Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/39/20230583

Sentiment analysis based on long short-term memory model

Jiahao Deng[1], Qing Wang[2, 4], and Zi Ye[3]

[1]School of Information Engineering, Tianjin university of commerce, Tianjin, 300133, China
[2]College of Applied Science and Technology, Beijing Union University, Beijing, 100012, China
[3]College of Information&Electrical Engineering, China Agricultural University, Beijing, 100083, China

[4]2022190432104@buu.edu.cn

Abstract. An important area of natural language processing is text emotion analysis. Emotion analysis is a very meaningful research work and has a broad application prospect, such as social media monitoring, reputation research of commodity brands, market research, and so on. By analyzing the time and content factors of the data, considering the final application scenario, and finally comparing the advantages and disadvantages of the methods, There are three main techniques for analyzing emotions in text: emotion analysis using an emotion dictionary, emotion analysis using machine learning, and emotion analysis using deep learning. Among them, Convolutional neural networks (CNN), recurrent neural networks (RNN), and long short-term memory networks (LSTM) are examples of deep learning-based techniques. For processing temporal relate issues such as video, speech, and text, CNN algorithms often consume a large amount of computational time, especially for processing image datasets, which may encounter specific problems. To address this issue, RNN is more suitable for solving temporal related issues such as video, voice, and text. In natural language, word order is an extremely important feature. RNN may potentially process sequences of any length, add memory units based on the original neural network, handle pre- and post-word dependencies, and process sequences of any length. The main work is as follows: LSTM adds or deletes unit states through a structure called gate, Determine the experimental thinking of text analysis, Crawl and train data sets through AI Studio, pycharm and other tools.

Keywords: RNN, Long Short-Term Memory (LSTM) Model, Sentiment Analysis.

1. Introduction

As the number of Internet users continues to increase, people's daily activities on the Internet, such as purchasing products, commenting, publishing content, and so on. While generate a large amount of data on the Internet. The authors uses comments as an example. Whether it is buying goods through e-commerce platforms, expressing opinions on social platforms, or leaving impressions on the Internet, the data will be retained. These data are often closely related with the subjective wishes of users. Although the majority of the user's data's emotions are complicated, they may be split into two groups: good emotions and negative emotions. Text sentiment analysis has become an important research topic

DOI: 10.54254/2755-2721/39/20230583

because the emotional factors behind various text information are very important for the analysis of some tasks.

Text sentiment analysis refers to the study of texts with subjective intention and emotion, mining the emotional tendency behind the text information, and classifying the degree of emotion. One of the key subfields of natural language processing is text sentiment analysis [1]. The three types of sentiment analysis approaches are based on distinct techniques: deep learning, classical machine learning, and sentiment analysis based on a sentiment dictionary [2]. The sentiment analysis method is shown in Figure 1.

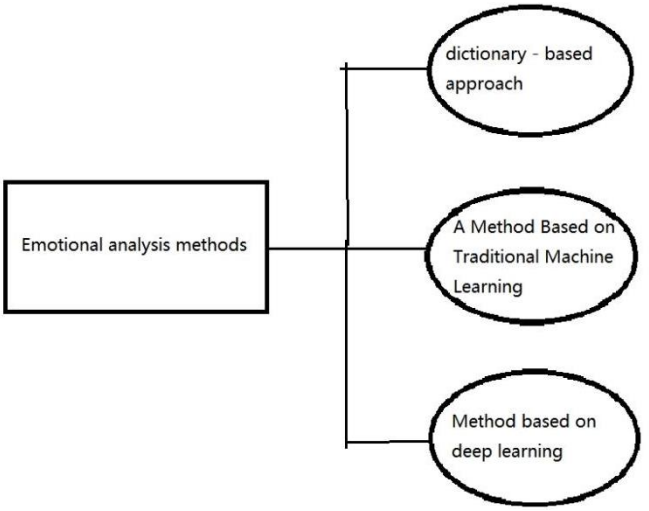

Figure 1. Methods of Emotional Analysis.

