] to add PD unit to GCN in the aspect feature vector generation module and build CPGCN layer, so as to carry out integrated training on the output of GCN of the local layer and CPGCN of the previous layer. The internal structure of CPGCN layer is shown in Figure 3.

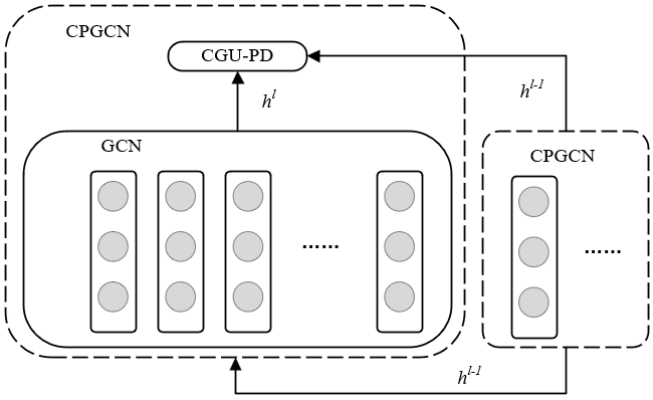

Fig. 3. Internal structure of improved GCN

In Figure 3, h^l is the output of GCN in this layer. If this layer CPGCN is the first layer, h^{l-l} is the output of LSTM hidden layer; Otherwise, h^{l-l} is the output of the previous CPGCN layer.

The CPGCN layer based on CGU-PD and GCN can dynamically adjust the weight of the input vector and the vector transformed by GCN in the output of this layer. The output vector of $l-th$ layer CPGCN is:

$$\eta_i^l = \sigma(W_{gate}^l h_i^l + b_{gate}^l). \tag{3}$$

$$h_i^l = \eta_i^l \odot h_i^l + (1 - \eta_i^l) \odot h_i^{l-1}. \tag{4}$$

Where h_i^l is the output of node i in $l-th$ layer CPGCN. h_i^{l-1} is the output of node i in CPGCN at layer $l-1$. η_i^l is the gating weight of node i in $l-1$ layer CPGCN, which is obtained by h_i^l training. σ is the sigmoid activation function. W_{gate}^l and b_{gate}^l are trainable weight matrices and deviations in the calculation of gated weights, respectively. The $\odot$ dot product represents the multiplication of vectors by their corresponding bits. After CGU-PD transformation, the output of the last CPGCN layer is:

$$H^C = [h_1^C, \cdots, h_{\tau+1}^C, \cdots, h_{\tau+M}^C, \cdots, h_N^C]. \tag{5}$$

Where $h_i^C \in R^{2d_L}$ is the dimension of the hidden state vector of the unidirectional LSTM output. $[h_{\tau+1}^C, h_{\tau+2}^C, \cdots, h_{\tau+M}^C]$ corresponds to the aspect word in the statement.

CPGCN integrates the output of GCN and its upper layer CPGCN to effectively reduce information loss during network propagation and retain the context information of statements more completely. Compared with the common LSTM foundation network, in the final output vector of CPGCN, each word can be fused into the comprehensive semantic, syntactic and interdependent information of a larger range of nodes. Finally, by shielding the non-aspect words in the final output of CPGCN, the aspect feature vector of fusion context information is obtained:

$$H^C_{mask} = [0, \cdots, 0, (h^C_{\tau+1})_{mask}, \cdots, (h^C_{\tau+M})_{mask}, 0, \cdots, 0]. \tag{6}$$

3.2. Context-dependent Learning Based on Deep LSTM Model

Core modules such as user emotion change and influence change relationship are the basis of the entire temporal network. Combined with the advantages of LSTM correlation and easy training, as well as the characteristics of deep network strong expression ability, the deep LSTM module is designed, as shown in Figure 4. The transfer relationship of variables is as follows:

$$z_{t+1} = f^1_{deep}(H_t, X_{t+1}, R_{t+1}). \tag{7}$$

$$r_{t+1} = f^2_{deep}(H_t, X_{t+1}, R_{t+1}). \tag{8}$$

$$\tilde{H_{t+1}} = f^3_{deep}(H_t \cdot r_{t+1}, X_{t+1}, R_{t+1}). \tag{9}$$

$$H_{t+1} = (1 - z_{t+1}) \cdot H_t + z_t \cdot \tilde{H_{t+1}}. \tag{10}$$

Where f^1_{deep}, f^2_{deep}, f^3_{deep} are deep neural networks, which are designed in the form of short-circuit modules. Compared with the classical LSTM, the linear part is replaced by the deep residual neural network, and the state is more compact. The input, R_{t+1} (processed observation data), and the previous state, through f^1_{deep}, f^2_{deep} becomes two activation quantities z_{t+1}, r_{t+1}, which are used to modulate the influence of the state on the new intermediate state $\tilde{H_{t+1}}$ (generated by the deep network f^3_{deep}), and the contribution of the new intermediate state $\tilde{H_{t+1}}$ and state H_t to the final new state, respectively.

