Dual channel sentiment classification model based on grammar rules and multi attention

Xianjin Yi*, Chen Li
School of Software, Beijing University of Technology, Beijing, China

*Corresponding author: yixianjin@emails.bjut.edu.cn

Abstract. As convolutional neural networks (CNN) and recurrent neural networks (RNN) have achieved excellent results in the field of Chinese text sentiment analysis. More and more researchers are extracting features of text based on the advantages of CNN and RNN in extracting features. However, the current scholars fail to make full use of sentiment language resources such as sentiment words, negatives and degree adverbs when they adopt deep learning methods. For implicit texts without explicit sentiment words, they cannot fully identify the differences between words and sentiment tendency. At the same time, it fails to consider the grammatical structure of the text, which leads to poor classification effect for some turning sentences or summary sentences. In addition, most of the models are mainly input in the form of word vector. For English, it is very convenient to segment words through spaces between words, but for Chinese, there may be inaccurate word segmentation, which will reduce the accuracy of classification. To solve these problems, a dual channel sentiment classification model based on grammar rules and multi attention (DCGA) is proposed. Firstly, the text with clear sentiment tendency is obtained according to the grammar rules, and the local features of the text are extracted by CNN channel. Considering that the grammar rules may ignore the context information, bi-directional long short term memory network (Bi-LSTM) channel is used to extract the global features containing the context information, and attention pooling is used to improve the sentiment information extracted by CNN channel. Then, the part of speech attention mechanism is used to mine the implicit sentiment features in the sentiment text to solve the poor classification effect caused by the existence of implicit text. Finally, the local features obtained by CNN channel and the global features obtained by RNN channel are fused for classification. Experimental results on four Chinese comment text datasets show that the proposed model outperforms most existing methods in accuracy.

Keywords: Sentiment analysis, neural network, grammatical rule.

1. Introduction
With the development of Web2.0 technology, the network has not only become an important source of information, but also a platform for people to express their views. Online comments have become the main way for the public to participate in social hot topics. Through online comments, people can comment on hot events, express opinions of film reviews, and describe product experience on the
Internet. Thus, a large number of text information with sentiment tendencies are generated. Through the sentiment analysis of these text information, we can better understand the user's behavior, and find the user's tendency to the product and the degree of attention to hot events [1]. For the government, analyzing the comment data of BBS and other network platforms to understand the sentiment tendency of the masses towards social events can help the government guide the development of network public opinion in a healthy direction and promote the development of network and social environment; for businesses, analyzing customers' consumption comments and understanding customers' preference for goods can help businesses improve product quality and service level.

Sentiment analysis, also known as opinion mining, refers to people's sentiments, opinions, evaluations, attitudes and sentiments about services, products, organizations, individuals, problems, events, topics and their attributes [2]. Sentiment classification is a task in sentiment analysis. According to the meaning and sentiment information expressed in the text, the text is divided into praise, derogatory, neutral and other sentiment categories. It is the division of the author's inclination and views, so it is also called tendency analysis. With the rapid increase of information scale, only relying on manual processing has been unable to complete this task, which promotes the development of text sentiment analysis technology.

Sentiment analysis methods mainly include machine learning based method and sentiment dictionary-based method. With the development of neural network research, deep learning-based method is widely used in text sentiment analysis task [4]. The method based on sentiment dictionary is first used in the field of sentiment analysis. It first needs to build a sentiment dictionary, then manually label the polarity and strength of the sentiment dictionary, and finally realize the sentiment classification of the text. Although this method can classify the sentiment of the text, the efficiency of classification is not high, because it needs to manually construct the sentiment dictionary and label it manually. Based on the traditional machine learning algorithm, structured text information features are constructed by artificial design features, and then analyzed by machine learning method. Common text sentiment analysis methods include naïve Bayes, support vector machine, maximum entropy method, etc. Although the function of machine learning model is simple, it usually needs complex feature engineering, and the quality of features can directly affect the classification effect. In addition, the generalization ability of machine learning method is relatively low. In recent years, the emerging deep learning method makes up for the defects of machine learning and sentiment dictionary-based methods, and has been successfully applied to sentiment analysis tasks, such as convolutional neural network (CNN) [4,5] and long short-term memory (LSTM) network [6,7].

As mentioned above, these methods provide different solutions to affective analysis problems, which are widely used in various affective analysis tasks, and achieve good performance evaluation results. However, most of the methods based on sentiment dictionary and traditional machine learning have the limitation of relying too much on language knowledge or difficult to extract text features comprehensively. In addition, the method of deep learning often uses the form of pre trained word vector to express the text. For Chinese, it fails to make full use of sentiment language resources, fails to consider the grammatical structure of Chinese text, and ignores the situation of word segmentation errors. Therefore, the task of sentiment analysis is still facing great challenges.

