doi:10.3772/j.issn.1006-6748.2021.03.011

Study on the fusion emotion classification of multiple characteristics based on attention mechanism①

Li Ying (李 颖), Shao Qing②, Hao Weichen
(School of Optoelectronic Information and Intelligent Engineering, University of Shanghai for Science and Technology, Shanghai 200093, P. R. China)

Abstract

The current research on emotional classification uses many methods that combine the attention mechanism with neural networks. However, the effect is unsatisfactory when dealing with complex text. An emotional classification model is proposed, which combines multi-head attention (MHA) with improved structured-self attention (SSA). The model makes several different linear transformations of input by introducing MHA mechanism and can extract more comprehensive high-level phrase representation features from the word embedded vector. Meanwhile, it can realize the parallelization calculation and ensure the training speed of the model. The improved SSA structure uses matrices to represent different parts of a sentence to extract local key information, to ensure that the degree of dependence between words is not affected by time and sentence length, and generate the overall semantics of the sentence. Experiment results show that the current model effectively obtains global structural information and improves classification accuracy.

Key words: multi-head attention (MHA), structured-self attention (SSA), emotion classification, deep learning, bidirectional long-short-term memory (BiLSTM)

0 Introduction

The rapid development of the modern Web, social media service providers give users a convenient way to share and create their own space. Weibo is a social platform with full participation, it has a wide range of topics, including life sharing, social event comments, star chasing, etc. Due to the simplicity and randomness, the number of Weibo users has surged across in recent years. Weibo posts contain a lot of emotional and opinion information.

Emotion classification is part of natural language processing (NLP). By using computational models to understand and classify the potential emotion, which helps people to identify social events and make better decisions quickly. Among the documented models, the neural network model is most commonly used for emotion classification. Most neural network models can encode sentences so that the output hidden layer contains contextual information about words. In the existing research, neural networks cannot completely process sequence relationships between sentences for longer text. In building language temporal features, the word information is sequentially fed back, which takes a long time and it is difficult to implement parallel computing. Moreover, the dependence between words in a sentence will decrease with the increase in distance and time. In addition, single attention calculation makes it impossible for the model to extract the dependencies and local features of words in a sentence. Attention feature extraction usually focuses on the specific components of a sentence, which cannot represent the overall semantic meaning of a sentence. The multi-head attention (MHA) mechanism can enable the model to selectively screen important information from a large amount of information without focusing on external knowledge, such as syntactic analysis, focusing on this information, and ignoring most of the less important information. When facing longer texts, important features of the text can be obtained more accurately. Therefore, an emotion classification model combining a MHA mechanism with an improved structured-self attention (SSA) structure is proposed. The main innovations include:

(1) Through MHA mechanism, learning attention

① Supported by the National Key Research and Development Program of China (No. 2018YFB1702601) and the Science and Technology Commission of Shanghai Municipality (No. 19511105103).

② To whom correspondence should be addressed. E-mail: sq_usst@yahoo.com.cn

Received on Aug. 26, 2020

representation under different transformations can more fully capture some syntactic or semantic features between words to improve the accuracy and efficiency of feature extraction.

(2) Utilize the improved SSA to represent all the components of a sentence with a matrix to capture the deeper local features and global semantics of the sentence.

1 Related research

1.1 Emotion classification

Recurrent neural network (RNN)[1] is specialized for sequential modeling, but traditional RNN has the problem of gradient explosion and gradient disappearance for long data sequences. Long short-term memory (LSTM)[2] is a special RNN architecture with long-term and short-term characteristics. The storage unit effectively solves the problem of vanishing gradient and gradient explosion. LSTM can extract high-level text information for natural language processing. Bidirectional long-short-term memory (BiLSTM)[3] is the advanced form of LSTM. Now, text classification achieved some results[4-7]. In Ref. [8] through the use of text analysis, automatic annotation of images was realized. Ref. [9] proposed a neural network model using sentence context causality when classifying emotion in text. In Ref. [10], the word level vector was converted to the character level vector for the Chinese setting, the accuracy was improved. Ref. [11] proposed a method to improve the word vector representation. They input the emotion information into the traditional term frequency-inverse document frequency (TF-IDF) algorithm to become weighted word vector, then input the word vector into BiLSTM model to catch the context information. Ref. [12] proposed a new model that combines with RNN and convolutional neural network (CNN). The data of Treebank1 and Treebank2 were used to verify the precision of the model. Ref. [13] added context-aware emotion to the model for training, the model learned more emotional information.