X, I represent the observation data and the data after processing. H, A, R represent state variables. f represents all kinds of mapping functions. θ is the model parameter. $X_{u_i,d_m,t}$ represents class d_m observation data of user i at time t, such as uploaded pictures, texts, videos, etc. $X_{u_i,t}$ represents the summary vector of user i observation results at time t, output by f_{AT}. $I_{u_i,u_i}t$ is the interaction between user j and i at time t. For example, j leaves a message to i.

In the aspect of obtaining label correlation, the same way as GCN is used to construct the feature module of extracting label relation. The module uses two layers of GCN learning to embed the label dependencies of the training set. The natural language processing algorithm GLOVE4 is used to generate a 300-dimensional word vector for each class label (there are C class labels in total) to generate the label feature matrix $X \in R^{C\times 300}$. In addition, the correlation matrix $A \in R^{C\times C}$ is constructed based on the statistics of

category labels, that is, the number T_i of the occurrence of the $i-th$ category label o_i in the data set and the number T_{ij} of the co-occurrence of the $i-th$ category label $j-th$ and the j category label o_j are counted. Therefore, conditional probabilities can be used to generate dependencies for class labels:

$$P_{ij} = P(o_i|o_j) = \frac{T_{ij}}{T_J}. \tag{11}$$

Where P_{ij} represents the probability that category label i appears when category label j appears. Therefore, category label correlation matrix A can be defined as:

$$A_{ij} = P_{ij}. \tag{12}$$

GCN learns the relationship between different category labels based on category label feature matrix X and category label correlation matrix A to extract the features of label nodes. Using ReLu as the activation function of the first layer, the working principle of two-layer GCN can be expressed as follows:

$$W = f(X, A) = \bar{A}LeakyRelu(\bar{A}XW^0)W^1. \tag{13}$$

Where, X is the label feature matrix. A is the label correlation matrix. $\bar{A}$ is the normalized form of the label correlation matrix. W^0 and W^1 are the learning parameters of the first and second layer graph convolutional neural networks respectively. W is the interdependent object classifier learned through two layers of GCN.

After label feature matrix X and label correlation matrix A pass through two layers of GCN, the learned interdependent object classifier corresponds to the label co-occurrence embedded classifier $W \in R^{C\times D}$ in the image feature and label relationship feature fusion module. The dimension of the final residual attention feature $\bar{f}$ obtained from the multi-head class specific residual attention module is adjusted to R^D, and then the learned classifier W is applied to the residual attention feature to obtain the prediction score of the multi-label image:

$$\bar{y} = W\bar{f}. \tag{14}$$

Assuming the true label $y \in R^C$ of an image, where $y^i = 0, 1$ indicates whether the label appears in the image, the entire network is trained using traditional multi-label classification losses:

$$L = \sum_{c=1}^{C} y^c \log(\sigma(\bar{y_c})) + (1 - y^c) \log(1 - \sigma(\bar{y_c}). \tag{15}$$

Where $\sigma(\cdot)$ is the sigmoid function.

3.3. A Coarse-grained Attention Computation Based on Two Branches

The two-branch-based coarse-grained attention computation method (CGAC) mainly focuses on the interaction between specific aspect and context at the phrase level, and consists of two sets of symmetrical aspect words (CGAC-avg/CGAC-max)-context and

context-aspect words. In CGAC, the average pooling method is used to extract the comprehensive information of specific aspects and complete sentences, and the maximum pooling method is used to extract the representative information of both. Finally, the vector product is used to allocate attention to words in complete statements or specific aspects. The use of maximum pooling can further reduce the influence of redundant information and noise in feature vectors.

Aspect words-context focuses on describing the degree of attention of different context words to specific aspects, and is used to calculate the coarse-grained attention of each word to a certain aspect. Since a particular aspect may cover multiple words, the average semantic vector is generated by means of average pooling for the aspect feature vector that integrates context information, and the features of all words are integrated. At the same time, the main feature vector is generated by means of maximum pooling. The vector product is used to calculate the average semantic vector, the main feature vector and the semantic correlation between the words in the sentence, so as to allocate attention to all the words. Aspect word-context coarse-grained attention is calculated as follows:

$$s_i^{cA2C-avg} = (H_{mask}^C)_{avg}(h_i^L)^T. \tag{16}$$

$$(H_{mask}^C)_{avg} = \frac{1}{M}\sum_{j=\tau+1}^{\tau+M} h_j^C. \tag{17}$$

$$s_i^{cA2C-max} = (H_{mask}^C)_{max}(h_i^L)^T. \tag{18}$$

$$H_{mask}^C)_{max} = max_{d=0}^{2d_L}(H_{mask}^C)_d. \tag{19}$$

Where, $H_{mask}^C \in R^{2d_L \times N}$ represents an aspect feature vector that fuses context information. $(H_{mask}^C)_{avg}$ represents the average semantic vector of a particular aspect. $H_{mask}^C)_{max}$ represents the main feature vector of a particular aspect. h_i^L represents the eigenvector of the word i that contains the statement word order information. $max_{d=0}^{2d_L}$ indicates the maximum value in each dimension. $s_i^{cA2C-avg}$, $s_i^{cA2C-max}$ represent aspect word-context coarse-grained attention assigned by aspect mean semantic vector and aspect main feature vector for word i, respectively.