In order to solve the above problems, we consider a dual channel neural network model which combines grammar rules and multi attention. Its goal is to make better use of grammar rules and sentiment resource information to extract text features, reduce the impact of word segmentation errors, improve the learning and generalization ability of the network, and improve the accuracy of sentiment analysis. In this paper, a DCGA model is proposed. Firstly, grammar rules are used to obtain the text with clear sentiment tendency. Local features are extracted by (CNN) channel, and global features including context information are extracted by (BILSTM) channel. Attention pooling is used to improve the sentiment information extracted by CNN channel. Then the part of speech attention mechanism is used to focus on the important sentiment features in the text. Finally, the features extracted from the two channels are fused for classification. Experimental results show that the proposed model outperforms most existing methods in accuracy.
2. Related work

Sentiment classification has always been a hot topic in the field of natural language processing. In the early years, it mainly used the method based on sentiment dictionary and machine learning. Although using sentiment dictionary for classification is easy to operate, the construction of sentiment dictionary is time-consuming and labor-consuming, and the scope of expansion is limited. Machine learning based sentiment classification is usually regarded as a supervised learning problem. Pang et al. [8] used maximum entropy, naive Bayes and support vector machine (SVM) to carry out experimental comparison in text sentiment analysis, and found that using SVM for text sentiment classification can achieve the best effect. Lee et al. [9] applied the maximum entropy classification to the polarity evaluation of a given electronic product review. Using machine learning method requires complex feature selection. This process often needs to be designed manually, which will lead to poor scalability, and it is difficult to adapt to different fields and application requirements.

In recent years, deep learning method has gradually become the mainstream method of sentiment analysis. It does not need manual intervention. It is an end-to-end method, which can realize the automatic selection of features. Therefore, the efficiency will be higher than the method based on dictionary and machine learning, and the domain applicability will be stronger. Kim [4] proposed to apply CNN to English text sentiment classification task, and achieved good classification results at that time. Kalchbrenner et al. [5] proposed a wide convolution model, and used k-max pooling instead of maximum pooling to retain more feature information. Yin et al. [10] used multi-channel convolutional neural networks with different sizes to classify sentences, and achieved good results in Chinese microblog sentiment analysis task. However, CNN based text sentiment classification cannot consider the semantic information of sentence context.

Compared with CNN, RNN introduces memory unit, which makes the network have a certain memory and can capture the long-distance dependence between texts. However, the traditional RNN has the problem of gradient vanishing. The LSTM and the gated recurrent unit (GRU) network introduce the gate mechanism on the basis of the traditional RNN, which overcomes the disadvantage of gradient disappearance in RNN. Based on LSTM and GRU models, Tang et al. [11] first used CNN or LSTM to realize single sentence representation, then used gatedRNN to encode the internal relations and semantic relations between sentences, and finally realized text coding at text level. This method can better capture the semantic information between sentences. Zhou et al. [12] proposed a C-LSTM model. Firstly, CNN was used to extract text features, and then LSTM network was used to replace the maximum pooling layer to get the final classification results. Sachin et al. [13] used two-way GRU and two-way LSTM methods to analyze Amazon reviews, and achieved good results. Although LSTM and GRU models can extract contextual semantic information and take into account the timing problem between words, they cannot extract the local features of sentences as well as CNN.

In order to give full play to the respective advantages of CNN and RNN, more and more researchers combine CNN and RNN to analyze text sentiment. Wang et al. [14] used single-layer CNN and RNN to form a fusion model for short text sentiment analysis, and the experimental effect is better than using CNN and RNN alone. Zhang et al. [15] proposed a multi-channel CNN-LSTM model for twitter text sentiment classification task. Alayba et al. [16] proposed a method combining CNN and LSTM model for Arabic sentiment analysis, and achieved good classification results.

In recent years, attention mechanism has been widely used in text classification tasks, because attention mechanism can make the model selectively select important information and ignore some unimportant information, which can make the model pay attention to the words that are important for sentiment classification in sentences. Yang et al. [17] combined bidirectional RNN and attention mechanism, constructed attention model and applied it to text classification task at text level, and achieved the best results of text classification task at that time. Wang et al. [18] combined multi-layer attention mechanism with CNN in sentence relation classification task. The experimental results on multiple data sets show that the model using attention mechanism has higher classification accuracy than the model not used. In view of the excellent performance of neural network model integrating attention mechanism, this paper also introduces attention mechanism in the task of text sentiment analysis.
orientation analysis, so that the network model can pay more attention to the words that contribute greatly to the text sentiment polarity.