1.2 Attention mechanism

In natural language processing, Ref. [14] used attention mechanism to the field of natural language processing. Experiments proved the effectiveness of the attention mechanism in NLP tasks. The force mechanism connects each word in the source language with a word to be predicted currently, and the effect is greatly improved. At present, attention mechanisms have been applied to many aspects to solve problems in text analysis, including text generation[15], machine translation[16], and machine understanding[17]. Ref. [18] combined the MHA mechanism and CNN for sentence relation classification. The experimental results show that the model by using attention mechanism has more classification accuracy than the unused model. Ref. [19] proposed an LSTM network combining the attention mechanisms. By vectorising specific targets words and inputting specific targets words as attention mechanisms into the LSTM network, the accuracy of the model was improved. In 2017, Ref. [20] proposed a MHA mechanism. Self-attention captures the long-range dependence of a sentence by calculating the attention probability of each and every words. Through multiple self-attention calculations, each time the mapping matrix is changed. Finally, the results of each calculation are stitched together as the final multi-head calculation result. At present, the MHA mechanism can complete multiple tasks more efficiently than the RNN model. Ref. [21] used a new thought for extracting sentence embedding by adding in self-attention. A two-dimensional matrix is used instead of a vector to indication the embedding. Ref. [22] used the attention mechanism to divide the problem into sub-problems to be processed in parallel. Ref. [23] documented the face of sparse text, text classification was performed for the complex semantics of natural language processing. A model is proposed to extract features from word vectors by using convolutional layers, combined with BiLSTM to obtain forward and backward temporal relationships. Finally, the attention mechanism gives different weights to the information sent by the hidden layer of BiLSTM. Ref. [24] combined MHA mechanism and BiLSTM, which made the accuracy rate of emotional classification reach 92.11%.

2 MHA-SSA model

To extract more efficient text information, a MHA-SSA model is proposed. The model consists of the word embedding layer, the MHA layer, the improved SSA layer, and the softmax output layer. Fig. 1 illustrates the structure of the model.

The model first uses GloVe pre-trained word vectors to map words in the text as a real vector representation of low dimensions. The MHA layer of the model performs several different linear transformations according to the word vector matrix. This enables the model to capture the feature information of the context in many ways and incorporate the feature information into the improved SSA layer. The improved SSA layer can effectively capture the interrelation and important information between sentences and model text based on preser-

ving sentence feature information. Finally, the output is sent to the softmax layer to get the results of the emotional classification.

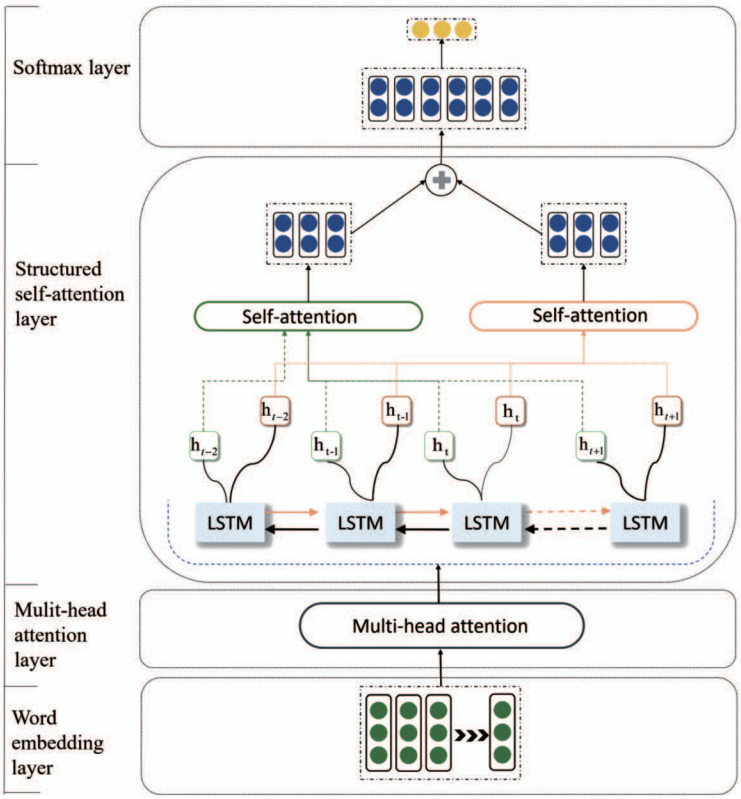

Fig. 1 MHA-SSA model structure

2.1 GloVe word embedding model construction

Since the neural network model can only accept numerical input, it needs to numerically represent the character text. Usually, the text is segmented and preprocessed to remove punctuation and stop words, and then trained with a word vector model to learn the context and semantic information and finally represent the word in the form of a vector and use it as the model input. This paper adopts the GloVe[25] word embedding model. Compared with the Word2vec model, it uses the local window information and fully considers the entire corpus information, such as the co-occurrence of words. The GloVe model focuses on how many meanings are generated by these co-occurrence statistics and how the final word vector represents these meanings. The model is trained based on the co-occurrence record matrix between words and words in the corpus. This matrix records the frequency of co-occurrence between words, and there is no element that is 0.

Compare the co-occurrence probability ratio of solid and gas to ice and steam with the co-occurrence probability ratio of water and fashion. It can be found that the co-occurrence probability ratio of solid relative to ice and steam is greater than 1, gas is much less than 1, and the ratio of ice steam is close to 1. It shows that the co-occurrence probability ratio can judge the relevance between words and distinguish the similarity between words.