Context-aspect words focus on describing the influence of context on the semantic expression of each word in a specific aspect, and are used to calculate the coarse-grained attention of each word in a specific context. In this paper, the average semantic vector is generated by means of average pooling to integrate the semantic information of the whole sentence. At the same time, the maximum pooling method is used to generate the main feature vector, and the vector product is used to calculate the average semantic vector, the main feature vector and the semantic correlation between the words in the specific aspect, so as to allocate attention to the words in the specific aspect. Context-aspect coarse-grained attention is calculated as follows:

$$s_i^{cC2A-avg} = (H^L)_{avg}((h_i^C)_{mask})^T. \tag{20}$$

$$(H^L)_{avg} = \frac{1}{N}\sum_{j=1}^{N} h_j^L. \tag{21}$$

$$s_i^{cC2A-max} = (H^L)_{max}((h_i^C)_{mask})^T. \tag{22}$$

$$(H^L)_{max} = max_{d=0}^{2d_L}(H^L)_d. \tag{23}$$

Where H^L represents the statement feature vector that contains word order information. $(H^L)_{avg}$ represents the average semantic vector of the statement. $H^L)_{max}$ represents the main feature vector of the statement. $(h_i^C)_{mask})$ represents the feature vector of the word i that incorporates contextual information in a particular aspect. $s_i^{cC2A-avg}$ and $s_i^{cC2A-max}$ represent the context-aspect coarse-grained attention assigned by the sentence average semantic vector and the statement main feature vector to the word i in a particular aspect, respectively.

Fine-Grained Attention Computing Method (FGAC) [36] focuses on the interplay of specific aspects and context at the word level. When calculating coarse-grained attention, the pooling method is used to calculate the average semantic vector and the main feature vector, which leads to the loss of some information. In this paper, the aspect-based fine-grained attention computing method uses the product of the eigenvector of aspect and statement to calculate the semantic correlation between the words in the statement and the statement, so as to allocate attention to all words and make up for the information loss caused by coarse-grained attention computing. The aspect-based fine-grained attention is calculated as follows:

$$s_i^{fg} = \sum_{j=1}^{N}(h_j^C)_{mask}(h_i^L)^T = \sum_{j=\tau+1}^{\tau+M}(h_j^C)_{mask}(h_i^L)^T. \tag{24}$$

Where, h_j^C represents the feature vector of the word j that incorporates contextual information in a particular aspect. h_i^L represents the eigenvector of the word i that contains word order information. s_i^{fg} represents the fine-grained attention of the word i.

In this paper, the multi-grained attention of the emotion classification model is obtained by splicing and averaging the coarse-grained attention and fine-grained attention based on double branches. Finally, through the normalization operation of multi-granularity attention, the multi-granularity attention weight is obtained:

$$s_i^{multi} = [s_i^{fg}; [s_i^{cA2C-avg}; s_i^{cA2C-avg}]_{avg}; [s_i^{cA2C-maxg}; s_i^{cA2C-max}]_{avg};]_{avg}. \tag{25}$$

$$\alpha_i = \frac{s_i^{multi}}{\sum_{k=1}^{N} exp(s_i^{multi})}. \tag{26}$$

Where s_i^{multi} represents the multi-grained attention of the word i. $[\cdot]_{avg}$ indicates attention splicing and averaging. α_i represents the multi-granularity attention weight of the word i.

The emotion classification module of GCN-LSTM screens the emotion features related to specific aspects from the LSTM hidden state vector H^L of the statement according to the multi-granularity attention weight α_i. The aspect level emotion expression of the statement is:

$$e = \sum_{i=1}^{N} \alpha_i h_i^L. \tag{27}$$

3.4. Network Training Process

The GCN network training process can be divided into two stages: forward propagation and back propagation. Forward propagation is used to transfer the feature information of the image, and back propagation is used to transfer the error information in reverse to update the network weight.