Based on the respective advantages of appeal method, this paper proposes a multi-channel CNN and BiLSTM network model based on attention mechanism. The model uses CNN channel to extract the local feature information of text word granularity filtered by grammar rules, and uses BiLSTM channel to extract the semantic information of text context. Then attention mechanism is introduced into each channel, and finally they are spelled Then, a new dual channel neural network model based on attention mechanism is obtained. In this paper, the advantages of CNN, BiLSTM network and attention mechanism are comprehensively used, which not only considers the local information of the text, but also takes into account the context semantic information, so as to avoid the influence of text segmentation errors, but also pays attention to the words in sentences that contribute a lot to sentiment classification through attention mechanism, which improves the performance of the model for text sentiment classification.

3. Our model

Sentiment analysis can be divided into two categories and multiple categories. For two categories, each text is divided into positive or negative categories. For multi-category, in addition to positive and negative, each text can also be divided into other categories. In this article, we use the binary classification method to predict whether a text is positive or negative.

Our proposed model architecture is shown in Figure 1. It consists of a text processing layer, an embedding layer, a feature extraction layer, a collection of part-of-speech attention layer and attention pooling layer, and a fully connected layer and softmax layer on the output side. Further details of each layer will be described in the following subsections.

![Figure 1. The structure of DCGA.](image)
3.1. Text processing layer

3.1.1. Text preprocessing. Due to the free form of comment text, it contains not only the text with semantic information, but also a large number of noise data. In order to reduce the impact of noise data on text sentiment analysis, we need to preprocess the comment text as follows: 1) filter out all punctuation and special characters, and only retain the Chinese text with semantic value information. 2) Use Jieba word segmentation tool for word segmentation. 3) The intersection of Harbin Institute of technology stop thesaurus and Baidu stop thesaurus is used to remove noise data.

3.1.2. Grammar rules. Through the study of Chinese text grammar rules, it is found that there are a large number of complex sentences in the text, such as summary sentences, connective sentences and turning sentences. The summary sentence is the key point of the author's opinion, which directly affects the sentiment tendency of the sentence, while the turning sentence realizes the effect of sentiment reversal before and after. The turning words are divided into two types. The first type of turning words has obvious sentiment tendency in the sentence, and the second type of turning words play a transitional role. The content of the sentence cannot express the sentiment tendency of the text, and its sentiment tendency is often negative in the rest of the sentence. In order to give full play to the role of summary words and turning words in the extraction of text sentiment tendency information, reduce the semantic complexity of the text, so as to obtain more explicit text information of sentiment tendency. In this paper, according to the data set of these three types of words, we will design a dictionary: summary word dictionary and transition word dictionary. Then, according to the dictionary, the text grammar rules are set to obtain more explicit information of sentiment tendency, so as to facilitate the acquisition of CNN sentiment tendency features in training. The rules are set as follows. It is assumed that it represents a comment text, where it represents a sentence in the text and represents the text processed by the rules.

   Rule 1: match the comment text w with the summary word dictionary. If there is a summary word, the clause wi after the summary word is extracted directly. If there are multiple summary words in the text, in order to improve the classification efficiency, only the clause wi after the first summary word is extracted by default.

   Rule 2: if there is no summary sentence in the comment, search whether there are turning words in the text and judge their categories. Class a lead the positive sentence and extracts the content of the positive sentence. If class B, it ignores the partial sentence content of the B-type turning words and extracts the rest of the comments.

   Rule 3: if there are neither summary words nor turning words in the text, the original content is directly retained, and then the character vector features are extracted by convolution neural network.

3.2. Embedding layer

The sentences processed by grammar rules often express the important sentiment tendency of commentators. Considering that there are errors in word segmentation, which will lead to the loss of sentiment information and the introduction of noise, we use the form of single word to express the sentence information. Considering that the context may be ignored in the rule processing and the part of speech resources such as sentiment words, negative words and degree adverbs are not fully used in the form of single word, which is very important information in implicit text. Therefore, in the form of single word, only preprocessed text is expressed in the form of words. Therefore, in the word embedding layer, this paper mainly considers the word and word granularity of text representation. In our paper, we use a 300-dimensional pre trained word2vec for vectorization.

