Membrane Technology
ISSN (online): 1873-4049

Sentiment Analysis Based on Multiscale Convolutional Neural Network and Bidirectional Long Short-Term Memory

Yan Chen, Rongrong Hong*

Department of Investigation, Fujian Police College, Fuzhou, 350007, Fujian, China

**Corresponding author.*

Abstract:

With the prevalence of social media, Weibo has emerged as a crucial platform for individuals to express emotions and opinions, making accurate analysis of its sentiment content significantly meaningful. In order to more effectively capture the semantic and emotional information within Weibo text, a deep learning model that integrates multiscale CNN and BiLSTM is proposed. Firstly, this research utilizes a CNN model with multiscale convolutional kernels to perform feature extraction on Weibo text using various kernel sizes, thereby capturing both local and global information within the text. Subsequently, the incorporation of BiLSTM networks as a component of sequence modeling effectively captures long-distance dependencies within Weibo text. Through the combination of CNN and BiLSTM, the model gains a better understanding of the semantic structure and emotional expressions within Weibo text. Comparative experiments demonstrate that the proposed approach outperforms other sentiment analysis methods, showcasing higher accuracy and better overall effectiveness. This study proposes an innovative model for Chinese microblog sentiment analysis by integrating multi-scale CNN and BiLSTM technologies. It not only enhances the accuracy and efficiency of sentiment recognition but also deepens the theoretical foundation of online sentiment research, exerting significant influence and potential application value in fields such as social media sentiment monitoring and public opinion mining.

Keywords: Sentiment Analysis, Convolutional Neural Network, Long Short-Term Memory Network.

INTRODUCTION

With the rapid development of the Internet and the widespread use of social media, the quantity and frequency of people expressing emotions and opinions on various online platforms have significantly increased. Among these, Weibo, as a platform characterized by strong immediacy and substantial information volume, carries a vast amount of users' emotional information, including attitudes and viewpoints towards events, products, services, and diverse topics.[1] Hence, accurately analyzing and understanding sentiments within Weibo holds crucial significance for comprehending public sentiment, market trends, social mentality, and more.

Sentiment analysis, also known as emotion recognition, primarily aims to extract subjective emotions such as the stance and sentiment orientation of the publisher towards a particular event through the mining and analysis of subjective texts.There are three common approaches to sentiment analysis: sentiment analysis based on sentiment dictionaries, sentiment analysis based on machine learning, and sentiment analysis based on deep learning.The sentiment analysis method based on sentiment dictionaries involves calculating the sentiment value of a text through a pre-constructed sentiment dictionary, and then determining the sentiment orientation of the text based on a certain threshold.[2] Currently, common English sentiment dictionaries include SentiWordNet, General Inquirer, among others.[3] Ahmed et al.[4] argue that the sentiment polarity and intensity of words may vary across different domains, and therefore propose a method for constructing domain-specific sentiment dictionaries.In order to effectively address the issues of limited data volume and lack of colloquial expressions in Chinese sentiment dictionaries, Xu et al.[5] Constructed an extended sentiment dictionary that includes basic sentiment words, scenario-specific sentiment words, and polysemous sentiment words, thereby effectively achieving sentiment classification of texts.The limitation of the sentiment dictionary method lies in the high cost of maintaining the dictionary, especially for short texts like those on microblogs, where irregular expressions also pose significant challenges for dictionary construction. The sentiment analysis method based on traditional machine learning primarily involves feature extraction and selection from a training set, followed by the use of machine learning algorithms to predict the sentiment orientation of samples.Commonly used machine learning analysis algorithms for sentiment analysis include Naive Bayes, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and others.Manek et al.[6] proposed a method for sentiment analysis in large-scale movie reviews, which achieved the extraction of aspect terms relevant to sentiment analysis by combining the Gini index feature selection method with an SVM classifier. Huq et al.[7] applied KNN and SVM to sentiment polarity recognition in Twitter texts and found that KNN, due to its classification based on text Euclidean distance, performed better than SVM. Rathor et al.[8] conducted a comparative analysis of SVM, NB, and ME, three machine learning techniques, incorporating letter weighting, and all of them achieved better classification results with SVM. However, sentiment analysis methods based on traditional machine learning techniques often struggle to capture contextual information and semantic meanings when processing texts, especially when dealing with complex Chinese texts for sentiment analysis, where limitations exist.To address these limitations, researchers have increasingly turned to deep learning techniques, which are capable of automatically learning hierarchical representations of data and capturing nuanced semantic and contextual nuances. These techniques, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and their variants like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs), have shown promising results in sentiment analysis tasks, particularly for Chinese texts, where they can better understand the intricate relationships and nuances within the language.[9] Therefore, in the field of sentiment analysis, deep learning-based methods, particularly models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have garnered significant attention and demonstrated immense potential in processing natural language texts.[10] In 2014, Kim [11] first proposed the TextCNN neural network model, which utilizes multiple sliding windows to extract local semantic information from texts and demonstrated through experiments that CNNs are suitable for natural language processing tasks. RNNs, on the other hand, can accept a wider range of time-series structured inputs and better describe the word order information in texts compared to CNNs.[12] Tang D et al.[13] introduced a gated recurrent neural network sentiment classification model that learns sentence representations through convolutional neural networks or Long Short-Term Memory (LSTM), and then utilizes a gated recurrent neural network to adaptively encode sentence semantics and their relationships into document representations. Experimental results show that the proposed model performs well. Wang et al.[14] proposed an RNN capsule-based sentiment analysis model that, without using any linguistic knowledge, is capable of outputting words with sentiment orientations reflecting capsule attributes. These models can automatically extract features and contextual information from texts, improving the accuracy of sentiment classification. J. Shobana et al.[15] proposed an APSO-LSTM model that utilizes an adaptive particle swarm optimization algorithm to achieve higher accuracy than traditional models. Wu et al.[16] explored a sentiment analysis method based on a two-level LSTM network that combines lexicon embedding and polar flipping techniques to enhance the accuracy and robustness of sentiment analysis.

