International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 11 Issue 10 ǁ October 2023 ǁ PP. 213-219

Design of a Sentiment Classification Model for Product Reviews Based on Deep Learning

Demao Xiong[1], Zhan Wen[1,2], Hong Jiang[3], Xiaoqing Zhang[3], Wenzao Li[1,2]
[1] College of Communication Engineering, Chengdu University of Information Technology, Chengdu, 610225, China;
[2] Meteorological information and Signal Processing Key Laboratory of Sichuan Higher Education Institutes of Chengdu University of Information Technology, Chengdu, 610225, China;
[3]Sichuan Guangfeishi Information Technology Co., Ltd, Chengdu, 610225, China.
Corresponding Author: Zhan Wen, email:wenzhan@cuit.edu.cn

Abstract
The rapid growth of the internet and e-commerce has led to a rising preference among consumers for online shopping, which has, in turn, boosted the success of online shopping platforms. Product reviews play a pivotal role in assisting other consumers in making informed choices regarding suitable products. They also empower businesses to enhance their product quality control and post-sales services, facilitating timely adaptations to meet consumer and market requirements. Nonetheless, the issue at hand pertains to the substantial quantity of product reviews, rendering it impractical to manually scrutinize each and every one of them. To enhance the efficiency and accuracy of sentiment classification for product reviews, we have designed and implemented a deep learning-based model. This model utilizes a structure built upon training a Long Short-Term Memory (LSTM) network. Through multiple iterations, the model achieves high accuracy and improves the efficiency of sentiment classification for reviews.

Keywords: *Deep Learning, Sentiment Classification, Long Short Term Memory*

Date of Submission: 09-10-2023 Date of acceptance: 23-10-2023

I. INTRODUCTION

As e-commerce platforms continue to gain widespread popularity, product reviews have emerged as an essential factor influencing consumers' purchasing choices. Nevertheless, the sheer abundance of reviews presents a formidable challenge for both merchants and consumers when it comes to efficiently categorizing and analyzing each review. This challenge, in turn, poses significant hurdles to ensuring product quality and delivering an optimal user experience. Consequently, the utilization of natural language processing (NLP) techniques for sentiment analysis of product reviews becomes pivotal for merchants in their efforts to improve product quality and enhance user satisfaction.

Currently, sentiment classification of product reviews has become a hot research topic. By training on extensive product review data, models can automatically extract sentiment information from text and determine the polarity of sentiment. However, existing sentiment classification models still have limitations in accuracy when dealing with complex sentiments expressed in product reviews. Thus, the objective of this research is to design a deep learning-based sentiment classification model for product reviews to improve the accuracy and reliability of sentiment classification, which is essential for providing more precise and effective product quality control and customer service to merchants.

In recent years, sentiment analysis technology has become one of the hot research topics in natural language processing. With the development of deep learning algorithms, many researchers have explored the performance of different models using Chinese sentiment analysis corpora and achieved promising results in sentiment analysis tasks. For instance, Jiang Congjun, Wang Wendong, et al. improved the accuracy of Chinese sentiment analysis by expanding sentiment lexicons [1]. Mao Xuefen annotated corpora with sentiment labels, constructing Chinese corpora with sentiment annotations [2]. The expansion of sentiment lexicons or the annotation of corpora has significantly increased the accuracy of sentiment analysis.

Many researchers have also explored different model structures. Qi Mengna, Zhu Liping, et al. proposed an online comment sentiment analysis model based on ERNIE and CNN [3]. Chen Shihao utilized a semi-supervised mechanism to build a novel training model, addressing issues related to the need for large amounts of high-quality labeled data, which are costly in terms of manual labor and time in traditional supervised learning methods [4]. Zhang Shunxiang, Yu Hongbin, et al. introduced an ELECTRA-based approach for sentiment classification of product review texts. This method employed the ELECTRA pre-trained model for feature extraction and classification, achieving excellent results on sentiment analysis datasets [5]. Li Xinhao proposed a research method for sentiment classification of product reviews based on word polarity algorithms and one-dimensional convolutional word vectors. This method used multiple approaches for feature extraction to improve text sentiment classification results [6]. These studies have all made significant contributions to the field of sentiment analysis, enhancing the efficiency of text sentiment analysis.

