INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, e-ISSN 1841-9844, 14(2), 272-285, April 2019.

Ensemble Sentiment Analysis Method based on R-CNN and C-RNN with Fusion Gate

F. Yang, C. Du, L. Huang

Fushen Yang
Beijing University of Chemical Technology
Beijing 100029, China

Changshun Du*
Center for Information Technology
Beijing University of Chemical Technology
Beijing 100029, China
*Corresponding author: ducs@mail.buct.edu.cn

Lei Huang
School of Economics and Management
Beijing Jiaotong University
Beijing 100044, China

Abstract: Text sentiment analysis is one of the most important tasks in the field of public opinion monitoring, service evaluation and satisfaction analysis in the current network environment. At present, the sentiment analysis algorithms with good effects are all based on statistical learning methods. The performance of this method depends on the quality of feature extraction, while good feature engineering requires a high degree of expertise and is also time-consuming, laborious, and affords poor opportunities for mobility. Neural networks can reduce dependence on feature engineering. Recurrent neural networks can obtain context information but the order of words will lead to bias; the text analysis method based on convolutional neural network can obtain important features of text through pooling but it is difficult to obtain contextual information. Aiming at the above problems, this paper proposes a sentiment analysis method based on the combination of R-CNN and C-RNN based on a fusion gate. Firstly, RNN and CNN are combined in different ways to alleviate the shortcomings of the two, and the sub-analysis network R-CNN and C-RNN finally combine the two networks through the gating unit to form the final analysis model. We performed experiments on different data sets to verify the effectiveness of the method.

Keywords: Sentiment analysis, convolutional neural network, recurrent neural network, fusing gate.

1 Introduction

The basic task of sentiment analysis is to classify the polarity of a given text at the document, sentence or feature/aspect level and to determine whether the opinions expressed in the document, sentence or entity feature/aspect are positive, negative or neutral. The complexity of textual information leading to the detection of emotions in plain text makes for a challenging task. At present, the sentiment analysis algorithms with good results are all based on statistical learning methods. The performance of this method depends on the quality of feature extractions, while good feature engineering requires a high degree of expertise and is time-consuming, laborious, and offers poor mobility. The neural network approach can reduce the dependency of feature engineering.

Currently, some neural network-based methods have been used for sentiment classification tasks and Socher et al. [8–12] proposed modeling using the Recursive Neural Networks. These methods have proven to be effective in constructing sentence representations. However, recursive neural networks need to construct text as a data structure representation of a tree structure in order to capture the semantics of the sentence. Therefore, to a large extent, the rationality and validity of the text tree structure determines the performance of this type of method. At the same time, the construction of this text tree representation has a time complexity of at least O(n ^{2}), so when such a model is applied to a long sentence or document, significant time overhead is required. In addition, it is difficult to express the relationship between two sentences using a tree structure. Therefore, recursion is not suitable for long text modeling.

Another neural network model commonly used for natural language processing tasks, the Recurrent Neural Network (RNN), has a time complexity of O(n). The model uses the linguistic symbolic order input model in the text to model the text and store all of the text semantics in a fixed-size hidden state. Compared to other methods, RNN has a good ability to capture context information, which is useful for modeling long text and obtaining the semantics of long text. However, RNN is a biased model, and the closer to the end of the expanded word, the more information is retained than the previous word. It is well known that key features related to sentiment analysis may appear anywhere in the document, not just at the end of the text. Therefore, when the RNN is used to capture the semantic features of the entire input text, the change of the position of the key features will result in different degrees of effectiveness reduction, and may even completely ignore the important information, resulting in greatly degraded performance of the model.

At the same time, other work also uses the Convolutional Neural Network (CNN) for sentiment classification [6] . One reason why CNN was introduced into natural language processing tasks is that CNN can solve the problem of sequential deviation of features such as words. This is because it has translation invariance when using pooling operations such as maximum pooling, that is, the semantic features of different positions in the text are unbiased, regardless of the location of the feature. Therefore, compared with recursive or recurrent neural networks, CNN is more conducive to capturing semantic features that are independent of position and order in text. The time complexity of CNN is O(n). However, previous research work on CNN tends to use a simple convolution kernel such as a fixed window size [1,4] . When using such a kernel, it is difficult to determine the size of the window. Small windows can cause important information to be lost, while large windows lead to huge parameter spaces, making network training extremely difficult. At the same time, text analysis methods based on CNN can obtain the important features of text through pooling, but it is difficult to obtain the context information. The piecewise pooling strategy utilized by the sentiment analysis model in Du et al's work can partially alleviate the shortage of CNN [3] . However, modeling for long-distance dependence is still poor. Mathieu Cliche [2] uses CNN and long/short-term memory networks to model sentences separately. Although this method can improve the experimental results, it still cannot overcome the defects of CNN and RNN.