3.3. Feature extraction layer

3.3.1. Bi-LSTM channel. RNN can mine temporal information and contextual semantic information of text, but it has the problems of long-term dependence and gradient disappearance. In 1997, Hochreiter proposed LSTM network in order to overcome the shortcomings of gradient vanishing and gradient
exploding in RNN model [14]. LSTM model makes up for the deficiency of short-term memory by using three gates to regulate cell state. The "gate" structure is used to add and remove information. In the process of training, it learns to save important information and forget irrelevant information. The network structure of LSTM is shown in Figure 2.

LSTM has three gates: forget gate ($f_t$), input gate ($i_t$) and output gate ($o_t$). The gate structure contains sigmoid activation function. Sigmoid compresses values between 0 and 1, which helps update or forget information. When the information needs to be removed, sigmoid compresses the value to 0; the more important the information is, the closer the sigmoid is to 1. 1) Forget gate is used to decide which irrelevant information should be discarded or which important information should be retained. 2) The input gate is used to update the cell state. 3) The output gate is the value that determines the next hidden state, in which the previously input information is included. The LSTM unit calculates the new context memory information ($c_t$) by combining the input ($x_t$) of this time and the output of the hidden layer ($h_{t-1}$) of the previous time step. The calculation formula is as follows (1) - (6):

$$f_t = \text{sigmoid}(W_f(x_t, h_{t-1}) + b_f)$$  \hspace{1cm} (1)
$$i_t = \text{sigmoid}(W_i(x_t, h_{t-1}) + b_i)$$  \hspace{1cm} (2)
$$o_t = \text{sigmoid}(W_o(x_t, h_{t-1}) + b_o)$$  \hspace{1cm} (3)
$$g_t = \tanh(W_g(x_t, h_{t-1}) + b_g)$$  \hspace{1cm} (4)
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$  \hspace{1cm} (5)
$$h_t = o_t \odot \tanh(c_t)$$  \hspace{1cm} (6)

Among them, $W_f, W_i, W_o, b_f, b_i, b_o$ is the weight matrix and offset of forgetting gate, input gate and output gate respectively. The symbol $\odot$ represents element-wise vectors multiplication.

The BiLSTM layer is composed of two LSTM layers with opposite directions, which model the text sequence from two directions. For sentence sequence $[W_1, W_2, ..., W_n]$, forward LSTM processes from $W_1$ to $W_n$, and then LSTM processes from $W_n$ to $W_1$. The hidden state representations of forward LSTM and backward LSTM are spliced to obtain the context sensitive feature representation of each word. By
connecting the hidden layer output of each model, BiLSTM model uses the positive and negative order of sequence information. For the word WT, its forward implicit state representation $\vec{h}_t$, backward implicit state representation $\vec{h}_t$ and final representation $\vec{h}_t$ are shown as follows (7)-(9).

$$\vec{h}_t = LSTM\left(w_t, \vec{h}_{t-1}\right)$$ (7)
$$\vec{h}_t = LSTM\left(w_t, \vec{h}_{t+1}\right)$$ (8)
$$\vec{h}_t = [\vec{h}_t; \vec{h}_t]$$ (9)

3.3.2. CNN channel. In this channel, we use CNN to extract character features. we use a text CNN model similar to that proposed by Kim et al [4]. Here we only use a window size convolution kernel.

3.4. Attention layer

3.4.1. Part of speech attention. Attention is a mechanism to capture important features by automatically learning the weights of sequences. Considering different sentiment polarity will lead to the difference of attention weight associated with each tendency. This difference can be used as an important feature of sentiment recognition and polarity classification. We use a part of speech enhanced attention mechanism to learn and embed differences in representations. For each sentiment polarity, we introduce an independent query $q_j$, which is used to capture the sentiment characteristics of words in a specific sentiment tendency scene. The calculation formula is as follows (10) - (13):

$$e_{ij} = q_jW_h_i$$ (10)
$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{kj})}$$ (11)
$$v_j = \sum_{i=1}^{n} \alpha_{ij} h_i$$ (12)
$$V = [v_1, v_2, \cdots, v_n]$$ (13)

Among them, $v_j$ is the weighted context embedding in sentiment polarization j scene. Finally, we connect the weighted embedding of sentiment polarity as the final representation of sentence semantics. We believe that the pre trained sentiment words already contain subjective information, which will make attention learning more accurate. We fuse the embedded items of words with the same sentiment polarity and initialize the attention query vector. The calculation formula is as follows (14):

$$q_j = \frac{1}{N_j} \sum_{i=1}^{N_j} w_{ij}$$ (14)

$w_{ij}$ is the word embedding of the word i, and the sentiment polarity level of the word is j. We will be $q_j$ is the initialization of the ith attention query vector, and $N_j$ is the number of sentiment polarity classes.