Among them, The CNN algorithm, RNN algorithm, and LSTM algorithm are deep learning-based techniques. For timing-related issues such as video, voice, and text, CNN often consumes a lot of computing time, especially for longer texts. RNN will have problems such as gradient explosion and gradient disappearance [3]. Longer sequences can be handled by LSTM, and it performs better for solving long-term dependent problems like the gradient disappearance issue, so it can better handle the task of text sentiment analysis.

2. Methodology

2.1. RNN

In this project, authors used LSTM, which is a variant based on RNN. The RNN model was modified to address the problem of Vanishing gradients in long sequences. Figure 2 demonstrates the structure of the RNN model.

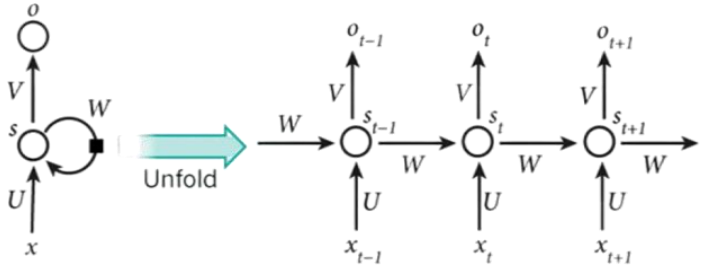

Figure 2. Structure of RNN.

The unexpanded drawing approach is on the left side of the above figure, while the unfolded drawing method is on the right. In Figure 3 [4], the internal organisation is displayed.

DOI: 10.54254/2755-2721/39/20230583

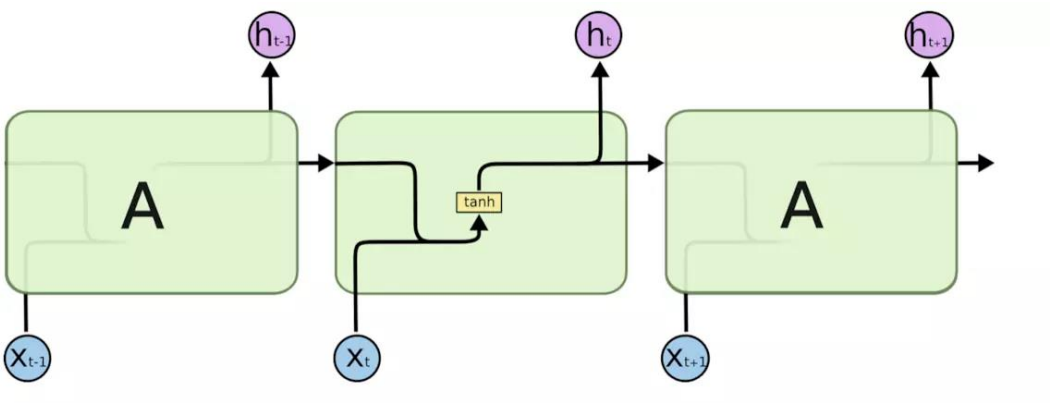

Figure 3. Internal structure of RNN.

The calculation formula can be expressed as:

$$y_t = g(V \cdot s_t + d) \quad (1)$$

Among them, x_t represents the input at time t; s_t indicates the hidden state at time t, f and g represent the Activation function; Input layer hidden layer weight, hidden layer output layer weight, and hidden layer hidden layer weight are represented by the letters U, V, and W, respectively. All weights and biases are shared for each moment t, drastically reducing the number of model parameters.

When calculating, authors utilise the backpropagation algorithm to transmit errors after using the forward propagation technique to calculate in chronological order. The BPTT (Back Propagation Through Time) algorithm's only distinction from the standard BP (Back Propagation) network is the addition of the time sequence and a slightly different calculating approach.