In order to solve the limitations of the above-mentioned recurrent neural network, recurrent neural networks and convolutional neural network models, some work has been attempted to integrate the Recurrent Neural Network and the convolutional neural network. For example, these works [5,13] use convolutional cyclic neural networks to extract sequence features. Unlike other works, this paper proposes a sentiment analysis based fusion Recurrent-Convolutional Neural Network (R-CNN) [7] and Convolutional-Recurrent Neural Network (C-RNN) sentiment analysis method. Firstly, RNN and CNN are combined in different ways to alleviate the shortcomings of the two. The sub-analysis networks R-CNN and C-RNN are constructed respectively. Lastly, the final analysis and analysis model is composed by combining the two networks with a fusion gate. We performed experiments on different data sets to verify the effectiveness of the method.

2 Sentiment analysis method based on R-CNN and C-RNN with fusion gate

2.1 R-CNN-based text sentiment feature extraction model

Firstly, this paper proposes a deep neural model based on the R-CNN to capture the semantics of the text, and uses the obtained semantic features as the characteristics of the sentiment analyzer to analyze the sentiment orientation. Figure 1 shows the network structure of the model in this section. The input to the network is the text S, which consists of the word sequence $w_1, w_2, \cdots, w_n$ of the text.

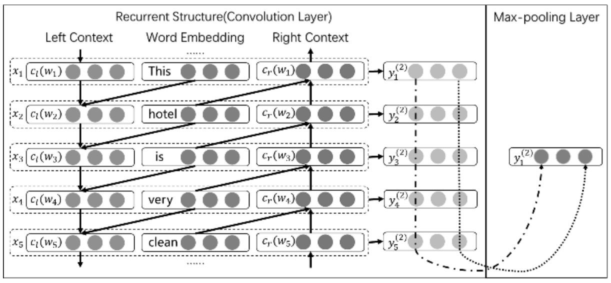

Figure 1: structure of recurrent-convolution neural network

Semantic feature extraction of words

In the text, the meaning of each linguistic symbol is related to the context in which it is located. The contextual location of the word can help the model to obtain a more precise meaning of the word in a particular scene. In order to enable the semantic features extracted by the neural network to fuse the context of the word, this paper uses the R-CNN model to extract features from the customer's comment text. The model first uses a Bidirectional Recurrent Neural Network to capture the contextual information of the word and combine it with the word embedding representation of the word itself to obtain an enhanced representation of the word. The definitions $c_l(w_i)$ and $c_r(w_i)$ represent the left context information (the loop is expanded from the front to the back) and the right context information (expanded from the back), respectively, of the word w_i. $c_l(w_i)$ and $c_r(w_i)$ are realvalued dense vectors of dimension c, and the calculation processes are expressed as Equation 1 and Equation 2, respectively.

$$c_l(w_i) = f(W^l c_l(w_{i-1}) + W^{sl} e(w_{i-1})) \tag{1}$$

$$c_r(w_i) = f(W^r c_r(w_{i+1}) + W^{sr} e(w_{i+1})) \tag{2}$$

$e(w_i)$ represents the word embedding representation of the i-th word w_i in the input text sequence. $c_l(w_i - 1)$ represents the left context information of the previous word w_{i-1}. W^l is a parameter matrix that converts the forward-expanded hidden layer state (left context information feature) of the bidirectional cyclic neural network into the next hidden layer state. W^{sl} is a parameter matrix that embeds the information of the current word into the left context information feature added to the next word. f is a nonlinear activation function. The right context information feature $c_r(w_i)$ is calculated in a similar manner, as shown in Equation 2. In order to conveniently calculate the context information of the first and last words of the input text, a rightmost context feature $c_r(w_n)$ of all text sharing is set to estimate the leftmost context information feature $c_l(w_1)$.