3.4.2. Attention pooling. After convolution, common pooling operations include maximum pooling and average pooling. Maximum pooling is achieved by selecting a maximum value of features in the pooling window. Average pooling is to get an average value of the features in the pooling window, which loses a certain amount of strength information. This paper uses an attention pooling method, which measures
the importance of local convolution features by calculating the influence weight of the local features extracted by CNN channel on the whole sentence. Compared with the simple maximum pooling method, it retains more information. Compared with the average pooling strategy, the feature strength information is retained. The calculation formula is as follows (15) - (17):

\[ c_{sm_i} = csm(V, B) \]  \hspace{1cm} (15)

\[ \alpha_i = \frac{\exp(c_{sm_i})}{\sum_{i=1}^{n} \exp(c_{sm_i})} \]  \hspace{1cm} (16)

\[ C = \sum_{i=1}^{n} \alpha_i B_i \]  \hspace{1cm} (17)

3.5. Fusion layer
In the above calculation process, we get the part of speech enhanced semantic information V through the bilstm channel, and the local information C including the feature strength through the CNN channel. In this layer, we use the direct splicing strategy to fuse the two features, and the calculation formula is as follows (18).

\[ F = V \oplus C \]  \hspace{1cm} (18)

3.6. Classification layer
In the fully connected layer, the model maps the polarity-related fusion distributed feature representation to the instance label space. It acts as a classifier in the entire neural network model. In the output layer, the softmax function is used for normalization, and the output of the fully connected layer is converted into the approximate probability value \( \hat{y} \) of each sentiment polarity category. The calculation formula is as follows (19).

\[ \hat{y} = softmax(AF^T + b) \]  \hspace{1cm} (19)

A is the parameter matrix of the connection layer, F is the characterization of the fusion distributed characteristics, b is the bias, and softmax is a normalization function.

4. Experiments and results

4.1. Datasets
In order to verify the effectiveness and generalization of the proposed model on different datasets, we select the following three datasets and a dataset that we automatically grab by octopus. All data are divided into training/validation/test set according to the ratio of 8:1:1. The statistical information of the data set is shown in Table 1.

1. ChnSentiCorp: it is a corpus of hotel reviews collected by Dr. Tan Songbo of the Chinese Academy of Sciences, with a scale of 10000 articles. The data set is composed of four sub data sets of different sizes. In this paper, we choose the ChnSentiCorp-Htl-ba-6000(CSC) data as the Balanced Corpus, with 3000 positive and negative categories.

2. online_shopping_10_Cats (OS): it is an open-source e-commerce shopping review. There are ten categories of review objects, including books, tablets, mobile phones, fruits, etc. The comment sentiment tag is divided into two categories [0,1]. The positive comment sentiment tag is 1, and the negative comment sentiment tag is 0. There are 31351 positive comments and 31421 negative comments.

3. Smp2019: it is a Chinese implicit sentiment analysis task evaluation data set in China social media processing academic conference. In the training data set, there are 3828 and 3957 positive and negative implicit affective sentences, and 6989 negative implicit affective sentences. The results show that there are 1232 and 1358 positive and negative implicit affective sentences, and 2553 negative affective
sentences. In the test data set, 919 and 979 positive and negative implicit affective sentences and 1902 negative implicit affective sentences. In order to keep the same number of categories with other data sets, we remove the neutral data without sentiment and only keep the data set with sentiment for experiment.

4. **BK**: it’s from the binke website ([https://www.booking.com](https://www.booking.com)) Among the 6000 comments with polarity, 3000 are positive sentiment samples and 3000 are negative sentiment samples.

| Table 1. Characteristics of the datasets after preprocessing |
|-------------------------------------------------------------|
| Dataset | No. of classes | Average length | Train set size | Validation set size | Test set size | Dataset size |
|--------|---------------|----------------|---------------|---------------------|--------------|--------------|
| CSC    | 2             | 74             | 4800          | 600                 | 600          | 6000         |
| OS     | 2             | 56             | 49818         | 6227                | 6227         | 62272        |
| SMP2019| 2             | 35             | 7785          | 2590                | 1898         | 12273        |
| BK     | 2             | 145            | 4800          | 600                 | 600          | 6000         |

4.2. **Parameter settings**

The model is implemented with Pytorch. The word vector and character vector are initialized as 300 dimensional word2vec vectors pretrained in Wikipedia corpus, and the words out of vocabulary are initialized as all zero vectors. We get the best parameters through repeated training. All weighting matrices are sampled from a normal distribution with a mean value of 0 and a standard deviation of 0.1. All bias terms are initialized to vectors of all 0.1. In the convolution layer, the size of the sliding window is 3, the number of filters is 32, and the activation function of the convolution layer is relu. The mini batch size is set to 16. The number of neurons in BiLSTM layer was set to 128 dimensions. The L2 regularization coefficient $\lambda$ is set to 0.001. Using Adam method to optimize, and set the learning rate to 0.001. In order to avoid over fitting, the dropout rate is set to 0.5. The initial value of epoch is set to 50. When the validation loss lasts for 10 periods without improvement, the iteration stops.