Due to computational reasons, RNN is prone to gradient vanishing, resulting in it being unable to learn information from the past and handle long sequence and long-term dependent tasks. In actual projects, authors will normalize the input so that the input value is less than 1 and the weight is less than 1. The multiplication of the two will decrease. RNN model training can be hindered by gradient vanishing and gradient explosion. The occurrence of gradient vanishing and gradient explosion is caused by the cyclic multiplication of RNN's weight matrix, and multiple combinations of the same function can lead to extreme nonlinear behaviour. Due to the weight matrix used for each time slot in RNN, gradient vanishing and gradient explosion are common. Although NDDs involve multiple matrix multiplication, it is possible to avoid gradient disappearance and gradient explosion problems by carefully designing the weight ratio [5].

2.2. LSTM

A RNN version is the Long Short-Term Memory (LSTM) model. In 1997, Schmidhuber and Hochreiter [6] argued that the distinctive design structure of LSTM enables it to effectively handle longer sequences, solve long-term dependent tasks, and address the Vanishing gradient problem. Large-scale deep neural networks can be built relatively effectively using LSTM [7]. In 2009, using the enhanced LSTM, they took first place in the International Document Analysis and Recognition Competition (ICDAR) Handwriting Recognition Competition. In 2014, Yoshua Bengio's team proposed an improved LSTM variant called the GRU (Gated Recurrent Unit). In 2016, Google used LSTM [8] for speech recognition and text translation. The same year, Apple used LSTM to improve the Siri applications. Its structure is shown in Figure 4 [4].

DOI: 10.54254/2755-2721/39/20230583

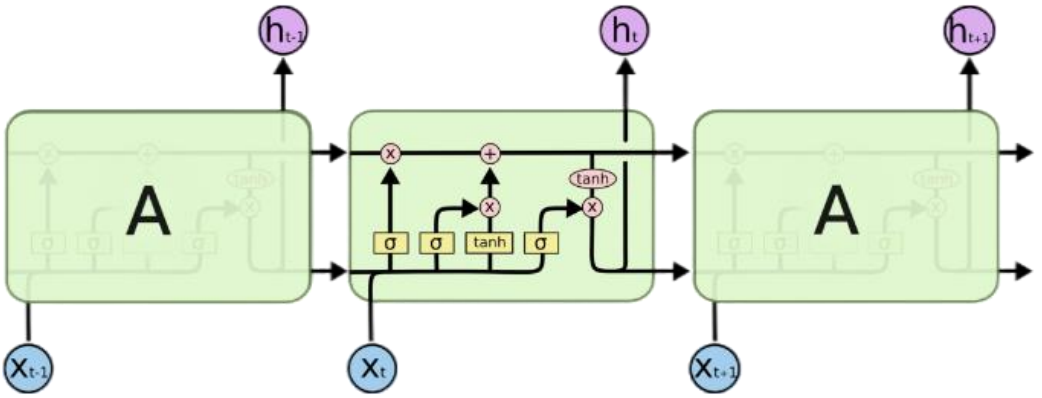

Figure 4. LSTM Structure Diagram.

The cellular state, which is represented by horizontal lines passing between the cells, is the fundamental component of LSTM. Similar to a conveyor belt, the cell state. With only a few branches, it traverses the entire cell, guaranteeing that information moves unchanged throughout the entire framework. Through a structure known as a gate, which consists of three gates, the LSTM simultaneously adds or removes unit states. Forgetting Gate: Figure 5 illustrates how to decide which data to reject.

