A sentiment analysis method based on bidirectional long short-term memory networks

Haifei Zhang¹, Jian Xu², Liting Lei¹, Qiu Jianlin¹,², Riyad Alshalabi³

¹School of Computer and Information Engineering, Nantong Institute of Technology, Nantong 226002, Jiangsu, China
²School of Information Science and Technology, Nantong University, Nantong 226019, Jiangsu, China
³College of Administrative Sciences, Applied Science University-Bahrain, East Al-Ekir, Bahrain

Abstract

Although the traditional recurrent neural network (RNN) model can cover the time information of the whole sentence theoretically, the gradient is dominated by the short-term gradient, and the long-term gradient is very small, which makes it difficult for the model to learn the long-distance information, and thus the effect of RNN on long text sentence recognition is poor. The long short-term memory network (LSTM) introduces the gate mechanism, especially the forgetting gate, which improves the disappearance of the gradient of RNN. Therefore, LSTM can store long text information and remove or increase the ability of information interaction by adding gate structure, which has natural advantages for long text processing. Based on the word vector matrix of GloVe model, on the open-source comment sentiment140 data set, we use the TensorFlow framework to construct the LSTM neural network and divide the data into the training set and test set based on the ratio of 4:1, design and implement the sentiment analysis published by Twitter users based on LSTM model, and then propose the bidirectional LSTM (Bi-LSTM) sentiment analysis method. The experimental results show that the accuracy of bidirectional LSTM is higher than that of unidirectional LSTM in sentiment analysis.

Keywords: Sentiment analysis, deep learning, Bi-LSTM
AMS 2010 code: 68T50.

1 Introduction

With the rapid development of the economy, the connection between the Internet and people’s daily life has become closer. In this highly developed Internet era, people are no longer just passively accepting information from the outside world, but more and more people are playing the role of information makers. More
and more Internet users, regardless of their age, are keen to express their opinions on the online interactive platform.

Sentiment analysis shows that the positive and negative sentiments reflect the views of Internet users on people, events and goods. It has high value for users as Internet platforms and governments explore the sentiment of a large number of comments. In recent years, with the development of deep learning technology, great achievements have been made in the field of natural language processing.

As the gradient of the RNN model is mainly controlled by short-term gradient and the long-distance gradient is very small, it becomes difficult for the model to learn long-term information. Therefore, the effect of the RNN model on long text sentence recognition is poor. This paper proposes a text sentiment analysis method that utilises the Bi-LSTM network and is based on the LSTM model. Experiments show that the method based on the Bi-LSTM network has higher accuracy in text sentiment analysis than the method based on the LSTM network.

2 Related work

In the past, machine based learning has attracted the attention of many scholars. The obtained text features are mapped into multi-dimensional feature vectors and sent to the model for training to learn the text feature information.

Machine learning mainly includes unsupervised methods and supervised methods [1]. Sentiment classification methods based on supervised learning methods mainly include support vector machine (SVM), naïve Bayes, k-nearest neighbour (KNN), support vector machine (SVM) and maximum entropy [2]. Pang et al. first introduced the machine learning method into sentiment analysis, studied the data of 2000 film reviews, classified the experimental text by naïve Bayes and SVM, and judged the sentiment of the text [3]. In 2008, Pang et al. used the CBOW model to analyse sentiment in order to continue to improve accuracy. Subsequently, many researchers researched this basis to improve the model. Liu Zhiming et al. [4] used a variety of calculation methods such as feature item weight and feature value extraction in the sentiment tendency analysis task, and combined these with the original machine learning algorithm; and through the sentiment analysis of microblog text, the final experiment shows that combining SVM with information gain (IG) and term frequency inverse document frequency (TF-IDF) feature extraction method lead to higher accuracy. Lin Shiping and others have also achieved good results by integrating multiple features into the support vector machine model. Li Tingting et al. [5] extracted many Chinese text features in Chinese sentiment classification and combined them with support vector machine. Cao Yu et al. [6] expanded the existing diversified sentiment database, combined with expanded sentiment dictionary, special symbols, negative words and so on, and achieved good results in the texts of microblog comments. The methods described above can also identify the sentiment tendency of the text, but their accuracy is not high, manual annotation is required when the amount of data is large and the effect is not very ideal. In recent years, in order to solve the problem of sentiment analysis, researchers began to use algorithms based on deep learning, which has yielded good results and has been widely recognised as being effective.

In recent years, deep learning has been increasingly applied in data analysis. Many experts and scholars began to use the method based on deep learning, and they adopt better model algorithms to analyse the sentiment of the obtained text. It has been applied to sentiment analysis and achieved good results. Du Changshun et al. [7] adopted the dropout method to prevent the overfitting phenomenon of the model during training, improve the accuracy of the model and use the segmented pooling strategy to extract the main features of sentences. The final results show that both the dropout algorithm and the segmented pooling strategy algorithm are helpful in the performance of model classification. Wang [8] added the attention mechanism to the LSTM network, which will have more human brain thinking and pay more attention to some important specific goals during training, so as to make more effective sentiment judgement from all aspects. Cai Huiping et al. [9] proposed a sentiment classification model based on word embedding and convolutional neural network (CNN). Based on this model,
it is found that sentiment text analysis has been greatly improved compared with traditional machine learning. Mesnil et al. [10] proposed a language model to distinguish the positive and negative aspects of sentiment.