Through a review of the literature, it is observed that BiLSTM, as a Recurrent Neural Network (RNN) model, possesses strong capabilities in context modeling, while CNNs excel at extracting local features. Based on this, this paper proposes a Chinese microblog sentiment analysis model that combines Multi-Scale CNN and BiLSTM (hereinafter referred to as the "MSCNN-BiLSTM model"). The main contributions are as follows:

(1) Leveraging word embedding techniques to transform microblog samples from a high-dimensional vector space to a low-dimensional vector space;

(2) Extracting important local sentiment features from the text using Multi-Scale CNN, and then capturing contextual information and semantic features with BiGRU (a variant of RNN);

(3) Validating the feasibility and superiority of the model on relevant Chinese microblog datasets.

MODEL CONSTRUCTION

The MSCNN-BiLSTM model is composed of an embedding layer, multiscale CNN layer, BiLSTM layer, and fully connected layer, as illustrated in Figure 1. Firstly, the input text is transformed into word vector representations through the embedding layer. Secondly, the text undergoes feature extraction using three different sizes of convolutional kernels, followed by a max-pooling layer for each convolutional kernel. The features from different kernel sizes are concatenated together. Thirdly, the concatenated features sequence is processed by the BiLSTM layer to capture long-term dependencies. A dropout layer is applied after the BiLSTM output for regularization, reducing the risk of overfitting. Lastly, the classification results are obtained through the fully connected layer.

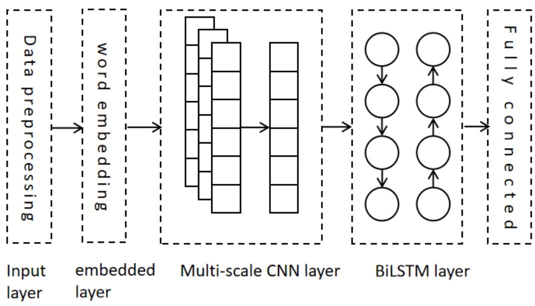

Figure 1. Model structure

Word Embedding Layer

The Embedding Layer is a commonly used neural network layer that transforms discrete words or symbols into dense real-numbered vector representations. In natural language processing tasks, Word2Vec is a commonly used method for word vector representation. Word2Vec is a technique used to represent words as dense vectors in a continuous vector space.[17] It was proposed by Tomas Mikolov and others in 2013 as a word embedding method.[18] The primary concept behind Word2Vec involves training a neural network to learn distributed representations of words, resulting in similar representations in the vector space for words with similar contexts. Word2Vec models typically have two architectures: Continuous Bag of Words (CBOW) and Skip-gram. The CBOW model predicts the target word based on a given context window of words by trying to predict the word vector of the central target word using the average vector of the context words, aiming to maximize the probability of predicting the target word. In contrast, the Skip-gram model predicts context words based on the target word. The Skip-gram model attempts to predict the surrounding context words based on the target word to learn word vector representations. This paper adopts the Skip-Gram model for training word vectors. Given a sentence S=$\{w_1, w_2, ..., w_n\}$, where w_i represents the i-th word in the sentence. After word embedding mapping, each word in sentence S will be mapped to the corresponding word vector. Therefore, the whole sentence S will be mapped to a sequence, namely:

$$Embedding(S) = \{e_1, e_2, ..., e_n\} \tag{1}$$

This mapping allows each word in the sentence to be transformed into a dense vector representation that captures the semantic and syntactic relationships between words.