However, research on sentiment classification of large-scale short phrase texts like product reviews is still insufficient, leading to suboptimal accuracy in sentiment classification for product reviews. Therefore, it is essential to conduct research on sentiment classification models for this type of text data, utilizing training models based on Long Short-Term Memory (LSTM) to enhance the accuracy of sentiment classification for product reviews.

II. RELEVANT TECHNOLOGIES

Long Short-Term Memory (LSTM) network is a variant of recurrent neural network (RNN) commonly used for processing time-series data. Compared to traditional RNNs, LSTM introduces three gate mechanisms: the forget gate, the input gate, and the output gate, to better control the flow of information. Among these, the input gate regulates the amount of data entering the network, the retention of historical information is controlled by the forget gate, and the output gate is responsible for outputting the current time step's state, while the cell state is used for storing and passing historical information. These gates are computed and controlled through a series of neural network layers to model and predict input sequences. LSTM offers several advantages over traditional RNNs:

Solving the vanishing gradient or exploding gradient problem: In traditional RNNs, when the number of input time steps is large, it can lead to vanishing or exploding gradients. LSTM addresses this issue effectively by introducing gate mechanisms that selectively retain or forget some information, thus avoiding vanishing or exploding gradients during training on long time-series data.

Long-term memory capabilities: Through the control of the input gate, forget gate, and output gate, LSTM can selectively retain or forget past information, giving it strong long-term memory capabilities. It can handle multiple time-series data, meaning it can process inputs and produce outputs for multiple time series.

Handling various relationships in sequences: The gate mechanisms in LSTM allow for selective retention or forgetting of different types of information, making it capable of handling various relationships within sequences, including temporal, spatial, and semantic relationships. LSTM's structure is illustrated in Figure 1.

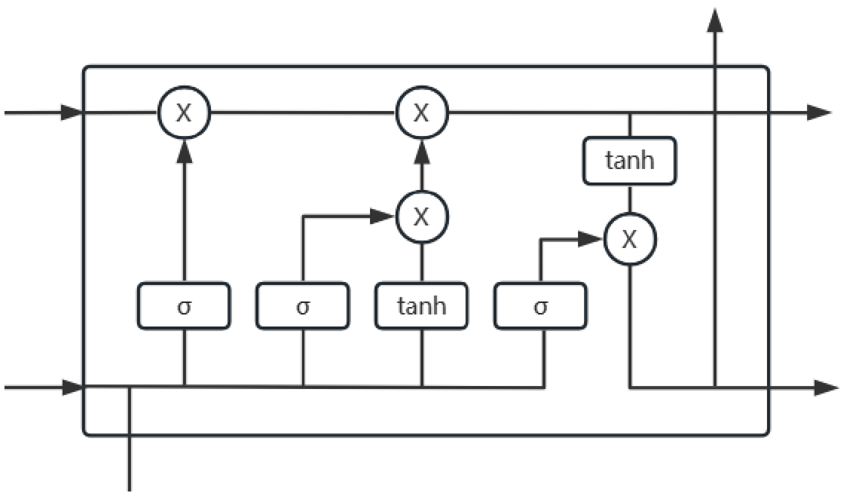

Figure1: LSTM Structure Diagram

In this diagram, the black lines represent input and output connections between nodes, circular symbols denote pointwise computations, and rectangular boxes represent neural network layers.

III. GENERAL DSEIGN

3.1 General Design Chart

The overall design diagram of this design is shown in Figure 2.