1. Forward propagation. First, the samples in the data set are output to the network, and the weights, biases and error thresholds of the samples are initialized by using the initiation-V3 model. The output results of forward propagation are calculated by layer by layer transformation through the neural network:

$$x^i = f(W^i x^{i-1} + b^i). \tag{28}$$

   Where x^i represents the output of the current layer, f represents the activation function through relu, W represents the weight, b represents the bias, and x^{i-1} represents the input of the current layer.
2. Back propagation. The core of backpropagation is mainly divided into two steps: the error is transmitted to the previous layer in reverse, and the difference that needs to be updated is calculated according to the learning parameter expression corresponding to the current error. The error function expression of backpropagation is as follows:

$$E^n = 0.5 \sum_{k=1}^{c} (o_k^n - y_k^n)^2 = 0.5 ||o^n - y^n||_2^2. \tag{29}$$

   Where, n is the data sample. c represents the number of output nodes. o represents the expected output value of the training sample. y represents the actual output value of network training. Let $X = (x_i, y_i)|i = 1, 2, \cdots, m$ means that there are m samples. $x_i = [x_{i1}, x_{i2}, \cdots, x_{id}]$ indicates that the $i - th$ sample has d eigenvectors. $y_i = [y_{i1}, y_{i2}, \cdots, y_{iq}]$ means that the $i - th$ sample has q label vectors. If $y_{ij} = 0$, there is no label l_j. In single-label object classification, a picture has a one-to-one relationship with the object label, and a Softmax is used after convolution features. The multi-label classification problem is a one-to-many relationship. Suppose there are c classes of objects, each labeled with probabilities $p_i = (P_{i1}, P_{i2}, \cdots, P_{ic})$. Using the softmax_cross_entropy_with_logits method in tensorflow, we select the largest labels in the probability distribution. According to the calculated co-occurrence relationship between labels, a certain threshold is set. The probability of labels that are larger than the threshold is retained, and the probability of labels that are smaller than the threshold is deleted. Finally, multiple labels are output. The specific flow of Softmax in GCN-LSTM is shown in Figure 4.

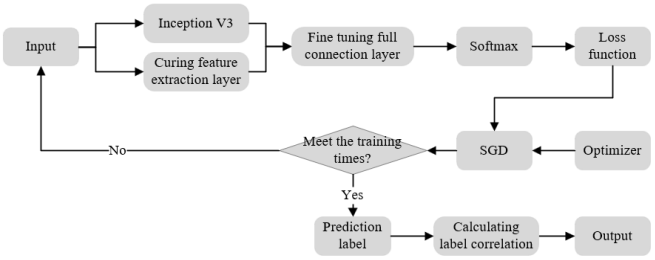

Fig. 4. Specific flow of Softmax in GCN-LSTM

4. Experimental Results and Analysis

4.1. Experimental Environment and Data Set

The experimental operating system is Ubuntul6.04.3LTS,16G memory, Core i5-7200U CPU, and 4G graphics card. The programming language is Python 3, the development tool is Pycharm, and the deep learning framework is Tensorflow-1.10 [37,38].

Since there are few well-labeled multi-label sentiment analysis datasets, the NLP&CC2013 Weibo sentiment analysis dataset is selected in this paper. The data set divides emotions into eight categories: none, dislike, liking, happiness, fear, sadness, anger, and surprise. There are 14000 marked Weibo corpora, each with one or two emotional tags. The training set has 4000 data and the test set has 10000 data. However, in the labeled test set, the distribution of emotional labels is uneven, and there are more than 5000 corpus without emotional labels. Therefore, the data set is re-labeled.

A. Data set annotation rule

Because everyone's criteria for emotion discrimination are inconsistent, it is easy to appear completely different labeling results. Therefore, in the process of annotation, this paper adopts the method of three-person independent annotation and collaborative cross-validation, that is, each person first annotates part of the corpus independently, and then randomly extracts the same number of text corpus from the annotated corpus of the three annotators for cross-evaluation. If the annotation accuracy is above 90%, the annotation results of the extracted text corpus are valid.

B. Data set label distribution

The label distribution of the re-labeled NLP&CC2013 dataset is shown in Table 1. The data in the table is the percentage of the two emotional labels in the total number of labels. This paper does not distinguish the primary and secondary relationship of emotions in the text, but only explores which emotional labels are included in the text. As can be seen from Table 1, about one-third of the text contains only one label, for which a label completion operation is required. Emotional labels of different polarity co-existed less, such as "dislike" and "like", "happy" and "angry", etc., while emotional labels of the same polarity co-existed more, such as "like" and "happy", "disgust" and "angry". Finally, the re-labeled NLP&CC2013 data set is divided and tested. The training set consists of 12000 corpus, the verification set of 1000 corpus, and the test set of 1000 corpus.