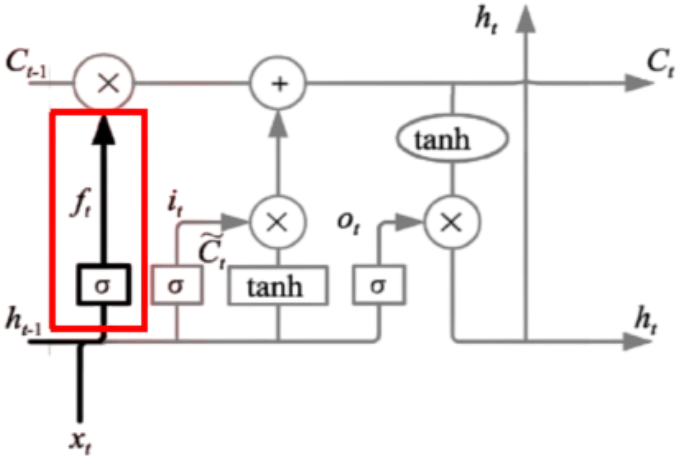

Figure 5. Forgetting Gate Structure Diagram.

The expression is:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{2}$$

Input is x_t the hidden state of the previous moment is h_{t-1} enter the multiplication weight w_f representing the weight matrix of the forgetting gate, the input is multiplied by the weight and biased to output a value under the influence of sigmoid. When the output is 1, it indicates complete retention, and when the output is 0, it indicates complete discarding Input gate: Determine which information to input as shown in Figure 6.

DOI: 10.54254/2755-2721/39/20230583

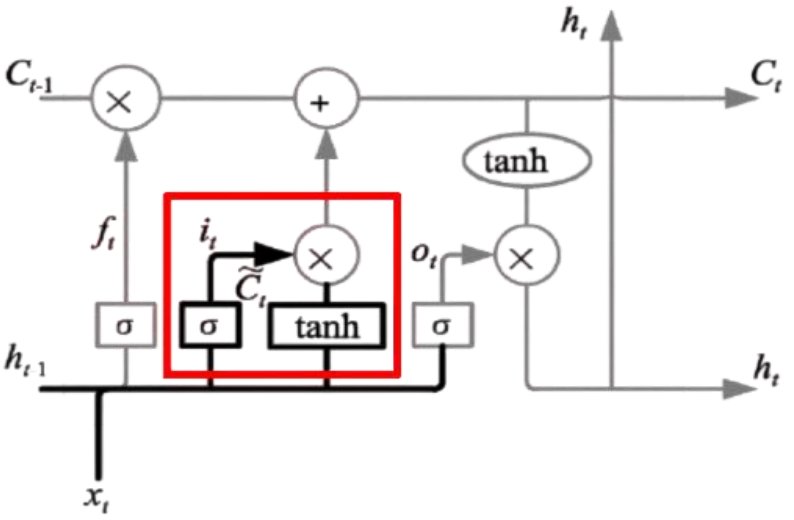

Figure 6. Input Gate Structure Diagram.

The expression is:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

$$\tilde{C}_t = \tan h(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{4}$$

The input gate is composed of two sub neural networks. Afterwards, these two sub neural networks will be combined together. These are two neural networks, so they use different Activation function.

The update of status will include data from the forgotten gate and input gate. The data of the forgetting gate determines which information is deleted, and the information of the input gate determines which important features are input, which can update the cell state, as shown in Figure 7.

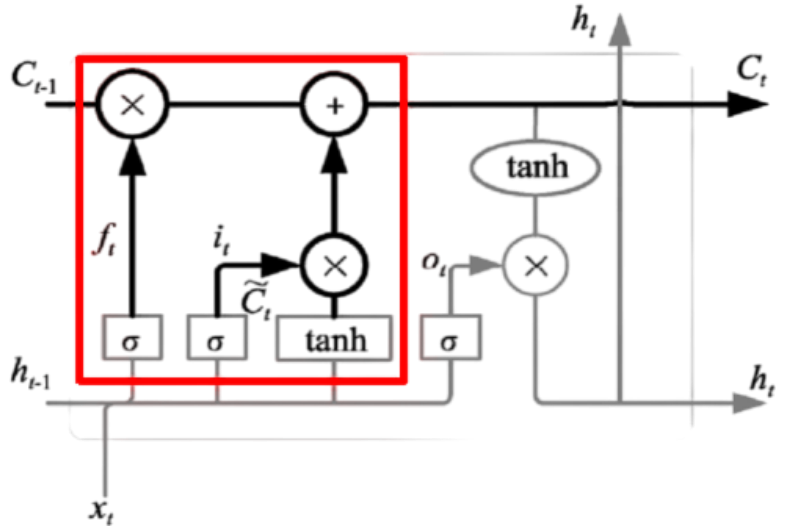

Figure 7. State update structure diagram.