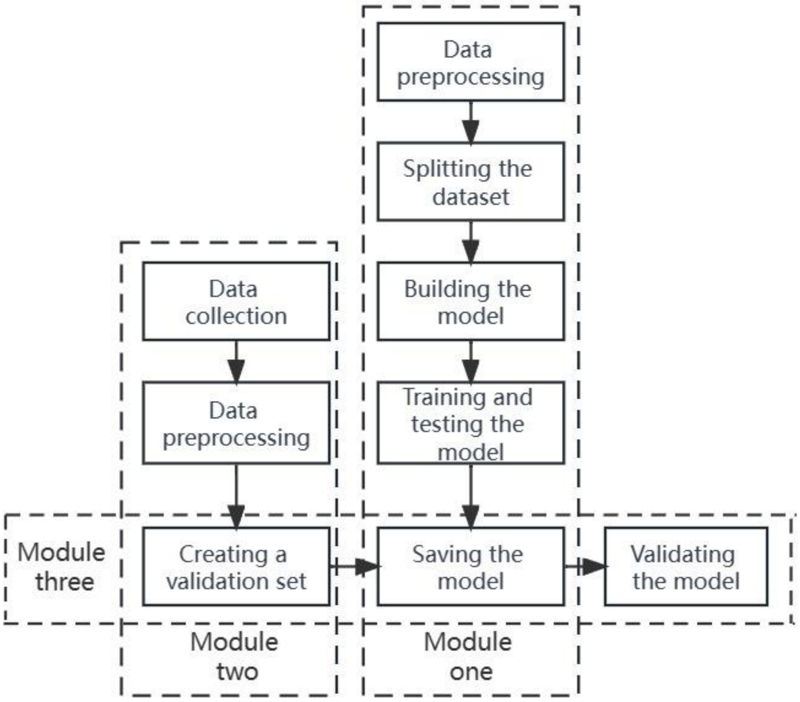

Figure2: Overall Design Framework

The overall design of this project is divided into three modules. Module one involves training and testing models, saving the best-performing model for subsequent validation of product reviews. Module two is responsible for creating a validation dataset through data collection and preprocessing. Module three involves using the validation set to verify the model saved in module one.

Modular design simplifies the complexity of the system by breaking down large models into smaller modules, making the system's structure clearer and easier to understand. Each module has its specific functions and responsibilities, making it easier to comprehend and maintain the entire system. Modular design also facilitates testing and debugging since breaking the model into modules allows for individual testing and debugging of each module. This enables quicker identification and resolution of issues, ultimately improving the quality and stability of the model.

3.2 Model Design Flowchart

The flowchart of the product review sentiment classification model design is shown in Figure 3.