At present, CNN [11] and RNN [12] are used more in sentiment analysis tasks. However, in terms of effect, CNN cannot effectively extract the contextual semantic information of long text, but RNN can capture context semantics. RNN is a temporal deep network for sequence modelling, which can apply the previously stored content to the current semantics; thus, it has obvious advantages over the spatial deep network CNN. For the problem of long text, the LSTM model can effectively solve the problem of long-term dependence of RNN in the training process through its long-term and short-term memory units. LSTM is improved based on RNN, which solves the problems of gradient disappearance and gradient explosion. A large number of experimental results show that the performance of the LSTM model is better than that of RNN. In order to be more accurate, sometimes it is bad to predict that this restaurant is dirty, ‘No’ is a modification of dirty, which needs to be determined by several inputs above and several inputs later, therefore, a Bi-directional recurrent neural network (Bi-directional RNN) based method is proposed, which can scan along with two directions [13]. Based on bi-directional RNN, this paper proposes a bidirectional LSTM method that is better used for emotion analysis.

3 Related models and methods

3.1 GloVe model

The model used in this experiment is the pre-trained GloVe model, a word vector trained based on 6 million elements of data, which is a word vector expression with functions similar to late semantic analysis (LSA) and Word2vec [13]. GloVe model characterises word vectors as real numeric vectors to vectorise words, so as to contain as much and accurate semantic and grammatical information between vectors as possible. The core of GloVe model is to construct word embedding matrix and process word vectors for each word in the obtained text. The dimension of this model is 300 dimensions, and the semantic similarity of text is often expressed by the similarity of vector space. Each word corresponds to a word vector in this model. If the words in the text are semantically similar, then the distance in the word vector is similar.

Glove, LSA and Word2vec are common methods to obtain embedded matrices. LSA is an early word vector representation tool. The dimension of large matrix is reduced by LSA based on singular value decomposition (SVD). However, due to the high complexity of SVD, its calculation process consumes time and is not friendly to computers with poor performance. In recent years, the Word2vec model is also used more in the process of deep learning, but since it contains 3 million word vectors, the word vector matrix is too large. Moreover, Word2vec does not use global co-occurrence, that is, it does not make full use of all corpora. Word2vec contains two structures, CBOW and skip-gram. CBOW model lacks the relationship between words in the whole sentence due to the direct addition of word vectors in the context, resulting in the lack of word relationship information. The skip-gram model is directly trained. Since this algorithm uses intermediate words to predict adjacent words, it is easy to get too much weight for high-frequency words. More importantly, the two models update the word vector with the information in one window at each training, but Glove is based on the global corpus (collinear matrix), that is, multiple windows, and therefore, the speed of model training is accelerated. Therefore, a more manageable matrix glove model completed by pre-training is used for training in this paper.

3.2 Text vectorisation representation

As a kind of unsupervised learning, deep learning does not need manual annotation in front of a large number of data elements. Its emotional classification of data is through its learning and model training. Therefore, it is only necessary to train the word vector after data pre-processing.

Data set pre-processing consists of the following sequence of tasks: input text, delete special symbols (punctuation marks, brackets) other than English words in the text and convert uppercase letters to lowercase. Since the computer cannot directly understand the text content, it is necessary to quantify the text into a numerical
form so that the computer can understand through machine learning. Each word in the text can be represented by a vector, and then a word vector matrix is obtained through Glove model.

The word embedding word vector adopted in this experiment is proposed by Pennington. In order to solve the problem of sparse vector, this method adopts distributed mapping of words from high latitude space to low dimensional space. There are two commonly used models: CBOW and Skip-gram. Intuitively, Skip-gram predicts the context given the input word. CBOW is used to ascertain the given context to predict an input word, as indicated in Figure 1.

3.3 Dropout method

In the process of deep learning, due to the need to train a large amount of text data, an overfitting phenomenon is often observed. Generally speaking, for example, after training the characteristics of a Persian cat with a large amount of data, the result is a cat. When taking the civet cat as a test, there is too much difference between the civet cat and the Persian cat due to the existence of overfitting, and thus the result says that the civet cat is not a cat. The over-fitted model has no practical value. In the experiment, in order to solve this problem, the integration method is usually used, that is, multiple models are combined to improve the accuracy. However, it is very difficult to train an algorithm to follow this method; it is also very time-consuming to test multiple models, and, additionally, doing so will result in excessive resource-consumption.