The state update expression is:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{5}$$

Information that needs to be forgotten has been discovered by Forgotten Gate. Next, compare it to the old (C_{t-1}) state multiply and delete the information that should be deleted. The update of the cell

DOI: 10.54254/2755-2721/39/20230583

state has been completed by adding the result $i_t \cdot \tilde{C}_t$ and obtaining new information about the cell state. Forgotten Gate has discovered information that should have been forgotten. Afterward, compare it to the previous state (C_{t-1}) and multiply and discard the information that needs to be discarded. By collecting new cell status information, the update of the cell status is complete. Ensure that the result is added again.

The output gate determines which information to output, as shown in Figure 8.

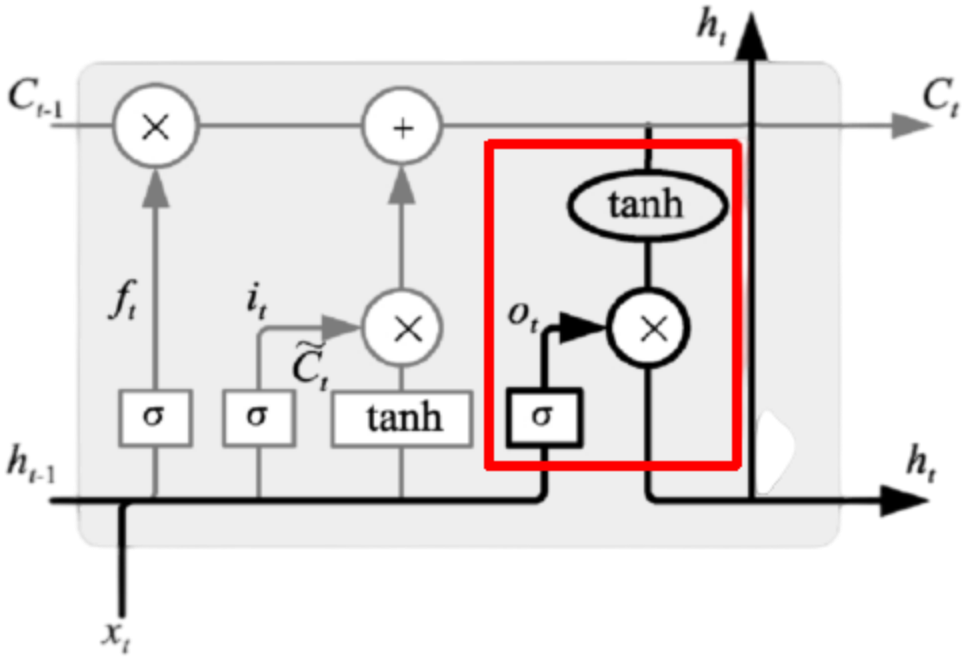

Figure 8. Output Gate Structure Diagram.

The output gate expression is:

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = O_t \cdot \tan h(C_t) \tag{7}$$

To determine which information will be output in the output gate, a sigmoid layer is utilized and then the cell state is passed through the $\tan h$ process (obtain a value between -1 and 1) and multiply it with the output of the sigmoid gate to determine the final desired output. In [9]'s paper, it was pointed out that by integrating b_o initializing the mean of LSTM to 1 can achieve similar effects to GRU.