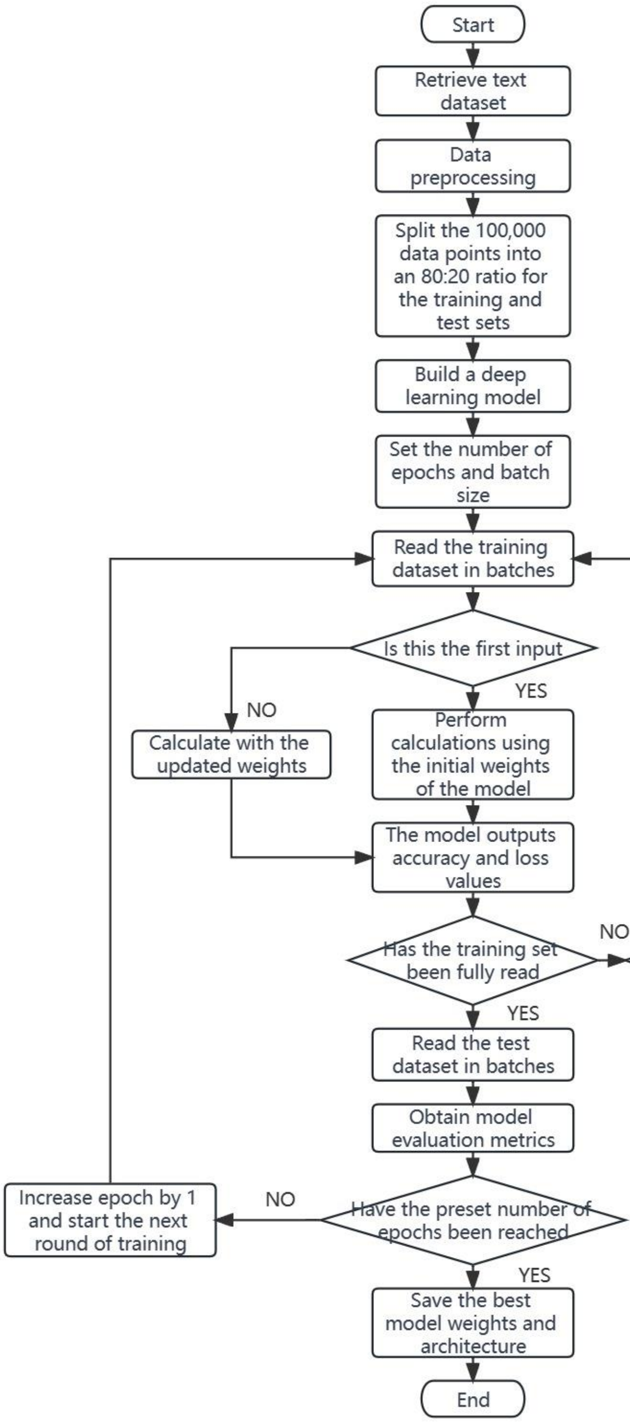

Figure3: Model Design Flowchart

The workflow for designing the product review sentiment classification model is as follows: Firstly, read the text dataset for training, perform preprocessing, construct a deep learning sentiment classification model based on LSTM, and finally, train and test the model.

3.3 Model Design and Analysis Based on Deep Learning

When using the LSTM model, it has the capability to automatically distinguish between useful and irrelevant information, storing the useful information. LSTM units take input data x(t) and produce hidden layer output h(t). Compared to traditional RNN networks, the hidden layer representation h(t) of LSTM is more complex, comprising three components: the input gate, output gate, and forget gate. These components facilitate the effective processing of input data, forgetting unnecessary information when needed, and retaining important information. These three gate structures are illustrated in Figure 4.

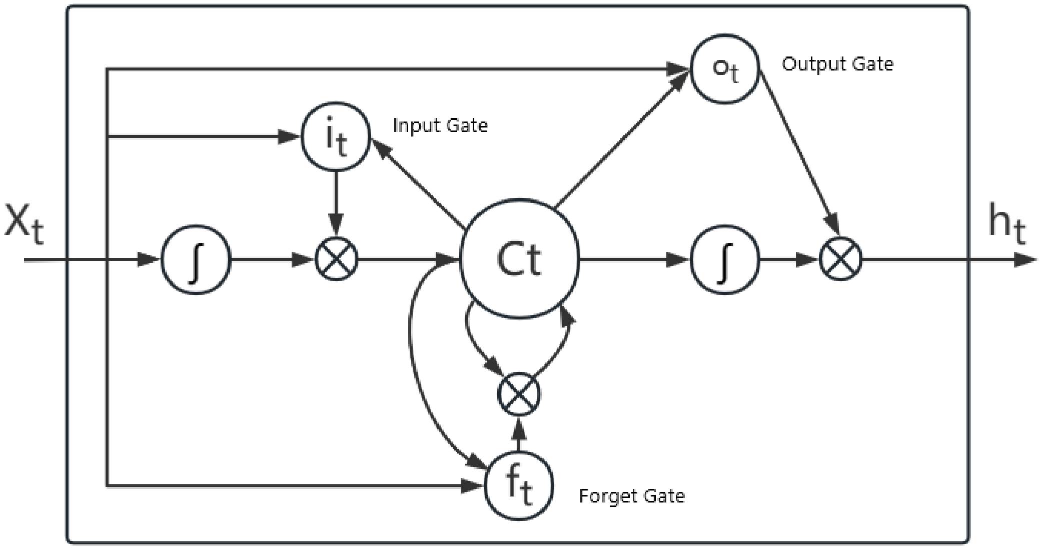

Figure4: Structure of Input Gate, Output Gate, and Forget Gate

Each gate utilizes the input data Xt to compute certain intermediate states, passing these states into different channels that ultimately converge into the output state ht. These gates can be seen as modules with distinct functions. The control parameter Ct is used to determine which values should be retained and which should be discarded. The input gate determines the weights applied to each input, while the forget gate decides which information should be discarded. The output gate determines the final output ht based on the intermediate states.