Table 1. Emotion label distribution statistics/%

	none	disgust	like	happy	fear	sad	angry	surprise
none	2.4	4.3	4.2	5.7	3.7	5.4	4.5	4.2
disgust			0.3	0.2	4.8	4.6	5.9	3.3
like				9.8	0	0.3	0.1	3.7
happy					0.1	0.1	0	3.5
fear						5.5	6.2	4.1
sad							3.5	4.7
angry								4.9
surprise								

4.2. Experimental Setup

A. Education text preprocessing

It mainly includes text standard processing and word segmentation. Standard processing includes denoising by regular, converting traditional characters into simplified ones and so on. Word segmentation is performed on the re-labeled NLP&cc2013 data set by stuttering word segmentation package. Finally, a dictionary is built to implement text preprocessing. In the process of establishing the dictionary, words that appear less than 5 times are filtered out, and words that do not appear in the dictionary are replaced by $< unk >$.

B. Word vector training

Word embedding model Word2Vec converts words into a low-dimensional real vector by using the relationship between words and context, which can effectively distinguish between a word with multiple meanings or multiple words with one meaning. The Skip-gram model in Word2Vec is used in this article. The default size of the text word vector is 256, the size of the label word vector is 128, and the default value of the maximum distance window of the word vector context is 8. This value can be dynamically adjusted according to the size of the corpus during training. The default value of the maximum number of iterations in stochastic gradient descent is 5, which can be increased for large data.

C. Setting of network model parameters

Table 2. Parameters in GCN

Parameter	value
Word vector dimension	200
Sliding window size	3,4,5
Number of sliding Windows	128
Activation function	ReLU
Epoch	20
Optimizer	Adam

Table 3. Parameters in LSTM

Parameter	value
Word vector dimension	200
Learning rate	0.001
Hidden layer size	256
Loss function	Cross entropy
Epoch	20
Optimizer	Adam

4.3. Evaluation Index

In order to evaluate the performance of GCN-LSTM, four widely used performance evaluation indexes of multi-label learning, including Hamming loss (HL), one-error rate (OE), ranking loss (RL) and average precision (AP), are adopted.

Hamming loss (HL). It is used to evaluate the proportion of inconsistency between the prediction label and the relevant real label. The smaller HL denotes the more accurate the prediction result. The calculation is shown in Equation (30).

$$HL = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{Q}|h(x_i \Delta y_i|. \tag{30}$$

Where Δ represents the symmetric difference between the set of predicted labels and the set of real labels. N is the number of test set samples. $h(x_i$ represents the prediction of sample x_i. Q is the dimension of the label space. y_i indicates the real label.

One-error rate (OE). It is used to evaluate the probability that the first label in the predicted result is not in the set of real labels, calculated as equation (31).

$$OE = \frac{1}{N}\sum_{i=1}^{N}[[arg_{y_i} minY_i \in Z_i]]. \tag{31}$$

Where, $[[\cdot]]$ means that the result returns 0 when the prediction is correct and 1 when the prediction is wrong. Y_i indicates the predicted label result. Z_i is the true label result. $arg_{y_i} minY_i$ indicates the label at the front of the sample. N is the number of test set samples.

Ranking loss (RL). It is used to estimate the average number of times an incorrect label appears before a correct label in a sort sequence in a predicted result label set. The calculation is shown in equation (32).

$$RL = \frac{1}{N} \times \sum_{i=1}^{N}\frac{|(l_k, l_j)|f_k(x_i)| \leq f_j(x_i)|}{|y_i||\bar{y_i}|}. \tag{32}$$

Where $\bar{y_i}$ is the complement of the set y_i in the label space. $f()$ is a real-valued function corresponding to a multi-label classifier. l_k is the prediction label. l_j indicates the true label. N indicates the number of test set samples.

Average precision (AP). It is used to measure the average number of correct sorting in the prediction result label. The calculation is shown in equation (33).

$$AP = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|y_i|}\sum_{l_k}\frac{A = |l_j|rank(x_i, l_j) \leq rank(x_i, l_k)|}{rank(x_i, l_k)}. \tag{33}$$

Where y_i is the label space set. l_k is the prediction label. l_j indicates the real label. x_i is the prediction label probability. N indicates the number of test set samples. The larger the average accuracy value, the more accurate the prediction result.

4.4. Experiment Results Comparison

In order to evaluate the performance of the model combined with GCN and LSTM, the new model is compared with a single CNN and a single LSTM. During the training process, the condition of early stopping of the model is set, and the model is saved when the validation set $val - acc$ is no longer increased for three rounds. The comparison results of the three models are shown in Figure 5. The new model performs well on the test set.

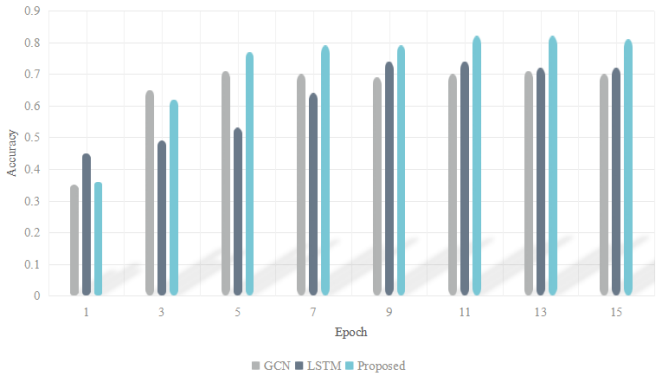

Fig. 5. Accuracy comparison of GCNN, LSTM and proposed.