2.3. Other LSTM

Lenovo previously introduced GRU [10], and the number and working mode of hidden layer nodes in LSTM seem to be very flexible. Is there a best structural model or a model with better performance than LSTM and GRU? Rafal et al. collected 100 of the top models that could be collected, and then generating 10000 new models using mutations based on these 100 models. Then, through experiments in four scenarios: string, structured document, language model, and audio, more than 10000 models were compared. The important conclusions drawn are summarized as follows:

The best performing models are GRU and LSTM, with GRU outperforming LSTM in all scenarios except for language models. When the mean bias of the output gate is set to 1, the performance of LSTM is comparable to that of GRU. In LSTM, the order of gates is based on forgetting gate>input gate>output gate.

DOI: 10.54254/2755-2721/39/20230583

3. Experimental part

3.1. Text classification

An increasing amount of text data is being transmitted on the network due to the continuous development of the Internet. More and more textual data is being transmitted over the network because of the continued development of the Internet. The management and integration of textual data is critical. The use of text classification is crucial for natural language processing, and it is primarily used for public opinion detection, news text classification, and other fields to sort and categorize text resources. The use of deep learning for text classification has been successful in achieving good classification performance in text data processing [11].

3.2. Concept and application of text categorization

The process of text classification involves dividing texts into different categories based on their content. For example, in a news system, every news report is categorized into various categories. It can categorize the content of the text. It not only filter emails such as spam filtering but also categorise users based on text content such as shopping mall consumption level and preferences. In addition, it can perform sentiment analysis on the content of comments, articles and conversations.

3.3. Experimental Purpose

Use the training data set to form the template and obtain an analysis of the feelings of the Chinese comment phrases. There are two types of emotions: positive and negative.

3.4. Experimental ideas

This project is essentially a classification issue. If authors want to analyse whether the sentence expresses positive or negative emotions, authors actually need to do a binary classification. The model here is shown in Figure 9.

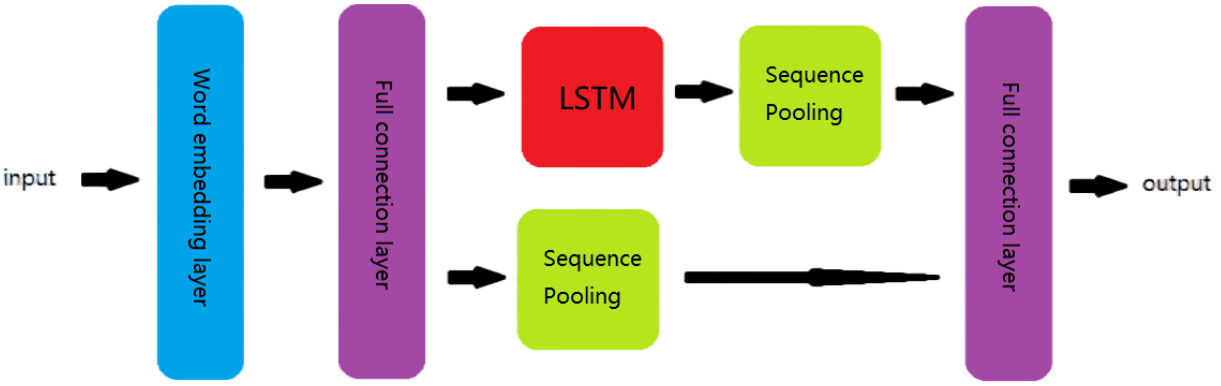

Figure 9. Project Structure Diagram.

Authors used the LSTM model. First, the Word embedding of each word is generated through a Word embedding layer, and divided into two paths through a full connection layer. Hand it over to LSTM to extract the dependency relationships before and after the sequence, and then extract the sequence features and perform pooling. Another column is directly pooled on fully connected output features.

3.5. Experimental steps

Firstly, it is necessary to clearly divide the steps for achieving the experimental objectives. The step diagram is shown in Figure 10.