4.3. **Comparison of classification accuracy**

The overall performance of our proposed model relies on the elements of the confusion matrix. This evaluation matrix contains four terms, namely, True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). Therefore, we can calculate the accuracy, precision, recall and F-measure as the evaluation criteria. The calculation formula is as follows (20) - (23).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$  \hspace{1cm} (20)

$$\text{Precision} = \frac{TP}{TP + FP}$$ \hspace{1cm} (21)

$$\text{Recall} = \frac{TP}{TP + FN}$$ \hspace{1cm} (22)

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$ \hspace{1cm} (23)

In order to verify the effectiveness of our proposed model, We compare our proposed method with the following method: SVM [8]: training classification by SVM, CNN [4]: the convolutional neural network is used to extract features, and the softmax layer is used to analyze the sentiment of the text, LSTM [11]: feature extraction is carried out by using recurrent neural network, and finally sentiment analysis is carried out on the text by softmax layer, Att-CNN [18]: the convolutional neural network and attention model are used for feature extraction, and the softmax layer is used for sentiment analysis of the text, LSTM-CNN [14]: the serial feature fusion model, first uses LSTM to extract forward text features, then uses CNN to extract local semantic features, and finally uses softmax layer to analyze the
text sentiment, BiLSTM-CNN[15]: parallel feature fusion model. Firstly, the text context features and local semantic features are extracted by BiLSTM and CNN respectively, and then the extracted context features and local semantic features are fused. Finally, the text is analyzed by softmax layer, CLA [19]: a hybrid hierarchical method is used to extract text features using convolutional neural network, extract context information using long-term and short-term memory network structure, and add attention mechanism into the network. The hidden features such as sentence semantics and structure are extracted from the word level and sentence level hierarchies respectively, and the features with high contribution rate of sentiment are focused through attention mechanism. The experimental results are shown in Table 2-5.

Table 2. Result comparisons of CSC dataset with other methods

| Model       | Precision (%) | Recall (%) | F1 (%) |
|-------------|---------------|------------|--------|
| Single model|               |            |        |
| SVM         | 81.25         | 81.19      | 81.21  |
| CNN         | 85.16         | 84.96      | 85.01  |
| LSTM        | 86.78         | 86.34      | 86.56  |
| Att-CNN     | 87.27         | 87.13      | 87.19  |
| Hybrid model|               |            |        |
| LSTM-CNN    | 90.14         | 91.42      | 90.78  |
| BiLSTM-CNN  | 91.06         | 92.52      | 91.88  |
| CLA         | 91.36         | 93.58      | 92.64  |
| Proposed    | 95.24         | 94.83      | 95.03  |

Table 3. Result comparisons of OS dataset with other methods

| Model       | Precision (%) | Recall (%) | F1 (%) |
|-------------|---------------|------------|--------|
| Single model|               |            |        |
| SVM         | 83.26         | 82.33      | 82.80  |
| CNN         | 89.32         | 88.14      | 88.73  |
| LSTM        | 89.78         | 89.35      | 89.57  |
| Att-CNN     | 91.25         | 92.37      | 91.81  |
| Hybrid model|               |            |        |
| LSTM-CNN    | 91.61         | 92.75      | 92.36  |
| BiLSTM-CNN  | 92.85         | 92.52      | 92.69  |
| CLA         | 92.91         | 93.21      | 93.06  |
| Proposed    | 94.38         | 95.82      | 95.10  |

Table 4. Result comparisons of SMP2019 dataset with other methods

| Model       | Precision (%) | Recall (%) | F1 (%) |
|-------------|---------------|------------|--------|
| Single model|               |            |        |
| SVM         | 57.42         | 58.75      | 58.06  |
| CNN         | 65.34         | 65.71      | 65.53  |
| LSTM        | 70.58         | 71.34      | 70.96  |
| Att-CNN     | 66.31         | 67.25      | 66.78  |
| Hybrid model|               |            |        |
| LSTM-CNN    | 72.18         | 72.24      | 72.21  |
| BiLSTM-CNN  | 73.51         | 74.16      | 73.83  |
| CLA         | 77.52         | 78.21      | 77.86  |
| Proposed    | 84.24         | 83.33      | 83.79  |
Table 5. Result comparisons of BK dataset with other methods