In order to evaluate the performance of GCN-LSTM model, it is compared with four multi-label classification algorithms including: Unigram+Context [39], FastText [40], Transformer model [41] and PPS [42].

1. Multi-label classification method combining Context and text features (Unigram+Context). This method uses a base classifier to analyze the microblog text and get the initial emotion classification result of each sentence. Then the transfer probability of adjacent sentences is used to modify the sentence emotion category.
2. FastText model. Compared with the traditional word embedding method, this model adds the embedding of character rgram, it can contain the local features of words, and has a very fast training speed. The input to the model is a sequence of words, using hierarchical softmax functions to calculate probabilities on predefined classes, and using cross entropy to calculate losses.

3. Transformer model. The model consists of encoder and decoder. Different from the traditional CNN and RNN, self-attention can effectively solve the problem of long distance dependent features in NLP tasks.
4. PPS method. Compared with the proposed method in this paper, the complete process of label matrix is less, and the correlation between labels is not considered.

The comparison results of the five methods are shown in Figure 6 and Table 4. It can be seen that, on the test set, the method that uses word unary grammar features as the main features and combines context to predict microblog emotion is less effective than the neural network method, and the accuracy rate is about 70%. The PPS method has an accuracy rate of 76%. FastText convergence speed is faster, the accuracy is lower than GCN-LSTM method, up to 72%. The convergence rate of Transformer method is slower than other methods, and the final accuracy rate is 73%. However, the method in this paper, GCN-LSTM, can slightly improve the accuracy of the whole process of adding label related information, and the final accuracy rate is 79%.

Table 4. Comparison with different methods

Method	1	3	5	7	9	11	13	15
PPS	0.19	0.42	0.69	0.75	0.74	0.72	0.73	0.74
FastText	0.24	0.61	0.70	0.69	0.72	0.69	0.68	0.70
Transformer	0.10	0.29	0.45	0.55	0.65	0.66	0.75	0.71
GCN-LSTM	0.12	0.43	0.71	0.75	0.77	0.79	0.76	0.75

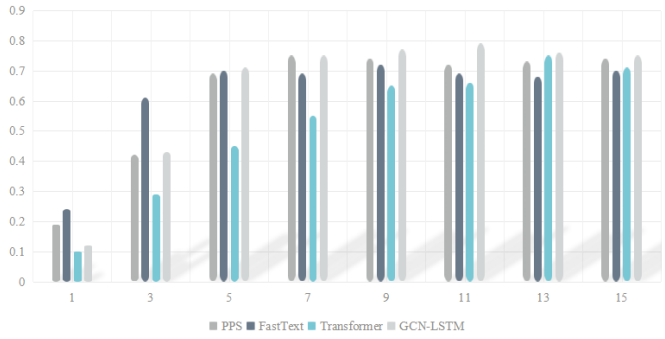

Fig. 6. Comparison with different methods.

In Table 5, the smaller value of "↓" denotes the better classification effect, and the larger value of "↑" denotes the better classification effect. As can be seen from Table 5, the GCN-LSTM model proposed in this paper performs relatively well in terms of average accuracy (AP) and Hamming loss (HL). Compared with the method of using neural network, the method of using unary feature and combining context has some shortcomings

in each index. The FastText method can obtain local word order information using N-gram. The Transformer method addresses sentence length dependency. The deep learning method PPS not only extracts local features, but also extracts global semantic information, and can complete multi-label sentiment classification (MLC) more accurately. These four methods do not consider the correlation between labels, but the GCN-LSTM model proposed in this paper can combine the local features of text with the global semantic information, and also consider the correlation between labels. The experimental results show that GCN-LSTM has a good multi-label classification effect.

Table 5. Experimental comparison results

Model	HL↓	OE↓	RL↓	AP↑
Unigram+Context	0.264	0.283	0.238	0.693
FastText	0.243	0.328	0.265	0.714
Transformer	0.239	**0.213**	0.206	0.725
PPS	0.241	0.297	**0.192**	0.750
GCN-LSTM	**0.224**	0.269	0.276	**0.784**

5. Conclusion

In this paper, we propose a multi-label classification method called GCN-LSTM, which combines GCN and LSTM neural networks to extract local features and global semantic information effectively. The Attention mechanism is used to assign higher weight to educational sentiment words. At the same time, the label correlation matrix corresponding to each text instance is completed by using the label correlation matrix, and the correlation between labels is investigated. That is, if only one emotion label is marked in a text, whether the emotion label with greater correlation is suitable for the text. Label completion of a single emotion label is beneficial to improve the accuracy of multi-label classification. Experiments show that the model presented in this paper performs well on the re-labeled NLP&CC2013 dataset. Future works will focus on more advanced method and apply them in real engineering applications.