DOI: 10.54254/2755-2721/39/20230583

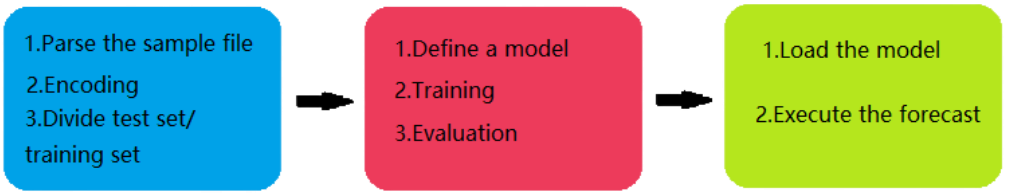

Figure 10. Project step diagram.

After dividing the steps, authors need to determine the dataset. The data for this project comes from Flying Paddle AI Studio. The content is Chinese reviews about hotels, including a total of 5265 user review data, of which 2822 are positive reviews and the rest are negative reviews. Then authors need to build a platform for code writing. This project uses PyCharm under the Linux system for writing. For Python programming, PyCharm is a handy, visual piece of software. That is why authors chose PyCharm for our programming platform.

Firstly, the authors divide the code into three main parts and write the corresponding code. Authors divide the procedure into three parts, namely preprocessing, model training and testing. In addition, these are the three steps of our current experiment.

Secondly, authors chose to conduct this experiment on AIStudio due to the performance of the device. AI Studio is a one-stop online development platform based on PaddlePaddle, Baidu self-developed deep learning framework. The startup environment (using GPU operations) and copy the code for the data preprocessing section to the first code block.

After the run is completed, two other TXT documents appear in the data directory, hotel_ Dict.txt is the dictionary file path, hotel_ Encoding.txt is the encoded dataset.

Authors copy the code from the model training section to the second code block. Authors chose to train this model 200 times due to equipment performance limitations.

In order to get the experimental results, first authors have to copy the test code to the third code block. Authors provided six utterances as a test sample. The first three statements have more positive emotions and the last three have more negative emotions. This was done to test whether the model is able to judge positive and negative emotions in a more balanced way. The six statements are:

Statement 1: “Overall the rooms are very clean, the bathrooms are quite nice, and the transport links are quite convenient.”

Statement 2: “The hotel is conveniently located, the environment is also good, just next to our place of business, I feel that the price-performance ratio is still okay.”

Statement 3: “The facilities are okay, the service staff is good, and the traffic is quite convenient.”

Statement 4: “The hotel has very poor service and poor facilities.”

Statement 5: “The worst hotel I’ve ever stayed in. I’ll never stay there again.”

Statement 6: “I was honestly disappointed and I don’t think I’ll be going back to this hotel in the future no matter what!”

The test results are shown in Figure 11.