For example, the co-occurrence matrix is represented by $\boldsymbol{X}$, X_{ij} represents the quantity of word j shows in the perspective of the word i. Use Eq. (1) X_i to express the number of times any word appears in the perspective of the word i. P_{ij} expresses the possibility that the word in order j appears in the context of the word i.

$$X_i = \sum_K X_{ik} \tag{1}$$

$$P_{ij} = p(j \mid i) = x_{ij}/x_j \tag{2}$$

$\boldsymbol{w}_i$, $\boldsymbol{w}_j$ and $\boldsymbol{w}_k$ represent the word vectors of words i, j, and k respectively, the meaning of the ratio of P_{ik}/P_{jk} is related to $\boldsymbol{w}_i$, $\boldsymbol{w}_j$ and $\boldsymbol{w}_k$, so an unknown function F is used to fit this relationship.

$$F(w_i, w_j, w_k) = P_{ik}/P_{ij} \tag{3}$$

$\boldsymbol{w} \in R^d$ represents the word vector, and the $\tilde{\boldsymbol{w}} \in R^d$ shows the context word vector. Following a number of derivations, the following cost function J is derived.

$$J = \sum_{ij} f(X_{ij})(w_i^{\mathrm{T}} w_J + b_i + b_j - \log X_{ij})^2 \tag{4}$$

$$f(x) = \begin{cases} (x/x_{\max}) & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

Practically, if two words co-occur more often, the two words' weight will be greater, so a weight value $f(x)$ is designed to weigh each item of the cost function. This article sets $x_{\max} = 100$, $\alpha = 3/4$. Finally, the text words are represented in the form of word vectors and sent to the MHA layer for feature extraction.

2.2 Feature extraction based on attention mechanism

2.2.1 Multi-head attention feature extraction

The article uses MHA to process the word vector, enabling the model to obtain more plane features from different characterization subspaces. This enables the model to capture more contextual information about sentences. With RNN or LSTM alone, it is necessary to calculate, in turn, for the characteristics of long-distance interdependence. This takes several time steps to accumulate information to link the two, and the distance, the less likely it is to be effectively captured.

Take the output of word embedding layer $\boldsymbol{X} = [w_1, w_2, w_3, \cdots, w_n]^{\mathrm{T}} \in R^{n \times d}$ as input, n represents the number of word vectors in the document, $\boldsymbol{w}_i$ is the word vector of the i word in the text, and d is the dimension of the word vector. $\boldsymbol{X}$ is assigned with three weights W_Q, W_K and W_V after linear mapping.

$$\boldsymbol{Q} = Linear(X) = \boldsymbol{X}W_Q \tag{6}$$

$$\boldsymbol{K} = Linear(X) = \boldsymbol{X}W_K \tag{7}$$

$$\boldsymbol{V} = Linear(X) = \boldsymbol{X}W_V \tag{8}$$

The dot product is a similar function applied in the current study. First, $\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}$ is calculated, that is, to find the attention matrix, and then use the attention matrix to weight $\boldsymbol{V}$ to get the final feature matrix. $\sqrt{d_k}$ is to turn the attention matrix into a standard normal distribution, making the results of softmax normalization more stable so that reverse propagation obtains a balanced gradient. Each unit in the attention matrix represents the attention weight value calculated by using the i-th word and the j-th word, which can describe how much attention the i-th word gives to the j-th word.

$$Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax\ (\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}/\sqrt{d_k})\boldsymbol{V} \tag{9}$$

Perform attention calculation on $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$, a process that only calculates one unit. In fact, to calculate multiple heads, each $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ linear transformation parameters are dissimilar. Next, the m-all attention results are connected, as shown in Fig. 2.

$$head_i = Attention(\boldsymbol{Q}W_i^Q, \boldsymbol{K}W_i^K, \boldsymbol{V}W_i^V) \tag{10}$$

$$W_t = Multihead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Concat(head_1, \cdots, head_m)W^O \tag{11}$$

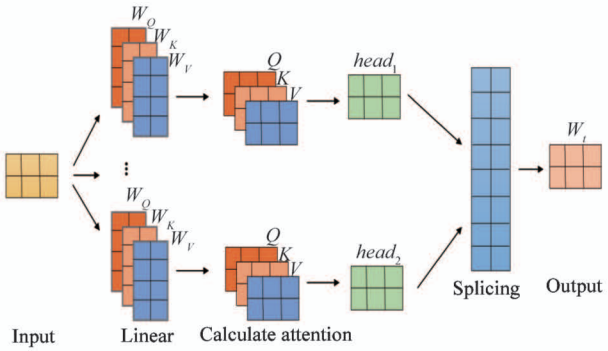

Fig. 2 Multi-head attention mechanism

2.2.2 Improved structured self-sttention feature extraction

The improved SSA structure is divided into two layers: (1) The BiLSTM layer is used to collect the timing of the context between sentences. It can combine the forward and backward hidden layers, both of which access the forward and backward context sequences. (2) Self-attention layers are used to perform each word vector in the context word vector matrix, capturing syntax or semantic features between words in the same sentence. It is also easier to capture the interdependent characteristics of long distances in a sentence. Previously, attention extraction was used to represent a sentence in a vector, but when the improved SSA feature extraction was extracted, then a two-dimensional matrix was used to characterise sentences. The attention points selected in the attention feature extraction are usually concentrated on specific sentence components, such as a collection of feature trees or related phrases. However, a sentence may have numerous components that make up the entire sentence semantics, especially in long sentences. Therefore, to represent the whole sentence semantics, it is required to focus on the various segments of the sentence using a matrix to express the different parts of the sentence.