Acknowledgments. This work was supported by the Key Soft Science Research Project of Henan Provincial Science and Technology Department in 2023 (232400411183).

References

1. Liu X, Li N, Xia Y. Affective image classification by jointly using interpretable art features and semantic annotations[J]. Journal of Visual Communication and Image Representation, 2019, 58: 576-588.
2. Bayrakdar S, Yucedag I, Simsek M, et al. Semantic analysis on social networks: A survey[J]. International Journal of Communication Systems, 2020, 33(11): e4424.
3. Tamil Priya D, Divya Udayan J. Transfer learning techniques for emotion classification on visual features of images in the deep learning network[J]. International Journal of Speech Technology, 2020, 23: 361-372.

4. Dias L L, Barrére E, de Souza J F. The impact of semantic annotation techniques on content-based video lecture recommendation[J]. Journal of Information Science, 2021, 47(6): 740-752.
5. Yin S. Object Detection Based on Deep Learning: A Brief Review[J]. IJLAI Transactions on Science and Engineering, 2023, 1(02): 1-6.
6. Jiang, Y., Yin, S.: Heterogenous-view Occluded Expression Data Recognition Based on Cycle-Consistent Adversarial Network and K-SVD Dictionary Learning Under Intelligent Cooperative Robot Environment. Computer Science and Information Systems, vol. 20, no. 4, 2023. https://doi.org/10.2298/CSIS221228034J
7. Zhang H, Xu M. Weakly supervised emotion intensity prediction for recognition of emotions in images[J]. IEEE Transactions on Multimedia, 2020, 23: 2033-2044.
8. Liu T, Wan J, Dai X, et al. Sentiment recognition for short annotated GIFs using visual-textual fusion[J]. IEEE Transactions on Multimedia, 2019, 22(4): 1098-1110.
9. L. Teng et al., "FLPK-BiSeNet: Federated Learning Based on Priori Knowledge and Bilateral Segmentation Network for Image Edge Extraction," in IEEE Transactions on Network and Service Management, vol. 20, no. 2, pp. 1529-1542, June 2023, doi: 10.1109/TNSM.2023.3273991.
10. Zhang H, Xu D, Luo G, et al. Learning multi-level representations for affective image recognition[J]. Neural Computing and Applications, 2022, 34(16): 14107-14120.
11. Parwez M A, Abulaish M. Multi-label classification of microblogging texts using convolution neural network[J]. IEEE Access, 2019, 7: 68678-68691.
12. Wang Q, Hao Y. ALSTM: An attention-based long short-term memory framework for knowledge base reasoning[J]. Neurocomputing, 2020, 399: 342-351.ntreal, Canada[J]. International Journal of Transportation Science and Technology, 2022, 11(2): 298-309.
13. Acheampong F A, Wenyu C, Nunoo-Mensah H. Text-based emotion detection: Advances, challenges, and opportunities[J]. Engineering Reports, 2020, 2(7): e12189.
14. Reza S, Ferreira M C, Machado J J M, et al. A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks[J]. Expert Systems with Applications, 2022, 202: 117275.
15. Kumar A, Verma S, Sharan A. ATE-SPD: simultaneous extraction of aspect-term and aspect sentiment polarity using Bi-LSTM-CRF neural network[J]. Journal of Experimental & Theoretical Artificial Intelligence, 2021, 33(3): 487-508.
16. Fadel A S, Saleh M E, Abulnaja O A. Arabic aspect extraction based on stacked contextualized embedding with deep learning[J]. IEEE Access, 2022, 10: 30526-30535.
17. Li X, Bing L, Lam W et al. Transformation networks for target-oriented sentiment classification. In: Proceedings of the 56th annual meeting of the association for computational linguistics (ACL), Melbourne, VIC, Australia, 15-20 July 2018, pp. 946-956. Stroudsburg, PA: ACL.
18. Zhang S, Tong H, Xu J, et al. Graph convolutional networks: a comprehensive review[J]. Computational Social Networks, 2019, 6(1): 1-23.
19. Zhu X, Zhu L, Guo J, et al. GL-GCN: Global and local dependency guided graph convolutional networks for aspect-based sentiment classification[J]. Expert Systems with Applications, 2021, 186: 115712.
20. Jang B, Kim M, Harerimana G, et al. Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism[J]. Applied Sciences, 2020, 10(17): 5841.
21. Kamyab M, Liu G, Adjeisah M. Attention-based CNN and Bi-LSTM model based on TF-IDF and glove word embedding for sentiment analysis[J]. Applied Sciences, 2021, 11(23): 11255.
22. Li W, Qi F, Tang M, et al. Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification[J]. Neurocomputing, 2020, 387: 63-77.
23. Liu M, Zhou F Y, He J K, et al. Self-attention networks and adaptive support vector machine for aspect-level sentiment classification[J]. Soft Computing, 2022, 26(18): 9621-9634.
24. Puthige I, Hussain T, Gupta S, et al. Attention Over Attention: An Enhanced Supervised Video Summarization Approach[J]. Procedia Computer Science, 2023, 218: 2359-2368.