The task of the forget gate is to receive long-term memory and decide what to retain and what to forget. The calculation formula for the forget factor is shown in equation (1).

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

The weight matrix Wf for the forget gate, which is used to control the behavior of the forget gate. [ht-1, xt] represents the concatenation of the previous hidden state ht-1 and the current input vector xt into a longer vector. bf is the bias term for the forget gate, and σ represents the sigmoid function.

The primary task of the input gate is to determine which information should be stored in the cell state (unit state). The input gate consists of two parts: a sigmoid layer and a tanh layer. The sigmoid layer is used to decide which values to update, while the tanh layer is used to generate the candidate input, which is the new candidate cell state. The calculation of the cell state ct at the current time step is as follows: first, the previous cell state ct-1 is multiplied by the forget gate ft, then the current input state ct is multiplied by the input gate it, and finally, these two products are added together. This combines the current memory ct with the long-term memory ct-1, forming the updated cell state ct. Through the control of the forget gate, the LSTM model can retain information from a long time ago. The control of the input gate prevents currently irrelevant content from entering memory. This mechanism allows LSTM to flexibly handle and retain important information. The formulas for the input gate are shown in equations (2), (3), and (4).

$$i_t = \sigma\ (W_t \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh\ (W_C \cdot [h_{t-1}, x_t] + b_C) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

Here, Wt represents the weight matrix of the input gate, and bi is the bias term for the input gate. Wc represents the weight matrix of the current input unit state, and bc is the bias term for the current input unit state.

The output gate determines the output values based on the cell state as follows: first, we use a sigmoid function to determine which parts of the cell state should be output. Then, we process the cell state through a tanh layer to scale its values between [-1, 1]. Finally, we multiply the processed cell state by the output of the sigmoid function to obtain the information ht that we want to output. The formulas for the output gate are shown in equations (5) and (6).

$$o_t = \sigma \left(W_o[h_{t-1}, x_t] + b_o\right) \tag{5}$$

$$h_t = o_t * \tanh \left(C_t\right) \tag{6}$$

Wo represents the weight matrix of the output gate, and bo is the bias term for the output gate.

IV. EXPERIMENTAL RESULTS

4.1 Dataset Preprocessing

This paper utilizes the "weibo_senti_100k" dataset, which comes with sentiment labels for each text data. The dataset has already been divided into positive and negative sentiments, where "1" represents positive comments, and "0" represents negative comments. In total, there are 100,000 text data samples. After training and testing, the dataset is split into a training set and a testing set in an 80:20 ratio, comprising 80,000 data samples for training and 20,000 for testing. This division ensures efficient testing while maintaining a substantial volume of data for model training, ultimately ensuring a high level of accuracy.

Data preprocessing plays a crucial role in data analysis and deep learning. It refers to the process of cleaning, transforming, and organizing raw data before applying specific algorithms or models. The primary goals of data preprocessing are to enhance data quality, usability, and interpretability, enabling better support for subsequent analysis and modeling tasks. Data preprocessing helps eliminate issues and noise in the data, improving data quality and usability, resulting in more accurate, reliable, and efficient subsequent analysis and modeling work.

Preprocessing mainly includes tasks such as tokenization, handling of stop words, and removal of meaningless words, breaking a sentence into meaningful phrases, and preparing the data for subsequent model training and testing.

4.2 Model Parameter Configuration

After preprocessing the experimental textual data, the model can undergo training. During the specific training process, the textual data is divided into a training set and a test set, with the training set accounting for 80% of the total data and the test set accounting for 20% of the total data. The key parameters of the model are shown in Table 1.