(1) BiLSTM timing capture

The traditional unidirectional LSTM can only record forward information. However, in emotional analysis tasks, the text's emotional orientation depends on the contextual information of the forward and backward direction. Therefore, the text's above information and the following information are obtained by using the BiLSTM model. LSTM model comprises various repeated memory units, each containing 3 gates with distinctive functions: input gate i_t, forget gate f_t, and output gate o_t. The unit has a structure diagram as Fig. (3). Suppose the feature $\boldsymbol{embs} = [e_1, e_2, \cdots, e_t]^{\mathrm{T}}$ as input, t as the present time, h_{t-1} means the previous moment hidden layer state value, and c_{t-1} means the cell state value at the last moment. LSTM status value calculated

the corresponding to t-moment:

$$i_t = \sigma(W_{xi}e_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (12)$$

$$f_t = \sigma(W_{xf}e_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (13)$$

$$g_t = \tanh(W_{xc}e_t + W_{hc}h_{t-1} + W_{cc}c_{t-1} + b_c) \quad (14)$$

$$c_t = i_t g_t + f_t c_{t-1} \quad (15)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (16)$$

$$h_t = o_t \tanh(c_t) \quad (17)$$

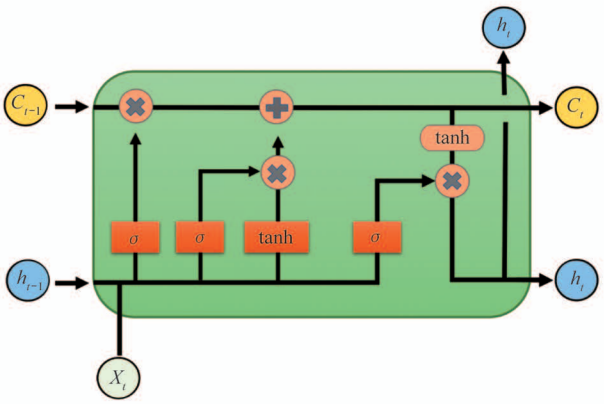

Fig. 3 LSTM structure diagram

The output value of the t-moment LSTM hidden layer state is finally obtained through the above calculation. LSTM only reflects the historical data of the sequence, it is unsufficient. BiLSTM operates as follows: the forward layer catches the historical knowledge of the series, and the backward layer catches the potential information. The prominent feature of this structure is that the background details of the chain are thoroughly regarded. Fig. 4 is BiLSTM structure.

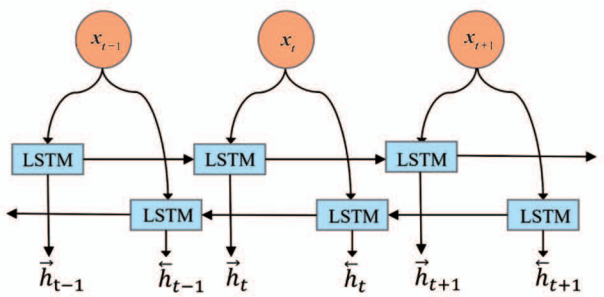

Fig. 4 BiLSTM structure diagram

Assume that the input at time t is an important feature $\boldsymbol{W}_t$ obtained by the MHA layer at $t-1$ moment, the output of the forward hidden unit is $\overrightarrow{\boldsymbol{h}}_{t-1}$ and the output of the backward hidden unit is $\overleftarrow{\boldsymbol{h}}_{t+1}$. Then the output of the hidden unit at t moment is $\overrightarrow{\boldsymbol{h}}_t \in R^{n\times u}$ and $\overleftarrow{\boldsymbol{h}}_t \in R^{n\times u}$.

$$\overrightarrow{\boldsymbol{h}}_t = L(w_t, \overrightarrow{h}_{t-1}, c_{t-1}) \quad (18)$$

$$\overleftarrow{\boldsymbol{h}}_t = L(w_t, \overleftarrow{h}_{t+1}, c_{t-1}) \quad (19)$$

(2) Self-attention feature extraction

The self-attention mechanism usually focuses on itself and extracts relevant information from it, rather than using additional information. The basic principle of the self-attention mechanism is that words in the same sentence have more or less semantic associations. By calculating the degree of association between a word and other words, the self-attention mechanism can capture some syntactic or semantic features between words in the same sentence, and make it easier to capture interdependent features of long distance in the sentence. The self-attention mechanism connects two words in a sentence during calculating, by allowing the distance between long-distance dependent features to be greatly reduced, and these features can be used effectively. Take the forward hidden state $\overrightarrow{\boldsymbol{h}}_t \in R^{n\times u}$ of LSTM as input, the implementation mechanism of self-attention is equivalent to a simple feedforward neural network, by passing through two fully connected layers, and the weight matrix $\boldsymbol{A}_1$ is the output.