25. Sun J, Han P, Cheng Z, et al. Transformer based multi-grained attention network for aspect-based sentiment analysis[J]. IEEE Access, 2020, 8: 211152-211163.
26. Ji Y, Zhou Z, Liu H, et al. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome[J]. Bioinformatics, 2021, 37(15): 2112-2120.
27. Liu H, Liu Y, Wong L P, et al. A hybrid neural network bert-cap based on pre-trained language model and capsule network for user intent classification[J]. Complexity, 2020, 2020: 1-11.
28. Mewada A, Dewang R K. SA-ASBA: A hybrid model for aspect-based sentiment analysis using synthetic attention in pre-trained language BERT model with extreme gradient boosting[J]. The Journal of Supercomputing, 2023, 79(5): 5516-5551.
29. Ahanin Z, Ismail M A, Singh N S S, et al. Hybrid feature extraction for multi-label emotion classification in English text messages[J]. Sustainability, 2023, 15(16): 12539.
30. Huang J, Li G, Huang Q, et al. Learning label-specific features and class-dependent labels for multi-label classification[J]. IEEE transactions on knowledge and data engineering, 2016, 28(12): 3309-3323.
31. Kwon O W, Lee J H. Web page classification based on k-nearest neighbor approach[C]//Proceedings of the fifth international workshop on on Information retrieval with Asian languages. 2000: 9-15.
32. Han K, Wang Y, Tian Q, et al. Ghostnet: More features from cheap operations[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 1580-1589.
33. Yoshimura T, Ojima M, Arai Y, et al. Three-dimensional self-organized microoptoelectronic systems for board-level reconfigurable optical interconnects-performance modeling and simulation[J]. IEEE journal of selected topics in quantum electronics, 2003, 9(2): 492-511.
34. Jisi A and Shoulin Yin. A New Feature Fusion Network for Student Behavior Recognition in Education [J]. Journal of Applied Science and Engineering. vol. 24, no. 2, pp.133-140, 2021.
35. Yang Q, Li Y, Gao X D, et al. An adaptive covariance scaling estimation of distribution algorithm[J]. Mathematics, 2021, 9(24): 3207.
36. Lv Z, Qiao L, Singh A K, et al. Fine-grained visual computing based on deep learning[J]. ACM Transactions on Multimidia Computing Communications and Applications, 2021, 17(1s): 1-19.
37. Liu Z, Jiang D, Zhang C, et al. A novel fireworks algorithm for the protein-ligand docking on the autodock[J]. Mobile Networks and Applications, 2021, 26: 657-668.
38. Liu Z, Zhang C, Zhao Q, et al. Comparative study of evolutionary algorithms for protein-ligand docking problem on the AutoDock[C]//Simulation Tools and Techniques: 11th International Conference, SIMUtools 2019, Chengdu, China, July 8-10, 2019, Proceedings 11. Springer International Publishing, 2019: 598-607.
39. Suciati A, Budi I. Aspect-based sentiment analysis and emotion detection for code-mixed review[J]. International Journal of Advanced Computer Science and Applications, 2020, 11(9).
40. Riza M A, Charibaldi N. Emotion Detection in Twitter Social Media Using Long Short-Term Memory (LSTM) and Fast Text[J]. International Journal of Artificial Intelligence & Robotics (IJAIR), 2021, 3(1): 15-26.
41. Lian Z, Liu B, Tao J. CTNet: Conversational transformer network for emotion recognition[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021, 29: 985-1000.
42. Zitouni M S, Park C Y, Lee U, et al. LSTM-Modeling of Emotion Recognition Using Peripheral Physiological Signals in Naturalistic Conversations[J]. IEEE Journal of Biomedical and Health Informatics, 2022, 27(2): 912-923.

Zhiguang Liu is with the School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology. Research direction:image processing, education data analysis, artificial intelligence.

Fengshuai Li is with the College of Civil and Architectural Engineering, Zhengzhou University of Science and Technology. Research direction: big data processing, pattern recognition.

Guoyin Hao is with the School of Music and Dance, Zhengzhou University of Science and Technology. Research direction: Image processing, data analysis, artificial intelligence.

Xiaoqing He is with the School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology. Research direction:image processing, education data analysis, artificial intelligence.

Yuanheng Zhang is with the School of Electronics and Electrical Engineering, Zhengzhou University of Science and Technology. Research direction:image processing, education data analysis, data analysis, artificial intelligence.

Received: March 14, 2024; Accepted: June 13, 2024.