After computing the text sequence using Equations 1 and 2, the left context information feature of the current word captures all semantic information that is expanded from front to back to the word. The right context information feature captures all the semantics that are expanded from back to front to the word. For example, in Figure 1, $c_l(w_5)$ retains the semantics of the context "...this hotel bathroom comparison" made up of all the previous words on the left side of the word "clean." $c_r(w_1)$ retains the semantics of the phrase "the hotel bathroom is clean..." on the right and end of the word "this home". After obtaining the contextual semantic features of the current word, the contextual semantic feature is merged with the word embedded representation of the word itself to obtain the final representation of the current word, and the formal representation is as follows:

$$x_i = [c_l(w_i); e(w_i); c_r(w_i)] \tag{3}$$

As shown in Equation 3, the final representation x_i of the word w_i consists of a splicing of the left context vector $c_l(w_i)$, the word embedding $e(w_i)$, and the right context vector $c_r(w_i)$. In this way, the word representations learned by the model contain rich word context information. Compared to traditional language models or neural network models that use only fixed window sizes, these models use only part of the textual information around the word when processing each language symbol. The RCNN model can model the entire text sequence and preserve the structural information such as the order of the words. The context information can better eliminate the ambiguity of the meaning of the word w_i. At the same time, the RNN structure forward scan obtains all left context semantic features c_l, and the backward scan obtains all right context features c_r, with a total time complexity of O(n), which has high efficiency.

After obtaining the representation x_i of the word w_i the linear activation function tanh transforms the x_i and outputs it to the next layer, namely:

$$y_i^{(2)} = tanh(W^{(2)}x_i + b^{(2)}) \tag{4}$$

$1 \leq i \leq |S|$. $y_i^{(2)}$ is a potential semantic vector that integrates word and context information, so each dimension can be thought of as an element that contains a certain type of semantic information. These semantic elements are useful for extracting features related to sentimental tendencies throughout the input text.

Obtaining the Semantic Features of the Full Text from the Semantic Features of Words

Directly using the representation of a single word in the text to analyze the sentiment orientation of the entire comment is still very difficult, so it is also necessary to use the representation of the words to calculate the representation of the entire comment text. From the perspective of CNN, the aforementioned RNN structure can be considered as the convolutional layer of CNN. When the representation of all words is calculated, the maximum value pooling operation is used to obtain the semantic features of the full text from all word representations of the comment text sequence, as shown in Equation 5:

$$y^{(3)} = \max_{i=1}^{|S|} y_i^{(2)} \tag{5}$$

The maximum value among the features obtained by extracting each convolution kernel convolution operation is different from the maximum pooling operation employed by other work. In the R-CNN model, the maximum pooling operation extracts the maximum value for all word representations of the entire comment text. That is, the maximum value of the k-th dimension of all the word semantic features $y_i^{(2)}$ is selected as the value of the full-text semantic feature $y^{(3)}$ of the k-th semantic element. After the maximum pooling operation, the comment texts with different lengths are converted into fixed-length full-text semantic vectors, which can better preserve the important semantic information of the entire text. There are many types of pooling operations in convolutional neural networks, such as average pooling, minimum pooling, and the like. As mentioned above, in the whole comment, some emotion-related words and their combined semantics are the most relevant features of the sentiment analysis task, and are very effective features for judging the emotional tendency of comments. Extracting these important features can improve the performance of the sentiment analysis model. Therefore, in the model of this section, the maximum pooling operation is still used. The pooling layer uses the output of the RNN as input, and the time complexity of the pooling layer is also O(n). Therefore, the time complexity of the cascaded RCNN sentiment feature extraction model is still O(n), which maintains high efficiency.

2.2 Text sentiment feature extraction model based on C-RNN

In the previous section, this paper proposes the use of RCNN for feature sentiment feature extraction of comment texts. In this section, this article will examine another combination of convolution and recurrent neural networks. That is to say, the convolution operation is performed first, and then the combination of the loop structure is used to propose a feature extraction model based on C-RNN.

Convolution operation segmentation extraction of local emotional features of review text

A standard convolutional neural network usually consists of a convolutional layer and a pooled layer. As mentioned above, the standard convolutional neural network can extract the combined features of words and has translation invariance. These combined features can be extracted anywhere in the text, and convolution operations can be extracted. In the model presented in this section, the input is a matrix of words embedded in the text, and each line of the matrix is a vectorized representation of a linguistic symbol. The convolution kernel slides in the direction in which the text is expanded, that is, the width of the input matrix (the dimension of the word vector) coincides with the width of the convolution kernel. Assuming that the height of the convolution kernel is w, and the width and word vector dimensions are both d, the convolution kernel can be represented as a matrix $W \in R^{w \times d}$. Let the vectorization of the i-th language symbol in the input be represented as s_i, and the input text can be represented by the matrix $S = (s_1^T, s_2^T, \cdots, s_{|S|}^T)$. Then the convolution operation can be expressed as follows:

$$c^j = W \bigotimes S_{j:j+w-1} \tag{6}$$

$1 \leq j \leq |S| - w + 1$, c^j is the eigenvalue extracted by the convolution operation between the convolution kernel starting from the word j and the convolution kernel height being the window.