$$\boldsymbol{A}_1 = softmax(\boldsymbol{W}_{s2}\tanh(\boldsymbol{W}_{s1}, \boldsymbol{h}_t^{\mathrm{T}})) \quad (20)$$

where $\boldsymbol{W}_{s1} \in R^{da\times u}$, $\overrightarrow{\boldsymbol{h}}_t^{\mathrm{T}} \in R^{u\times n}$, $\boldsymbol{W}_{s2} \in R^{1\times da}$. Softmax function is used for normalization, each dimension of $\boldsymbol{A}_1$ can be considered as the attention of the corresponding position. The shape of $\boldsymbol{A}_1$ is $R^{1\times n}$.

For sentence sentiment classification tasks, the input is usually a sentence. This sentence is marked as a matrix of word vectors, and the calculated attention weight is a one-dimensional vector, called attention weight vector. Each dimension of the attention weight vector represents the importance of weight in each word in the sentence, and the importance weight is multiplied by the corresponding word vector and then added to obtain the final attention vector. The self-attention mechanism can be understood as mapping the input from a two-dimensional matrix to a one-dimensional vector, as shown in Fig. 5.

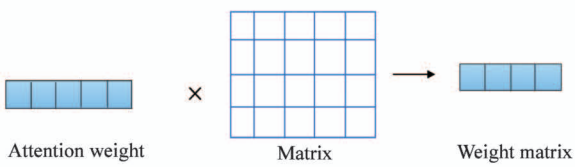

Fig. 5 One-dimensional attention matrix

$\boldsymbol{W}_{s1}$ is a weight matrix for a $d_a \times u$ -shaped, and d_a is a hyperparameter that can be determined. The chosen focus is usually on specific components of a sentence, such as a collection of feature trees or related phrases. Therefore, it should reflect the whole sentence. However, a sentence may have multiple components that make up the semantics of an entire sentence. To solve the phenomenon that a sentence represented by a one-dimensional vector will cause semantic

incompleteness, a two-dimensional matrix is proposed to represent each part of the sentence. The whole sentence is the scope, and r is the different parts extracted from the sentence and turned $\boldsymbol{W}_{s2}$ into an $r \times d_a$ matrix.

Fig. 6 is expressed as a matrix.

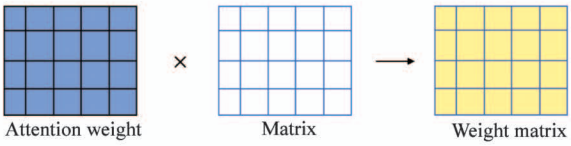

Fig. 6 Self-attention matrix

The final weight matrix is weighted for the forward hidden state $\overrightarrow{\boldsymbol{h}_t}$ and the weighted matrix $\boldsymbol{M}_1 \in R^{r \times u}$ is obtained.

$$\boldsymbol{M}_1 = \boldsymbol{A}_1 \overrightarrow{\boldsymbol{h}_t} \tag{21}$$

Similarly, taking the backward hidden state $\overleftarrow{\boldsymbol{h}_t}$ of the LSTM as an input, a weight matrix $\boldsymbol{A}_2$ is obtained, and the backward hidden state $\overleftarrow{\boldsymbol{h}_t}$ is weighted to obtain a matrix $\boldsymbol{M}_2$.

Finally, connect $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ to get the final weighted context.

$$\boldsymbol{M} = \boldsymbol{M}_1 + \boldsymbol{M}_2 \tag{22}$$

2.3 Softmax classification

This paper constructs a fully connected network to form an output network, with the output vector $\boldsymbol{M}$ of SSA as the input of this layer. The fully connected layer's function is to map the feature to vector, then multiply the vector. Finally, the dimension is decreased. Next, the vector is normalized by the softmax function to obtain the conditional probability of individual category, and the category corresponding to the maximum value of the conditional probability is selected as the predicted output category. The calculations are as follows.

$$P(\boldsymbol{y}' \mid S) = softmax(\boldsymbol{W}_o M + \boldsymbol{b}_o) \tag{23}$$

$$y = argmax\ p(\boldsymbol{y}' \mid S) \tag{24}$$

where, $\boldsymbol{W}_o \in R^{m \times r}$, $\boldsymbol{b}_o \in R^{m \times r}$ are the trainable parameters of the output layer, and represent the number of types of emotion classification $\boldsymbol{y}'$ which is a $1 \times$ m-dimensional vector (i. e., $\boldsymbol{y}' \in R^m$), and the value of each dimension represents probability that the target belongs to this emotional category. In the experiment process, such as Eq. (24), generally choose the category with the largest probability value as the emotional polarity of the target word.