| Model          | Precision (%) | Recall (%) | F1 (%)  |
|----------------|---------------|------------|---------|
| Single model   |               |            |         |
| SVM            | 76.27         | 76.51      | 76.39   |
| CNN            | 78.29         | 79.83      | 79.06   |
| LSTM           | 81.31         | 83.61      | 82.64   |
| Att-CNN        | 82.81         | 82.74      | 82.78   |
| Hybrid model   |               |            |         |
| LSTM-CNN       | 85.19         | 85.33      | 85.26   |
| BiLSTM-CNN     | 87.74         | 87.52      | 87.63   |
| CLA            | 91.31         | 92.73      | 92.02   |
| Proposed       | 90.18         | 91.41      | 90.80   |

4.4. Results and discussion

According to the results in Table 2 and Table 3, it is clear that the proposed model achieves better performance than the benchmark model on two similar short comment datasets. Using deep learning for classification is far better than using traditional machine learning methods. Compared with CNN, LSTM is better at capturing temporal features, while Att-CNN has better performance, which is related to the attention mechanism, which can further capture important information of text and reduce the interference of redundant features. The first mock exam is based on the CNN and RNN model. The reason is that the single CNN model only considers the influence of local semantic features on text sentiment analysis model, and does not consider the contextual relationship of the text. The single LSTM model only considers the following information of the text, and does not take into account the influence of the text information and the local semantic features on text sentiment analysis. CNN model with attention mechanism fully considers the influence of important features on text sentiment analysis, but ignores the importance of context information in text sentiment analysis. The hybrid model considers the influence of context and local semantic features on text sentiment analysis, so it achieves good results.

The parallel structure of CNN and RNN is better than the serial structure of LSTM-CNN. Compared with the serial model of LSTM-CNN, the F1 values of BiLSTM-CNN and CLA model are respectively 1.1% and 1.86% higher on CSC data set, and 0.33% and 0.7% higher on OS data set. However, the effect is not obvious. The reason may be related to the length of the text in the data set. Because there are a large number of short comments in OS data set, some noise features will be introduced when using parallel structure for feature extraction, while the average length of CSC data set is long, and some feature information will be lost when using serial structure for feature extraction. The method in this paper has achieved very good results on these two datasets, which is due to the use of grammar rules, filtering out some sentences that cannot express the user's central sentiment, and using the part of speech attention mechanism to give higher weight to important features. For this kind of short text, there are more abbreviations and some words that do not conform to the normal expression, this paper proposes some suggestions the proposed model effectively alleviates this problem through a word granularity channel, so the proposed method achieves high accuracy on both datasets.

According to the results in Table 4, it can be clearly found that the model proposed in this paper has achieved remarkable results on the implicit comment data set. Compared with the first mock exam LSTM model, the proposed model is 12.83% higher than the F1 value. Compared with the CLA model with the best performance in the mixed model, the proposed model is 5.93% higher than F1. The first mock exam is that the implicit text in the data set is very short and there are no sentiment words. This leads to the fact that the mixed model of single model and no attention mechanism cannot extract effective features, resulting in low accuracy. Compared with the CLA model, the hierarchical model of word and sentence attention mechanism can effectively extract the contribution of different words to the sentiment tendency. The results have been improved to a certain extent. The method in this paper not only adds part of speech information to the input layer, but also uses the attention mechanism and attention pooling mechanism of different parts of speech to further strengthen the importance of different
words and characters for sentiment tendency, and achieves remarkable results, which shows that the model proposed in this chapter can be effectively used for implicit sentiment classification.

According to the results in Table 5, compared with the benchmark model, the method proposed in this paper does not achieve the best results. The CLA model achieves the best results on this dataset. This may be because the comment text in this dataset is long, and each comment text has multiple sentences. The model proposed in this paper only adopts one layer structure. When inputting a fixed length, the long comment text is truncated, and some key information may be lost. For the text whose length is less than the fixed length, it needs to be filled, and noise is introduced, which leads to interference information in the subsequent extracted features. Furthermore, further analysis of the data set shows that there are a large number of words with the same part of speech in the sentence, and the model uses different single attention mechanism may not be able to capture deep semantic information. This leads to the problem of insufficient feature extraction when the model extracts long comment text.