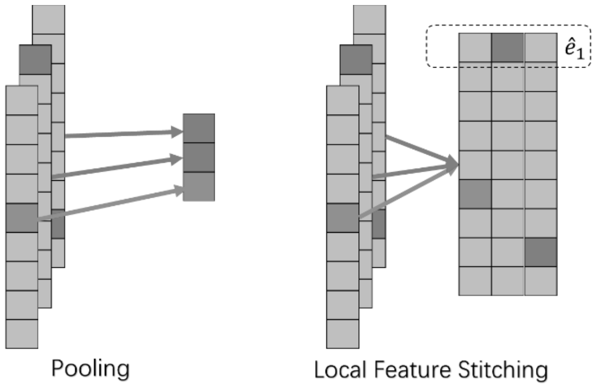

Figure 2: Local feature stitch and pooling

The feature extraction of text by relying solely on a convolution kernel is not comprehensive. In order to extract more abundant information from the text, multiple different convolution kernels are usually used. These convolution kernels can be expressed as a three-dimensional tensor $\hat{W} = \{W_1, W_2, \cdots, W_n\}$, and the convolution operation of the convolutional layer can be expressed as follows:

$$c_i^j = W_i \bigotimes S_{j:j+w-1} \tag{7}$$

$1 \leq i \leq n$, The input text is subjected to the ith convolution kernel convolution operation to obtain the feature vector $c_i = \{c_i^1, c_i^2, \cdots, c_i^{|S|-w+1}\}$. Then all convolution kernels can get a total of n feature vectors. In the model proposed in this section, all convolution kernels are of the same size, so the same comment text after convolution operation will get n vectors with the same dimension and containing the local semantic features of the comment text. The same dimension of these feature vectors can be regarded as different types of features of the same part in the comment, that is, each local semantic feature $\hat{e}_i$ is composed of n features. Different local semantic features may have different n feature strengths. Unlike the traditional convolutional neural network model, the convolutional neural network model in this section no longer uses the pooling layer, but splices the feature vector after the convolution operation into the sequence feature output to the next layer, as shown in Figure 2.

RNN fusion text sentence structure features

After obtaining the local features of the comment text by using the convolutional layer, the local feature is regarded as a text sequence input to the cyclic neural network, and the long-distance dependence of the comment text can be modeled, and the structural features of the sentence are incorporated into the local features of the text.

In this paper, the bidirectional RNN based on LSTM computing nodes is used to perform text representation learning on the stitched feature sequences, and then the feature vectors learned in the two directions are stitched together as a vector representation of the text. Thus, the semantics represented by the feature vector are more comprehensive and rich than RNN. The t-th local feature obtained after splicing is expressed as $\hat{e}_t$, and the one-way calculation process of incorporating the structural features of the text sentence through RNN can be expressed as follows:

$$\begin{cases} i = \sigma(U^i \hat{e}_t + W^i h_{t-1} + b_i) \\ f = \sigma(U^f \hat{e}_t + W^f h_{t-1} + b_f) \\ o = (U^o \hat{e}_t + W^o h_{t-1} + b_o) \\ \tilde{C} = tanh(U^c \hat{e}_t + W^c h_{t-1} + b_c) \\ C_t = C_{t-1} \bigotimes f + \tilde{C} \bigotimes i \\ h_t = o \bullet tanh C_t \end{cases} \tag{8}$$

h_t is the hidden state of the computing node. $\tilde{C}$ indicates that the candidate cell transition is calculated from the previous time hidden state h_t and the current input x_t. C_t indicates the cell state, which is obtained by the forgetting gate and input gate weighting calculation by the cell state C_{t-1} of the $t-1$ step and the cell-selective $\tilde{C}$. h_t is the current output of the computing node, and is also the current hidden state. It is the amount of information that the cell state C_t is finally output through the output gate selection. The local feature of the review text S is forwardly expanded through the RNN to obtain the hidden layer state $\overrightarrow{h_S}$ of the forward sentence structure feature. The backward expansion results in the hidden layer state $\overleftarrow{h_S}$ of the backward sentence structure feature. What is obtained after splicing is the final feature vector h_S representing the input comment text, that is, $h_S = [\overrightarrow{h_S}, \overleftarrow{h_S}]$.

Therefore, the overall structure of the C-RNN-based text sentiment feature extraction model proposed in this paper can be represented as shown in Figure 3. The computational time complexity of the convolutional layer and the cyclic layer is O(n), so the total time complexity of the model proposed in this section is still maintained at O(n), which maintains the efficiency of the model.