In the training process, the predicted result y output by the fully connected layer is compared with the correct result to get the model's error in the training set. Therefore, the aims of model training in the current study are to curtail the classification cross entropy loss.

$$J(\theta) = -(1/m) \sum_{i-1}^{m} t_i \log(y_i') \tag{25}$$

where, y_i' is the projected probability value of individual category in the softmax classification between 0 and 1, and the sum of the probabilities of all categories is 1, m is the number of types of emotional polarity, and the value of t_i is 0 or 1, representing whether the target word belongs to a particular emotional category.

Matrix $\boldsymbol{A}_1$ is a weighted matrix that must ensure that each row's weight is different as possible, because then different weights can capture different relationships and information between words and learn the various emphasis on sentence representations. Nevertheless, if left unchecked, multiple learning lines likely have the same weight, resulting in redundancy. To make $\boldsymbol{A}_1$ as diverse as possible (because if they are all similar, there will be redundancy), introduce the following penalty after the loss function expression.

$$J(\theta) = -(1/m) \sum_{i-1}^{m} t_i \log(y_i') + \| (\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} - \boldsymbol{I} \|_F^2 \tag{26}$$

The penalty term is to make the $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ diagonal elements as close as possible to 1, that is, to make the distribution of each set of attention weights more focused on a few words, which can ensure the diversity of attention and reduce redundancy between the weight vectors.

3 Experimental analysis

3.1 Emperimental setup

This paper uses the NLPIR Weibo corpus and Twitter dataset for comparison experiments. The sample's emotional polarity is divided into positive, negative, and neutral. Among them, the NLPIR corpus is obtained by public collection and extraction from Sina Weibo. The Twitter dataset is obtained by Kaggle competition tasks. For the Chinese corpus data set, the Jieba toolkit is used for word segmentation in the experiment, and converted into word vectors using the GloVe. The same pre-trained word vectors are used to ensure the single variable principle of the comparison experiment. Stopwords and special symbols in the text are reserved as ordinary words. It is verified that the proposed MHA-SSA model has high accuracy on different language datasets through comparative experiments. Table 1 enlists the training data used in this experiment.

Table 1 Statistics of training data used in experiments

Data set	Positive	Negative	Natural
NLPIR	19 356	18 549	12 095
Twitter	20 489	18 965	10 546

3.2 Baseline methods

MHA-SSA with CNN, LSTM, BiLSTM, BiLSTM + ATT, BiLSTM + MHA are proposed. These 5 methods are tested on 2 different data sets.

(1) CNN[26]. The CNN model is one of the most classic models. The word vector is obtained through GloVe training to study the emotion classification effect of CNN.

(2) LSTM[27]. LSTM adds threshold control on the basis of RNN, which can improve the timing relationship of vectors and avoid the generation of gradient explosions. This paper uses semantic vector as features and LSTM networks as classifiers to study the sentiment classification on LSTMs.

(3) BiLSTM[28]. As an improvement of RNN, BiLSTM has better performance for processing sequence data. This paper uses the semantic vector of text as input. BiLSTM captures the contextual emotional semantic content of sentences. Research the effect of BiLSTM on sentiment classification through experiments.

(4) BiLSTM + Attention[29]. In the experiment, the word vector is input into the model as a feature, and the BiLSTM model is used to extract bidirectional temporal features. Next, the attention mechanism is used to give different weights to the information.

(5) BiLSTM + MHA[30]. The MHA mechanism is introduced to allow model to obtain more information about sentences from different representative spaces, to improve the model's feature expression ability, and the model's emotion classification effect is studied.

3.3 Evaluation standard

The evaluation metrics of the classification model involve, in particular, the recall rate, the accuracy rate, and $F1$. The validity rate is the ratio of true positive samples to those expected to be positive. There are 2 options for predicting positive, one for predicting positive class as a positive class (TP) and the other for predicting negative class as a positive class (FP).

$$P = TP/(TP + FP) \tag{27}$$

The recall rate is the rate at which positive examples in the sample are predicted appropriately. First, when the prediction is accurate, i. e. , the original positive class is anticipated to be positive (TP), and the other is that the prediction fails, i. e. , the original positive class is envisaged to be negative (FN).

$$R = TP/(TP + FN) \tag{28}$$

Occasionally the recall rate and the accuracy rate may contradict each other. Neither the recall rate nor the accuracy rate can comprehensively measure the model's performance, so the F-measure (also called F-score) method is introduced. This article uses the most common calculation method.

$$F1 = (2 \times P \times R)/(P + R) \tag{29}$$

3.4 Emotion classification

3.4.1 Parameter settings

The parameter setting of the deep learning model is critical. The main parameters and parameter values are shown in Table 2. According to the ratio of 8:2, the data is divided into training set and test set. Training set is applied to train the model, and the test set is applied to test classification performance of model. The evaluation index used is the accuracy, recall rate, and F1 score.

Table 2 Parameter value

Parameter	Parameter description	Parameter value
lr	Learning rate	0.001
batch _ size	Data batch processing volume	128
epochs	Number of iteration rounds	50
embedding _ size	Word vector length	100
hidden _ unit	Number of hidden layer units	100
dropout	Neuron dropout rate	0.5

3.4.2 Results

This experiment uses the Mxnet deep learning framework and 50 000 data to test the effects of 6 models. The experimental results obtained are listed in Table 3.