Multiscale CNN Layer

In natural language processing tasks, textual data is commonly presented in the form of one-dimensional sequences, such as word sequences or character sequences. The convolutional layer is a core component of CNN, performing convolution operations on input data using a series of filters (also known as convolutional kernels). Each filter is a small weight matrix, typically much smaller in size than the dimensions of the input data. These filters slide over the input data and, at each position, perform element-wise multiplication and summation with a local window to compute each element on the output feature map. For a sequence E(T)=[e1e2⋯en], the convolution operation is performed on all contiguous word windows using a filter of size l as shown in equation (2):

$$y_i = f(W * e_{i:i+l-1} + b) \tag{2}$$

Where, $e_{i:i+l-1}$ represents the word window consisting of words, f（$*$）represents the convolution kernel function, W represents the weight matrix, and b represents the bias vector. The feature vector is obtained after the convolution kernel is extracted:

$$Y = [y_1, y_2, ..., y_{n-l+1}] \tag{3}$$

After the convolutional layer, it is common to use a pooling layer to reduce the dimensionality of features while retaining key information. The purpose of the pooling layer is to perform dimensionality reduction on the output feature maps from the convolutional layer to reduce computational complexity and the number of parameters. Common types of pooling include max pooling or average pooling. The computation process of max pooling is illustrated in equation (4):

$$\hat{Y} = max(Y) \tag{4}$$

To better capture diverse features within text data, Convolutional Neural Networks (CNNs) `on. As multiple convolutional kernels are utilized, we obtain an equal number of feature maps corresponding to the number of kernels. To integrate these feature maps, a concatenation operation can be applied, resulting in the final feature vector as follows [19]:

$$F = \{\hat{Y}_1, \hat{Y}_2, ..., \hat{Y}_z\} \tag{5}$$

BiLSTM Layer

LSTM is a variant of RNNs used for handling sequential data. Specifically designed to address the issue of long-term dependencies, LSTM networks are capable of effectively capturing long-range dependencies within sequences.[20] LSTM consists of multiple gating units, each comprised of an input gate, forget gate, output gate, and memory cell. The function of these gating units is to regulate the flow of information and determine when to remember or forget information. The mathematical representation of LSTM is as follows:

The activation value of the input gate:

$$F = \{\hat{Y}_1, \hat{Y}_2, \ldots, \hat{Y}_z\} \tag{6}$$

Activation value of the forget gate:

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \tag{7}$$

Candidate value for updating the cell state:

$$\tilde{c}_t = tanh(w_g[h_{t-1}, x_t] + b_g) \tag{8}$$

Updating the cell state:

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \tag{9}$$

Activation value of the output gate:

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{10}$$

Computing the hidden state for the current time step:

$$h_t = o_t \times tanh(c_t) \tag{11}$$

Where, i_t, f_t, $o_t$respectively represent the activation values of the input gate, forget gate, and output gate, with their values ranging between 0 and 1. $\tilde{c}_t$is the candidate cell state at the current time step, where the hyperbolic tangent function (tanh) is used to maintain the cell state within the range of -1 to 1.$c_t$represents the cell state at the current time step, while h_t denotes the hidden state for the current time step.

A Bidirectional LSTM (BiLSTM) consists of two LSTM units–one responsible for processing the input sequence in the forward direction and the other for processing it in the backward direction. Consequently, at each time step, each LSTM unit has two hidden states, each originating from the forward and backward information flows respectively. The hidden state h_t output by BiLSTM at time t is given by

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{12}$$

Where $\overrightarrow{h_t}$ denotes the hidden state for forward propagation, and $\overleftarrow{h_t}$ denotes the hidden state for backpropagation.

Fully Connected Layer

By passing through one or multiple fully connected layers, we can obtain sentiment predictions.

$$p = Soft\,max(Wh + b) \tag{13}$$

Where W and b represent the weights and biases of the fully connected layer, and p is the output of the Softmax layer, representing the probability distribution of different sentiment categories.