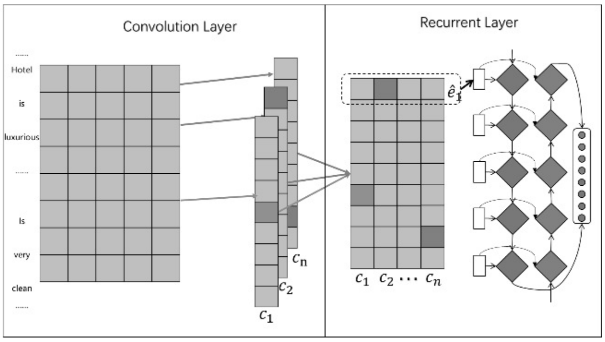

Figure 3: Text sentiment feature extraction model based on convolution-recurrent neural network

2.3 Sentiment analysis model based on R-CNN and C-RNN of with fusion gate

Feature fusion based on gating unit

After R-CNN and C-RNN respectively extract the emotional features of the review text, the features extracted by the two networks need to be integrated and input into the sentiment orientation analyzer to obtain the final sentiment analysis result.

For comments in different segments, there is a difference in the characteristics associated with emotional orientation, that is, emotional tendencies have domain dependence. The comment texts in the same segment have different styles, so there are some differences in the sentiment orientation characteristics. The two combined models of RNN and CNN proposed in this paper are rich in features. However, in different fields, the same feature has different effects on the discrimination of sentiment orientation. Too many features will introduce noise into the sentiment orientation, which degrades the performance of the analyzer. However, relying on manual screening of these features is very difficult. Therefore, an automatic method is needed to select these features to make domain, text adaptability, remove redundant information, and ensure the effectiveness of the sentiment orientation analyzer.

This paper proposes to use the fusion gate to automatically filter and fuse the two features extracted by R-CNN and C-RNN. The process of fusion can be represented in Figure 4.

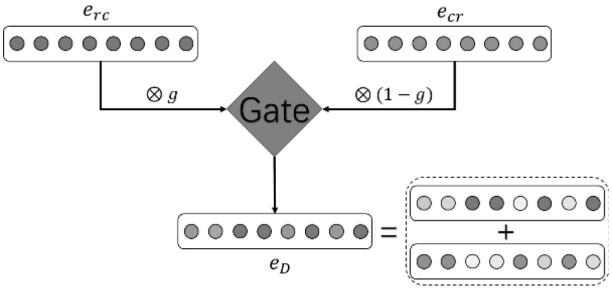

Figure 4: R-CNN combines C-RNN extract features based on gate unit

The fusion gating unit calculates, based on the two characteristics currently input, how many features can be passed through the fusion gating unit, that is, how many original features need to be retained for the two features of the output features. For convenience of representation, the features extracted by the R-CNN described above are denoted as e_{rc}, and the features extracted by the C-RNN are denoted as e_{cr} . The calculation process of the control ratio g of the fusion gating unit to the feature passing is as follows:

$$g = \sigma(U^g e_{rc} + W^g e_{cr} + b^g) \tag{9}$$

U^g and W^g are the weight parameter matrix of the fusion gating unit connected to e_{rc} and e_{cr}, respectively, and b^g represents the offset. In order to simplify the calculation, the fusion gating unit uses the same control ratio for feature extraction of the two extracted features, and the feature fusion calculation process can be expressed as follows:

$$e_D = g \bigotimes e_{rc} + (1-g) \bigotimes e_{cr} \tag{10}$$

e_D is a feature vector that characterizes the input commentary sentiment feature obtained by fusing the two features of the fusion gating unit.

Softmax classifier gets comments emotional sentiment

After obtaining the emotional feature e_D of the review text, the softmax classifier is used to classify the emotional sentiment of the review based on the feature vector, and obtain the emotional tendency of the review.

Set the network parameter W_e of the softmax classification layer, and the bias term is b_e, then the neural network output can be expressed as:

$$o = f(W_e e_D + b_e) \tag{11}$$

f represents the activation function. Then the probability that the entered text sentiment tends to i is:

$$p(i|\theta) = \frac{e^{o_i}}{\sum_{j=1}^{N} e^{o_j}} \tag{12}$$

θ represents all parameters of the neural network, o_i represents the value of the i-th item of the output vector, and N represents the number of categories of the text. Let the sample set be expressed as Ω, then the model's optimization objective function can be calculated by the following formula.

$$L_{sen} = \sum_{i=1}^{|\Omega|} -logp(y_i|S_i, \theta) + \lambda \parallel \theta \parallel_2^2 \tag{13}$$

λ is the parameter of the regular term. In the actual experiment, we use the stochastic gradient descent method to optimize the objective function, then the parameter θ is updated in the way:

$$\theta = \theta - \alpha \frac{\partial L}{\partial \theta} \tag{14}$$

α is the learning rate.