Table 1: Primary Model Parameters

Parameters	Parameter Value
Batch_size	200
Epoch	8
Loss	BCELoss
Optimizer	Adam
Learning rate	0.005
Dropout rate	0.5

4.3 Experimental results and analysis

After debugging, this project's model was trained with the following parameters: the number of training epochs was set to 8, and the batch size was set to 200. The model.save function was used to save both the trained model structure and the model weights.

Because the data was split into a training set and a test set at the beginning of model training, the training set was used to train the model, while the test set was used to evaluate the model's performance. This approach helps prevent the model from overfitting to the training data and enhances its ability to generalize to unknown data. Model evaluation often involves metrics such as accuracy, loss, recall, and F1-score.

Accuracy can be calculated by dividing the number of correctly predicted samples by the total number of samples in the test set. In this experiment, 80,000 samples were used as the training set, and 20,000 samples were used as the test set. The model was trained for a total of eight epochs. Table 2 presents the results of various metrics used to evaluate the model after each training epoch.

Table 2: Model Evaluation Metrics Results

Epoch	Loss	Accuracy	Recall	F1-score
1	0.214	93.595%	93.590%	0.9360
2	0.209	94.060%	94.063%	0.9406
3	0.204	94.125%	94.132%	0.9412
4	0.198	94.604%	94.598%	0.9461

5	0.186	94.812%	94.810%	0.9480
6	0.183	94.867%	94.856%	0.9487
7	0.179	94.890%	94.890%	0.9489
8	0.177	94.925%	94.924%	0.9493

Based on the evaluation metrics, it can be observed that after multiple rounds of training, the loss gradually decreases, and the accuracy gradually increases. This indicates that increasing the number of training epochs can effectively improve the accuracy of the trained model. The above results also demonstrate that the LSTM network model can perform well in text sentiment classification tasks.

V. CONCLUSION

Our research has been dedicated to tackling the intricacies of sentiment classification in the context of concise yet extensive text formats, notably product reviews, where traditional models may exhibit suboptimal performance. As a result, our primary emphasis has been on developing a neural network model grounded in deep learning to address the intricacies of sentiment classification within this particular text format. Through a series of experiments, we systematically fine-tuned the parameters of the LSTM network model and conducted multiple iterations of training on extensive datasets. Consequently, we achieved a notable enhancement in the accuracy of sentiment classification for product reviews.

ACKNOWLEDGEMENT

This work was supported by the Open Project of National Intelligent Society Governance Testing Area (ZNZL2023B07, ZNZL2023A04) and the Sichuan Science and Technology Program, Soft Science Project (No.2022JDR0076). We also would like to thank the sponsors of Industry-school Cooperative Education Project of Ministry of Education (No. 202002230010, 202101014067, 202101291013, 220500643270506)

REFERENCES

[1]. Jiang Congjun, Wang Wendong, Li Haiyan. Construction of Sentiment Lexicon for Yan'an Scenic Spot Reviews. Journal of Yan'an University (Natural Science Edition), 2023, 42(01): 74-78.

[2]. Mao Xuefen. Analysis of Consistency in Sentiment Annotation Based on Product Reviews: A Case Study of Automotive and Beauty Product Reviews. Educational Resources, 2018(35): 19-22.

[3]. Qi Mengna, Zhu Liping, Li Ning. Online Comment Sentiment Analysis Model Based on ERNIE and CNN. Computer Applications, 2022, 42(S1): 7-11.

[4]. Chen Shihao. Research and Implementation of a Product Review Sentiment Analysis System Based on Semi-Supervised Learning. Central University of Finance and Economics, 2022. DOI: 10.27660/d.cnki.gzczu.2022.002002.

[5]. Zhang Shunxiang, Yu Hongbin, Zhu Guangli. Sentiment Classification of Product Review Texts Based on ELECTRA. Journal of Anhui University of Science and Technology (Natural Science Edition), 2022, 42(01): 77-84.

[6]. Li Xinhao. Research on Sentiment Classification of Product Reviews Based on Word Polarity Algorithms and One-Dimensional Convolutional Word Vectors. Software, 2022, 43(12): 100-104.