EXPERIMENTAL RESULTS AND ANALYSIS

Experimental Environment

The experimental environment is as follows: CPU: Intel(R) Core (TM) i7-12700KF; GPU: GeForce RTX 3080 Ti; Memory: 32GB; Operating System: Windows 10; Development Language: Python 3.9.7; Deep Learning Framework: PyTorch 1.10.2.

Experimental Data and Evaluation Criteria

The dataset used in this paper was collected from the Sina Weibo platform, focusing on "COVID-19 pandemic." It comprises 12,500 positive sentiment samples and 7,500 negative sentiment samples. For sentiment analysis tasks, several evaluation standards are commonly used, including:

$$Precision = \frac{TP}{TP+FP} \tag{14}$$

$$Recall = \frac{TP}{TP+FN} \tag{15}$$

$$F1 = \frac{2*Precision*Recall}{Precision+Recall} \tag{16}$$

Where Precision is defined as the proportion of samples correctly identified as a certain class compared to the total samples identified as that class; Recall represents how many of the actual samples for a particular class were correctly classified; F1 Score is the harmonic mean of Precision and Recall.

Experimental Parameter Settings

For the CNN network, the parameters are set as follows: learning rate of 0.001, convolutional kernel sizes of 2, 3, and 4, Drop_out of 0.1, activation function as tanh, and the loss function as cross-entropy. For the BiLSTM network, the parameters are set as follows: learning rate of 0.001, hidden layer size of 128, Drop_out of 0.1, and activation function as tanh.

Analysis of Experimental Results

Model comparison experiment analysis

To validate the effectiveness of the models, this paper conducted the following comparative experiments:

CNN: Single CNN network.

LSTM: Single LSTM network.

BiLSTM: Single BiLSTM. BiLSTM combines forward LSTM and backward LSTM to obtain bidirectional semantic information.

The comparative experimental results between the MSCNN-BiLSTM model and the aforementioned baseline models on the dataset are presented in Table 1.

Table 1. Test results for each model %

Model	Precision	Recall	F1
CNN	78.2	77.5	77.1
LSTM	76.1	76.0	75.8
BiLSTM	76.8	76.7	76.6
MSCNN-BiLSTM	80.1	80.1	80.0

As can be seen from the table, the MSCNN-BiLSTM model has improved the accuracy rate by 1.9%, 4.0%, and 3.3% compared to other deep learning models CNN, LSTM, and BiLSTM, respectively; the F1 score has increased by 2.9%, 4.2%, and 3.4%, respectively. Compared to other models, the proposed model in this paper has improved both accuracy and F1 score, indicating that the fusion of multi-scale CNN and BiLSTM models can obtain richer semantic information, thereby improving the performance of the model. At the same time, as can be seen from the table, BiLSTM has improved the accuracy rate and F1 score compared to LSTM model, indicating that when processing sequence data, one-way LSTM may encounter long-term dependency problems, which means it is difficult to effectively capture long-distance dependencies when the sequence is long. BiLSTM combines the information flow in both forward and backward directions, and can consider both past and future information at the same time, enabling a more comprehensive understanding of sequence data.

Comparison experiment of different convolution kernels

Convolution kernels with different sizes and shapes can capture features of different scales and forms, allowing the model to better understand the input data. In this paper, six sets of convolution kernels, including (2,3,4), (3,4,5), (4,5,6), (3,5,7), (5,7,9), and (7,8,9), were selected for comparative experiments. The experimental results are shown in Table 2. From Table 2, it can be seen that the model performance is best when the convolution kernel is (2,3,4). The experimental results show that smaller convolution kernels can usually extract more local and fine-grained features, while larger convolution kernels can extract more global and macroscopic features. By adjusting the size of convolution kernels appropriately, the model can better adapt to the characteristics of the input data and improve the feature extraction ability of the model on different scales.