3 Experiment settings

3.1 Data set

Three data sets were used primarily in the experiment. The first is the Chinese Hotel Data Collection (Ctrip Hotel), with a corpus size of 10,000, including 7,000 positive evaluation samples and 3,000 negative evaluation samples. The second dataset is the English dataset, a film review dataset released by Pang and Lee in 2005. The dataset contains a total of 10,662 commentaries on the film, with emotional sentiment being half positive and half negative emotionally biased samples. The sentiment orientation tab of the review reflects the overall emotional sentiment of the reviewer's comments. This paper divides the data set into a training set, a validation set, and a test set, which contain 8530, 1066, and 1066 samples, respectively, with half having positive and half having negative emotional tendencies. The emotional polarity of each sentence is in the range of [0,1]. The smaller the score, the more the emotion tends to be negative. Otherwise, the emotion tends to be positive. The emotional scores of all sentences in the data set are manually labeled and then averaged. It has good reliability. In order to make the experiment closer to the real production environment, and to verify the effectiveness of the sentiment analysis method based on the R-CNN and C-RNN proposed in this paper, the third data set is used in this paper. It is the review text (Dianping for short) in different fields that we crawled from the public service website (http://www.dianping.com/), including data on six segments of food, hotel, movie, entertainment, marriage, and home improvement. And based on the scoring information in the comments, the comments are divided into different emotional tendencies. This paper selects 30,000 reviews as the training set and 10,000 as the test set.

3.2 Data pre-processing

For the Chinese data set, the Chinese word segmentation package NLPIR developed by the Chinese Academy of Sciences is first used for Chinese word segmentation. The English data itself is an independent word, so there is no need for word segmentation. Since the minibatch training model is used during training (multiple samples are learned at a time, the text length of multiple samples may not be the same). At the same time, the use of C-RNN to extract features of different texts needs to ensure the uniformity of dimensions. After convolution, the splicing can guarantee the feature dimension and therefore needs to complete the operation for the length of the text. Since the length of the text of the comment is inconsistent, the longest sentence length l_max is calculated first. For sentences with a sentence length less than l_max, the text must be unified with the $< \backslash s >$ symbol to the length l_max (the vector of $< \backslash s >$ is always set to 0), thus unifying the text length. The length of the unified text can improve the calculation efficiency, and when the length of the data is uniform, the calculation time overhead can be effectively reduced. At the same time, in order to ensure the feature extracted at the beginning and the end of the text during the convolution process, a certain number of $< \backslash s >$ corresponding to the convolution kernel is added at the beginning and end of the longest text as Padding.

3.3 Pre-training of word embedding

Word embedding is required before formal training of the model. Word embedding acts as a distributed representation of words as an input suitable for neural networks. Many current studies have shown that executing word embedding pre-training on a large-scale corpus, and then applying the obtained word embedding to subsequent training, can speed up the convergence of neural network models and achieve a better local optimal solution. In this paper, the word2vec algorithm is used to pre-train word embedding. The word embedding of this algorithm shows better performance in many natural language processing tasks, and it has higher efficiency. This paper chooses the Skip-gram model and the Negative Sampling model to pre-train the word embedding of Chinese and English words. The pre-training of Chinese word embedding uses the text content crawled on Baidu Encyclopedia, and the pre-training of English word vectors is performed on the New York Times corpus.

3.4 Setting of experimental parameters

In the training optimization process of the model, the Adam optimizer is used to train and optimize the parameters of the model. The parameters of the Adam optimizer are set by the author. In this paper, the model mainly has the following hyperparameters: the dimension $\boldsymbol{d}$ of the word vector, the number $\boldsymbol{n}$ of convolution kernels in the C-RNN, the dimension $\boldsymbol{N_{rc}}$ of the hidden state in the loop structure in the R-CNN, and the dimension $\boldsymbol{N_{cr}}$ of the hidden state in the C-RNN. In order to obtain the optimal hyperparameter setting, this paper uses grid-search to determine the value of some hyperparameters. The dimension $\boldsymbol{N_{rc}}$ of the hidden state in the loop structure in the R-CNN and the dimension $\boldsymbol{N_{cr}}$ of the hidden state in the C-RNN are selected from { 50, 100, 200, 300 }. The number n of convolution kernels takes values in { 100, 150, 200 }. In the experiments in this paper, multiple experiments were performed using these parameters, and then the average of the results was obtained.