Table 3 NLPIR experimental result accuracy

Model	P/%	R/%	$F1$/%
CNN	80.23	79.74	79.98
LSTM	83.39	84.39	83.89
BiLSTM	84.33	86.39	85.35
BiLSTM + Attention	96.49	79.59	87.23
BiLSTM + MHA	96.62	80.56	87.86
MHA + SSA	96.84	81.1856	88.32

For the Chinese data set, Table 3 shows that MHA + SSA has significantly promoted accuracy value and $F1$ values compared with other models. Compared

with CNN, LSTM, and BiLSTM, the accuracy of *F*1 has been significantly improved. Contrast BiLSTM + MHA model, the performance is equivalent, and the accuracy rate is improved by 0.46%. These results indicate that hybrid model can better make up for the lack of learning ability of a single deep learning model to a certain extent.

As shown in Table 4, the first three single neural network models have relatively low accuracy of sentiment analysis. It is proved that adding attention mechanisms based on a single model and composite model can effectively improve model classification accuracy, because attention mechanism can assign different weight values to features, allowing the model to rapidly grasp important features. From the comparison results, it can be seen that the accuracy of using the MHA-SSA model is significantly higher than traditional neural network methods.

Table 4 Twitter experimental result accuracy

Model	*P*/%	*R*/%	*F*1/%
CNN	87.62	87.27	87.39
LSTM	88.14	87.35	87.74
BiLSTM	88.79	88.35	88.57
BiLSTM + Attention	90.62	92.08	91.48
BiLSTM + MHA	92.03	91.11	91.56
MHA + SSA	92.73	92.47	92.87

This classification effect on Twitter data is higher than the NLPIR data set. By analysing the data set characteristics, the average sentence length of NLPIR data is shorter than Twitter data. In addition, according to the comparison of the model classification results of Chinese and English corpora, MHA-SSA model has obviously optimized.

Fig. 7 shows the changes in *F*1 values of 6 models in 21 iterations under the NLPIR dataset. The horizontal axis represents the number of epoch, and the vertical axis represents the *F*1 value. The *F*1 value of each model on the training set gradually increases as the number of epoch increases, and eventually stabilizes. As shown in the Fig. 7, MHA-SSA model first applied the multi-head attention mechanism to assign corresponding weights, and after 21 iterations, the *F*1 value is 0.46% and 1.09% higher than the BiLSTM + MHA and BiLSTM + attention model. Compared with the BiLSTM, LSTM, and CNN model, the *F*1 value increased by 2.97%, 4.43%, and 8.34% after adding the attention layer to obtain key information. Therefore, MHA-SSA model can not only capture the global connection, but also model the local information, and well to improve accuracy of emotion classification.

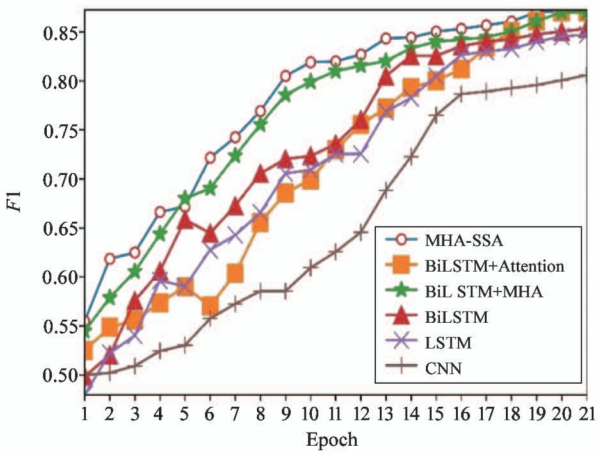

Fig. 7 Changes in *F*1 values during the iteration of different classification models under the NLPIR dataset

4 Conclusion

A new MHA-SSA model for emotion classification is proposed. The original text is preprocessed as a word vector. The MHA mechanism can extract deeper grammatical features from the word vector matrix, and the structured attention mechanism can extract multiple features from the local area. A two-dimensional matrix is used to represent a weighted sentence. From the experimental results, the accuracy of the MHA-SSA model is better than others.

During the research process, it is discovered that the experiment still needs improvement. When using the word embedding model to learn a sentence, two words with same context but completely opposite meanings have very close spatial distances. Therefore, the preprocessing in the data and the training methods of word vectors also have an important influence on sentiment classification, and are also the direction of further research on sentiment classification in the future.