Table 2. Experimental results of different convolution kernels %

Convolution kernel size	Precision	Recall	F1
(2,3,4)	80.1	80.1	80.0
(3,4,5)	79.9	79.9	79.7
(4,5,6)	79.4	79.2	79.0
(3,5,7)	79.5	79.1	78.8
(5,7,9）	78.8	78.5	78.6
(7,8,9）	78.8	78.6	78.2

CONCLUSIONS

Chinese microblogs are one of the primary avenues for people to express their emotions and opinions on social media platforms. By analyzing the sentiments expressed in Chinese microblogs, we can gain a deeper understanding of the public's emotional attitudes, preferences, and tendencies, thereby gaining insights into social opinions and popular sentiment trends. Deep learning-based sentiment analysis methods can automatically learn textual features, reducing reliance on manual design and enhancing performance and accuracy. In terms of theoretical contributions, this paper constructs a novel Chinese microblog sentiment analysis model based on Multi-Scale Convolutional Neural Networks (MCNN) and Bidirectional Long Short-Term Memory (BiLSTM). This model can capture a richer semantic representation of the input text, providing new theoretical support for sentiment recognition in web texts. In terms of practical contributions, this paper utilizes Chinese microblog comments as a dataset to verify the feasibility and superiority of the model. The limitation of this study is that the model has only been validated on one public dataset; future research will involve comparative experiments on other datasets to further evaluate the model.

REFERENCES

[1] Zhou X Z, Li S, Sui D. Weibo Sentiment Analysis Based on Deep Learning and Attention Mechanism. Journal of Nanjing Normal University (Natural Science Edition), 2023, 46(02):115-121.

[2] Pan D, Yuan J, Li L, et al. Deep neural network-based classification model for Sentiment Analysis//2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC). IEEE, 2019: 1-4.

[3] Deng S Y, Sinha A P, Zhao H M. Adapting Sentiment Lexicons to Domain-Specific Social Media Texts. Decision Support Systems, 2017, 94: 65-76.

[4] Ahmed M, Chen Q, Li Z. Constructing domain-dependent sentiment dictionary for sentiment analysis. Neural Computing and Applications, 2020, 32（18）: 14719-14732.

[5] Xu G, Yu Z, Yao H, et al. Chinese text sentiment analysis based on extended sentiment dictionary. IEEE access, 2019, 7: 43749-43762.

[6] Manek A S, Shenoy P D, Mohan M C, et al. Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier. World Wide Web-internet & Web Information Systems,2017,20(2):135-154.

[7] Huq M R, Ahmad A, Rahman A. Sentiment analysis on Twitter data using KNN and SVM. International Journal of Advanced Computer Science and Applications, 2017, 8(6).

[8] Rathor A S, Agarwal A, Dimri P. Comparative study of machine learning approaches for Amazon reviews. Procedia computer science, 2018, 132: 1552-1561.

[9] Huang F L, Lian Y F. Senti-LSTM: A Sentiment Analysis Model Based on Recurrent Neural Networks. Journal of Fujian Normal University (Natural Science Edition), 2020, 36(01): 12-18.

[10] Wang C D, Zhang H, Mo X L, et al. A Survey of Microblog Sentiment Analysis. Computer Engineering and Science, 2022, 44(01): 165-175.

[11] Kim Y.Convolutional neural networks for sentence classification//Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing, 2014：1746-1751.

[12] Han Pu, Zhang Zhanpeng, Zhang Mingtao, Gu Liang. Research on Chinese Disease Name Normalization Based on Multi-Feature Fusion. Data Analysis and Knowledge Discovery, 2021, 5(05): 83-94.

[13] Tang D, Qin B, Liu T. Document modeling with gated recurrent neural network for sentiment classification//Proceedings of the 2015 conference on empirical methods in natural language processing. USA: Association for Computational Linguistics, 2015: 1422-1432.

[14] Wang Y Q, Sun A X, Han J L, et al. Sentiment analysisby capsules//Proceedings of the 2018 World Wide WebConference, Lyon, Apr 23-27, 2018. New York: ACM, 2018:1165-1174.

[15] Shobana J, Murali M. An efficient sentiment analysis methodology based on long short-term memory networks. Complex & Intelligent Systems, 2021, 7(5): 2485-2501.

[16] Wu O, Yang T, Li M, et al. Two-level LSTM for sentiment analysis with lexicon embedding and polar flipping. IEEE Transactions on Cybernetics, 2020, 52(5): 3867-3879.

[17] Ge X Y, Zhang M S, Wei Bin, et al. Explainable Rumor Detection Based on Dual Emotion Perception. Journal of Chinese Information Processing, 2022, 36(09):129-138.

[18] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [EB/OL]. arXiv preprint arXiv:1301.3781. 2013. https://arxiv.org/abs/1301.3781.

[19] Qiu H, Fan C, Yao J, et al. Chinese microblog sentiment detection based on CNN-BiGRU and multihead attention mechanism. Scientific Programming, 2020,10(15);1-13.

[20] Hochreiter S, ,Schmidhuber J. Long Short-Term Memory. Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997.