4 Results and analysis

In order to verify the effectiveness of the proposed R-CNN and C-RNN integrated sentiment analysis method based on the fusion gating unit, the method of this paper is compared with some mainstream sentiment analysis baseline methods.

4.1 Comparison method

In order to verify the validity and correctness of the proposed model, this paper selects a model based on traditional methods and a neural network method such as RNTN proposed by Richard Socher et al. as a baseline method. The first method of comparison is the naive Bayesian method (abbreviated as NB) using the word bag feature. The second method is to use the word bag feature as input to perform emotion classification using a support vector machine (abbreviated as SVM) classifier. The third method is the naive Bayesian method by using the bag feature obtained from the binary grammar language model (abbreviated as BiNB). The fourth method is to use the average word vector of the sentence as the input feature and use the fully connected network as the classifier (abbreviated as VecAvg). The fifth method is the Recurrent Neural Network (RNN). The sixth method is a recurrent neural network (MV-RNN) with a semantic transformation matrix [1] . The seventh method is based on a tensor-based cyclic neural network (RNTN). The last comparison method is the traditional convolutional neural network.

4.2 Analysis of experimental results

It can be seen from the results in Table 1 that the neural network method generally has higher performance than the conventional method. Especially in the five-level sentiment analysis with finer granularity, the neural network method can obtain the key features of the text well. An important reason for the BiNB method to achieve better results is that the binary grammar model considers a certain combination of semantics, but with it comes significant computational overhead. At the same time, compared with the cyclic neural network method, the use of maximum pooling in convolutional neural networks can automatically extract the features most relevant to sentiment analysis tasks, and has positive significance for text sentiment analysis tasks, so it has achieved good results. Traditional methods such as BiNB can only extract combined features from adjacent words, and traditional convolutional neural networks cannot model grammatical structures. The segmented convolutional neural network method has the best effect on both datasets because it simulates the grammatical structure information of the text, supplements the original emotional words, and extracts the combined semantic features of different positions.

In the Dianping data, the data comes from different fields. We compare the performance of different methods in this data to fully demonstrate the effectiveness of the proposed R-CNN and C-RNN integrated sentiment analysis method based on the fusion gating unit. It can be seen from Table 2 that the method proposed in this paper achieves the best results. The comments crawled from the Dianping contain different subdivisions. The comment text in each field is relatively small, and the comment text itself is relatively short. Therefore, samples in different fields have a certain degree of sample sparseness. This need model can effectively extract emotion-related features from the text in order to correctly judge the emotional tendency of the text. This paper also reduces the number of samples in each field, making the data sparse more serious. It can be seen that all methods have performance degradation problems to varying degrees, but the method proposed in this paper still achieves good results. The stability of the proposed method in the case of multi-domain data sparseness is fully explained.

Table 1: Sentiment analysis precision on different data sets(%)

Model	**Stanford**		**Ctrip**
	Two levels sentiment	**Five levels sentiment**	
NB	81.8	41.0	80.2
SVM	79.4	40.7	86.7
BiNB	83.1	41.9	85.9
VecAvg	80.1	32.7	82.1
RNN	82.4	43.2	87.8
MV-RNN	82.9	44.4	87.6
RNTN	**85.4**	45.7	89.3
CNN	81.9	45.6	88.5
R-CNN&C-RNN	**85.8**	**46.1**	**89.8**

Table 2: Sentiment analysis precision on Dianping data sets(%)

Model	**Two levels sentiment**	**Five levels sentiment**	**Two levels sentiment(reduced)**
NB	72.1	35.7	66.1
SVM	72.6	33.2	61.2
BiNB	75.2	36.6	62.4
VecAvg	74.1	36.3	62.3
RNN	77.3	40.2	69.7
MV-RNN	78.0	42.2	73.1
RNTN	80.3	42.6	75.7
CNN	79.1	41.5	73.5
R-CNN&C-RNN	**83.5**	**47.6**	**80.6**

Table 3: Sentiment analysis precision after adding negative words on the dataset(%)

Model	Negated Positive	Negated Negative
BiNB	39.0	27.6
VecAvg	16.5	17.9
RNN	43.3	44.2
MV-RNN	62.4	56.2
RNTN	81.4	72.6
R-CNN&C-RNN	**81.7**	**77.2**