References

[1] Kenichi F, Yuichi N. Approximation of dynamical systems by continuous time recurrent neural networks[J]. *Neural Networks*, 1993, 6(6):801-806

[2] Hochreiter S, Schmidhuber J. Long short-term memory [J]. *Neural Computation*, 1997, 9(8):1735-1780

[3] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures[J]. *Neural Networks*, 2005, 18(5):602-610

[4] Liu W, Liu P, Yang Y, et al. An attention-based syntax-tree and tree-LSTMn model for sentence summarization [J]. *International Journal of Performability Engineering*, 2017, 13(5):775-782

[5] Nowak J, Taspinar A, Scherer R. LSTM recurrent neural networks for short text and sentiment classification[C]// Proceedings of the 16th International Conference on Artificial Intelligence and Soft Computing, Zakopane, Po-

land, 2017: 553-562
[6] Chen T, Xu R, He Y, et al. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN [J]. *Expert Systems with Applications*, 2017, 72: 221-230
[7] Niu X, Hou Y, Wang P. Bi-directional LSTM with quantum attention mechanism for sentence modelling [C] // Proceedings of the 24th International Conference on Neural Information Processing, Guangzhou, China, 2017: 178-188
[8] Tian D P. Semi-supervised learning based probabilistic latent semantic analysis for automatic image annotation [J]. *High Technology Letters*, 2017, 23(4): 367-374
[9] Li X, Feng S, Wang D, et al. Context-aware emotion cause analysis with multi-attention-based neural network [J]. *Knowledge-Based Systems*, 2019, 174: 205-218
[10] Stojanovski D, Strezoski G, Madjarov G, et al. Twitter sentiment analysis using deep convolutional neural network [J]. *In International Conference on Hybrid Artificial Intelligence Systems*, 2015, 9121: 726-737
[11] Xu G, Meng Y, Qiu X, et al. Sentiment analysis of comment texts based on BiLSTM [J]. *IEEE Access*, 2019, 7: 51522-51532
[12] Usama M, Ahmad B, Yang J, et al. REMOVED: equipping recurrent neural network with CNN-style attention mechanisms for sentiment analysis of network reviews [J]. *Computer Communications*, 2019, 148:98-98
[13] Li X, Feng S, Wang D, et al. Context-aware emotion cause analysis with multi-attention-based neural network [J]. *Knowledge-Based Systems*, 2019, 174:205-218
[14] Bahdanau D, Cho K H, Bengio Y. Neural machine translation by jointly learning to align and translate [J]. *arXiv*:1409.0473*v*6, 2015
[15] Zheng H, Wang W, Chen W, et al. Automatic generation of news comments based on gated attention neural networks [J]. *IEEE Access*, 2018,6: 702-710
[16] Heo Y, Kang S, Yoo D. Multimodal neural machine translation with weakly labeled images [J]. *IEEE Access*, 2019, 7: 54042-54053
[17] Liu S, Zhang S, Zhang X, et al. R-trans: RNN transformer network for Chinese machine reading comprehension [J]. *IEEE Access*, 2019, 7: 27736-27745
[18] Wang L, Cao Z, De Melo G, et al. Relation classification via multi-level attention CNNs [C] // Proceedings of the 54th Annual Meeting of the Association for Computational Lingistics, Berlin, Germany, 2016:1298-1307
[19] Wang Y, Huang M, Zhao L, et al. Attention-based LSTM for aspect-level sentiment classification [C] // Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, USA, 2016:606-615
[20] VaswaniA, Shazeer N, Parmar N, et al. Attention is all you need [C] // In Advances in Neural Information Processing Systems, California, USA, 2017:6000-6010
[21] Lin Z H, Feng M W, Santos C N, et al. A structured self-attentive sentence embedding [J]. *arXiv*: 1703.03130*v*1, 2017
[22] Ankur P, Oscar T, Dipanjan D, et al. A decomposable model for natural language inference [C] // In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, USA, 2016:2249-2255
[23] Liu G, Guo J. Bidirectional LSTM with attention mechanism and convolutional layer for text classification [J]. *Neurocomputing*, 2019, 337:325-338
[24] Long F, Zhou K, Ou W. Sentiment analysis of text based on bidirectional LSTM with multi-head attention [J]. *IEEE Access*, 2019, 7: 141960-141969
[25] Jeffrey P, Richard S, Christopher D. GloVe: Global vectors for word representation [C] // In Empirical Methods in Natural Language Processing, Doha, Qatar, 2014: 1532-1543
[26] Yoon K. Convolutional neural networks for sentence classification [C] // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 2014: 1746-1751
[27] Kai Sheng T, Richard S, Christopher D. Improved semantic representations from tree-structured long short-term memory networks [C] // Association of Computational Linguistics, Beijing, China, 2015:1556-1566
[28] Xiao Z, Liang P. Chinese sentiment analysis using bidirectional LSTM with word embedding [C] // Proceedings of the 2nd International Conference on Cloud Computer and Security, Berlin, Germany, 2016:601-610
[29] Zhou P, Shi W, Tian J, et al. Attention-based bidirectional long short-term memory networks for relation classification [C] // Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 2016:207-212
[30] Liu F, Gao S, Yu B, et al. Relation classification based on multi-head attention and bidirectional long short-term memory networks [J]. *Computer Systems and Applications*, 2019, 28(6): 118-124

Li Ying, born in 1997. She is a postgraduate student of University of Shanghai for Science and Technology. She received a B. S. degree from Sanya University in 2018. Her research interests include natural language processing and emotion classification.