DOI: 10.54254/2755-2721/39/20230583

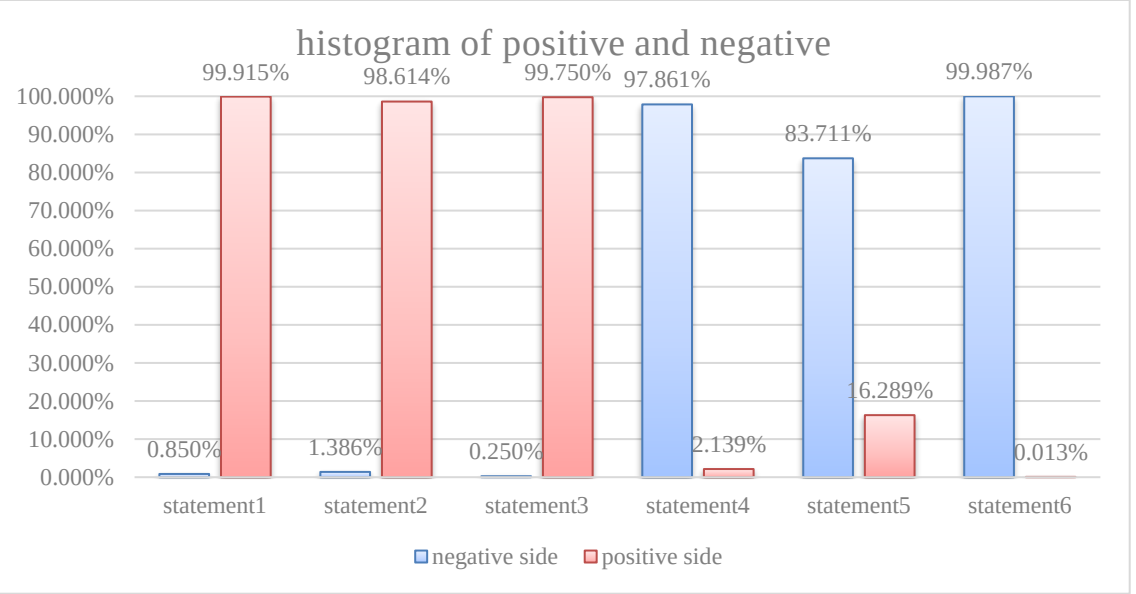

Figure 11. Test results.

It can be seen that for the test statements of this project (6 in total), the model authors used can judge them correctly. Although there may be the problem of a small sample size, the test results are sufficient to prove the correctness of our model and its usefulness in real-life scenarios.

4. Conclusion

This project carries out Sentiment analysis on consumers' comments on hotel services. The project applies Long short-term memory (LSTM) Model, which can better solve the problem of gradient disappearance. This project preprocesses, trains, and infers the model to achieve sentiment analysis of Chinese comment sentences. The model first generates each word's Word embedding through a Word embedding layer, and then divides it into two paths through a full connection layer. Hand it over to LSTM to extract the dependency relationships before and after the sequence, and then extract the sequence features and perform pooling. The other column is directly pooled on the fully connected output features. The final test results are divided into two types of emotions: positive and negative. There are also some shortcomings in this project, such as too little test data, the results may not be of guiding significance, the LSTM model is relatively more time-consuming, and cannot parallelize the data well, making it difficult to process long sequence data. So the improvements on the project will be made by increasing test data, and in the future, the variants of the developed better LSTM model to further will be used to solve problems such as computational efficiency, parallelization processing, and long sequence data processing, thereby improving the persuasiveness of the results. Sentiment analysis is the current research hotspot, and more achievements are believed that can be achieved through efforts in the future.

Authors Contribution

All the authors contributed equally and their names were listed in alphabetical order.

References

[1] Zhang X, 2023, Sentiment analysis Based on Self Attention Mechanism and LSTM, Xiamen University of Technology.

[2] Wang T and Yang WZ, 2021, Review of Research on Sentiment analysis Methods [J], Computer engineering and Application,57 (12): 11-24.

DOI: 10.54254/2755-2721/39/20230583

[3] Zhu Y and Chen S, 2020, Comment on the fusion of Convolutional neural network and attention Sentiment analysis [J], Minicomputer, 03.
[4] Christopher Olah, 2015, Understanding LSTM Networks [J].
[5] Sussillo D and Random W, 2014, Training Very Deep Non linear Feed Forward Networks with Smart Initialization [J], two thousand and fourteen.
[6] Su H, Chai M, Chen L et al, 2021, Deep Learning Based Model Predictive Control for Virtual Coupling Railways Operation, IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA.
[7] Wang Z and Wang G, 2017, Application of LSTM network in short-term power load forecasting under deep learning framework [J], Power Information and Communication Technology, (5): 8-11.
[8] Sundermeyer M, Schl A, Ter R and Ney H, 2012, LSTM Neural Networks for Language Modeling, Interspeech two thousand and twenty-one.
[9] Gers FA, 1999, Learning to target: continuous prediction with LSTM, 9th International Conference on Artistic Neural Networks: ICANN'99.
[10] Cho K, Van MB, Gulcehre C et al, 2014, Learning Phrase Representations using RNN Encoder--Decoder for Statistical Machine Translation, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar.
[11] Jia P and Sun W, 2020, A Review of Text Classification Based on Deep Learning [J].