In order to further illustrate the effectiveness of the integrated analysis method of R-CNN and C-RNN based on the fusion gate, this paper selects some samples from the Dianping data set. Adding negative words to the sample converts the emotional tendency of the sample into opposing emotions, called negative positive and negative negative. For example, the “food in this restaurant is delicious" is changed from positive to negative as “the food in this restaurant is not delicious". Turning “this film is ugly" from negative to positive by adding a negative to the original negative (negative negative), “this movie is not ugly." It can be seen that such a change is very subtle for the change of the review text. When the negative word is added to the positive sample, the emotional tendency changed to negative absolutely. However, when the negative word is added to the negative sample, the change of the emotional tendency is tiny. These samples can test the effectiveness and stability of sentiment analysis methods for emotion extraction, and test whether the model can capture emotional details in the text.

The results of the test are shown in Table 3. We can see that the proposed method is superior to all baseline methods. The experimental results show that our model can better capture the emotional features of the review text from the details.

5 Conclusion

In this paper, the performance of traditional text sentiment analysis methods depends on the quality of feature extraction, while good feature engineering requires a high degree of expertise, and is time-consuming, laborious, and affords poor mobility. The neural network approach can reduce the dependency on feature engineering. RNN can obtain context information, but the order in which the features appear in the text will cause the model modeling to be biased. The text sentiment analysis method based on CNN can obtain important features of text through pooling, but it is difficult to obtain context information, and the problem of poor modeling ability for long distance dependence is poor. In this paper, a sentiment analysis method based on fusion gating unit R-CNN and C-RNN is proposed. Firstly, CNN and RNN are combined in different ways to alleviate the shortcomings of the two, and then the sub-analysis networks R-CNN and C-RNN are constructed respectively. Lastly, the final analysis model is composed by combining two networks through the fusion gating unit.

The model proposed in this paper has carried out more detailed experiments on different data, which proves that the proposed method has strong adaptability in different fields. It is more able to extract emotion-related features from text, and maintains good validity in shorter texts and fewer samples. At the same time, the experimental results of adding negative words in the comment text show that the R-CNN and C-RNN integrated sentiment analysis methods based on the fusion gating unit can better capture the emotional features of the review text from the details and that it has good stability.

Author contributions. Conflict of interest

The authors contributed equally to this work. The authors declare no conflict of interest.

Bibliography

[1] Collobert, R.; Weston, J.; Bottou, L.; et al. (2011). Natural language processing (almost) from scratch, *Journal of Machine Learning Research*, 12, 2493-2537, 2011.

[2] Cliche, M. (2017). BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs, *arXiv preprint arXiv*, 1704.06125, 2017.

[3] Du, C.; Huang, L. (2019). Sentiment Analysis Method Based On Piecewise Convolutional Neural Network and Generative Adversarial Network, *International Journal of Computers Communications & Control*, 14(1), 7-20, 2019.

[4] Kalchbrenner, N.; Blunsom, P. (2013). Recurrent convolutional neural networks for discourse compositionality, *arXiv preprint arXiv*, 1306.3584, 2013.

[5] Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. (2014). A convolutional neural network for modelling sentences, *arXiv preprint arXiv*, 1404.2188, 2014.

[6] Kim, Y. (2014). Convolutional neural networks for sentence classification, *arXiv preprint arXiv*, 1408.5882, 2014.

[7] Lai, S.; Xu, L.; Liu, K.; et al. (2015). Recurrent Convolutional Neural Networks for Text Classification, *AAAI*, 333, 2267-2273, 2015.

[8] Luong, T.; Socher, R.; Manning, C. (2013). Better word representations with recursive neural networks for morphology, *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, 104-113, 2013.

[9] Socher, R.(2014). Recursive deep learning for natural language processing and computer vision, *Stanford University*, 2014.

[10] Socher, R.; Chen, D.; Manning, C.D.; et al.(2013). Reasoning with neural tensor networks for knowledge base completion, *Advances in neural information processing systems*, 926-934, 2013.

[11] Socher, R.; Huval, B.; Manning, C.D.; et al. (2012). Semantic compositionality through recursive matrix-vector spaces, *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning. Association for Computational Linguistics*, 1201-1211, 2012.

[12] Socher, R.; Perelygin, A.; Wu, J.; et al. (2013). Recursive deep models for semantic compositionality over a sentiment treebank, *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631-1642, 2013.

[13] Shi, B.; Bai, X.; Yao, C. (2017). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition, *IEEE transactions on pattern analysis and machine intelligence*, 39(11), 2298-2304, 2017.