5. Conclusions
In this work, we propose a deep learning model, which uses some custom grammar rules, combined with summary words and turning words to filter the Chinese text input into the model, selects the part reflecting the real sentiment of the commentator, effectively reduces the interference information, and uses two channels to extract information from the word and character granularity respectively. Experiments on short text reviews show that the effect is superior to other models. The experimental results show that, compared with the benchmark model, the model proposed in this chapter has some advantages in short text comments and implicit text comments, but the classification effect on the document level data set with more sentences is not superior to the hierarchical CLA model using multi-layer attention mechanism. This also urges us to further explore the better effect of sentiment classification in document level reviews.

References
[1] Wen Chen, Zhiyun Xu, Xiaoyao Zheng, et al, Research on Sentiment Classification of Online Travel Review Text. 2020.
[2] B. Liu, Sentiment analysis and opinion mining, Synth. Lectures Hum. Lang. Technol., vol. 5, no. 1, 2012, pp. 1 - 167.
[3] F. Hemmatian, M. K. Sohrabi, A survey on classification techniques for opinion mining and sentiment analysis. Artif. Intell. Rev., vol. 52, no. 3, 2019, pp. 1495 - 1545.
[4] Y. Kim, Convolutional neural networks for sentence classification, inProc. Conf. EMNLP, 2014, pp. 1746 - 1751.
[5] Kalchbrenner N, Grefenstette E, Blunsom P, A convolutional neural network for modelling sentences [C]. //Proc of the 52nd Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA:ACL, 2014, pp. 655 - 665.
[6] J. Xu, D. Chen, X. Qiu, X. Huang, Cached long short-term memory neural networks for document-level sentiment classification, inProc. Conf. EMNLP, 2016, pp. 1660 - 1669.
[7] Z. Hameed, B. Garcia-Zapirain, "Sentiment Classification Using a Single-Layered BiLSTM Model," in IEEE Access, vol. 8, 2020, pp. 73992 - 74001.
[8] Pang Bo, Lee L, A sentiment education:Sentiment analysis using subjectivity summarization based on minimum cuts [C]. //Proc of the 42nd Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA:ACL, 2004, pp. 271 - 278.
[9] Lee H, Bhd C, Chinese sentiment analysis using maximum entropy[C]. //Proc of the Workshop on Sentiment Analysis Where AI Meets Psychology. New York:ACM, 2011, pp. 89 - 93.
[10] Yin Wenpeng, Schütze H, Multichannel variable-size convolution for sentence classification[C]. //Proc of the 19th Conf on Computational Natural Language Learning. Stroudsburg, PA:ACL, 2015, pp. 204 - 214.
[11] Tang Duyu, Qin Bing, Liu Ting, Document modeling with gated recurrent neural network for sentiment classification[C]. //Proc of the 2015 Conf on Empirical Methods in Natural
[12] Zhou Chunting, Sun Chonglin, Liu Zhiyuan, et al, A C-LSTM neural network for text classification [J]. arXiv preprint, arXiv:1511.08630, 2015.

[13] Sachin S, Tripathi A, Mahajan N, et al, Sentiment analysis using gated recurrent neural networks [J]. SN Computer Science, 2020, pp. 1 - 13.

[14] Wang Xingyou, Jiang Weijie, Luo Zhiyong, Combination of convolutional and recurrent neural network for sentiment analysis of short texts [C]. //Proc of the 26th Int Conf on Computational Linguistics. New York:ACM, 2016, pp. 2428 - 2437.

[15] Zhang Haowei, Wang Jin, Zhang Jixian, et al, Ynu-hpcc at semeval 2017 task 4: Using a multi-channel CNN-LSTM model for sentiment classification [C]. //Proc of the 11th Int Workshop on Semantic Evaluation. Stroudsburg, PA:ACL, 2017, pp. 796 - 801.

[16] Alayba A M, Palade V, England M, et al, A combined CNN and LSTM model for Arabic sentiment analysis[C]. //Proc of the Int Cross Domain Conf for Machine Learning and Knowledge Extraction. Berlin:Springer, 2018, pp. 179 - 191.

[17] Yang Zichao, Yang Diyi, Dyer C, et al, Hierarchical attention networks for document classification[C]. //Proc of the 2016 Conf of the North American Chapter of the Association for Computational Linguistics. Stroudsburg, PA:ACL, 2016, pp. 1480 - 1489.

[18] Wang Linlin, Cao Zhu, Melo G D, et al, Relation classification via multi-level attention CNNs[C]. //Proc of the 54th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PPA:ACL, 2016, pp. 1298 - 1307.

[19] Rongmei Zhao, Xi Xiong, Geng Ju, et al, Chinese implicit sentiment analysis based on hybrid neural network [J]. Journal of Sichuan University (NATURAL SCIENCE EDITION), 2020, pp. 264 - 270.