The core of the improved idea of neural network, proposed by Pennington [13] in his early years, is that in the training process of neural network model, a neural network unit is irregularly removed or lost from the neural network. It is worth noting that this discarding is only temporary, and thus some neurons in the model can be independent of other neurons, which greatly reduces and weakens the synergistic fitting and generalisation between neuronal features.

Spatial dropout is a dropout method proposed by Tompson et al. Through experiments, it can be found that the dropout method is irregular and will randomly set some elements of the matrix to zero, while the spatial dropout will randomly set some regions to zero. The spatial dropout method in the Keras module is also very effective in solving the overfitting problem.
4 Sentiment analysis based on LSTM

4.1 Fundamentals

LSTM is a variant of RNN, and it is a higher-level RNN. With the strong context connection ability of a memory unit in a long text, RNN refers to a sequence whose current output is related to the output before this time. The specific manifestation is that the network will remember the previous information and apply it to the calculation of the current output, that is, the nodes between the hidden layers are no longer in a state of non-connection, but are connected, and the hidden layer input includes not only the input layer output but also the hidden layer output at the previous time. The RNN model is shown in Figure 2.

As can be seen from Figure 1, the value of $s$ depends on two aspects: (1) input vector $x$ and (2) the value of the hidden layer at the previous time. Eqs. (1) and (2) represent the calculation process of the recurrent neural network:

$$O_t = g(v \cdot S_t)$$
$$S_t = f(u \cdot X_t + w \cdot S_{t-1})$$

Where: $O_t$ represents the final output result, $v$ and $u$ are the weight matrixes respectively, representing the weight matrix from hidden layer to output layer and from input layer to hidden layer. $S_t$ represents the value of the hidden layer, $X_t$ represents the input value at the current time, and $w$ is the weight matrix, representing the last $S_t$ as the weight input this time.

RNN has some defects in the semantic understanding of long text, and thus this paper uses the LSTM model. The LSTM network adds a long-term state based on the RNN model. The hidden state of neurons is also called cell state, which is often represented by the symbol ‘C’. The core of LSTM is these gate structures. The concept of the gate in cell state is introduced below.

There are three gates in LSTM. These gates can effectively deal with some problems of gradient explosion and gradient disappearance caused by the RNN. Forget gate is used to delete excessive information in the cell at the previous time; the input gate is used to determine which information should be added to the cell as input at this time; the output gate is used to determine the output of information stored in the cell at the current time. Figures 3 and 4 describe the LSTM model and gate structure.

The calculation formula for each LSTM cell is as follows:

Forget gate: for each new input, the LSTM will decide which memories to be forgotten according to the
currently input and previously output results. Through the sigmoid neural layer, the input can be compressed to the (0, 1) interval.

\[ f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \quad (3) \]

\[ \tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, X_t] + b_c) \quad (4) \]
Input gate: the neural layer that calculates the output value at the current time. The integration vector uses
the sigmoid function to extract the output layer, calculates the output of the previous time and the current time,
and then maps the interval through the $tanh$ activation function.

$$\hat{I}_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \quad (5)$$

Output gate: it determines what to output based on the content saved in the cell state. Similar to the input
gate, the output of the output gate to the content also needs to be determined by using the sigmoid activation
function, and then the $tanh$ activation function is used to process the content of the cell state. Finally, multiplying
the two results is the final result that we need.

$$O_t = \sigma(W_0 \cdot [h_{t-1}, X_t] + b_0) \quad (6)$$

Two kinds of memory:
Long memory: to obtain the status value of the memory unit at the current time.

$$C_t = f_t * C_{t-1} + \hat{I}_t * \tilde{C}_t \quad (7)$$

Short memory: output of LSTM cell.

$$h_t = O_t * tanh(C_t) \quad (8)$$

Where $h_t$ represents the output result of the current unit and $C_t$ represents the state of the memory unit at the
previous time.

4.2 LSTM-attention text feature extraction

The inspiration for this method comes from humans themselves: when our vision perceives the scene in front
of us, we don’t see everything in a scene every time, but only the thing we want to see. That is when we learn
that in a particular scene, the thing we want to see almost always appears in a certain part. When we learn of
this characteristic through repeated exposure to similar scenes, we would spontaneously train ourselves to focus
on the particular focused part only and try not to look at other parts, the objective being to improve efficiency in
work.

The hierarchical attention network is shown in Figure 5.

This paper uses the attention mechanism in a hierarchical attention network, puts the features extracted from
vectorised words and sentences into the network layer, so as to realise different degrees of attention to text
information, and uses the obtained feature vectors to realise text classification.

5 Sentiment analysis based on Bi-LSTM

5.1 Bi-LSTM

The LSTM algorithm has two methods: forward propagation and backward propagation. In order to make
the model prediction more accurate, the LSTM model is combined with the LSTM in the positive direction and
the LSTM in the reverse direction from the input layer, and then used as the next input, as indicated in Figure 6.

The calculation formula is as indicated in Eqs. (9)–(11):

$$h_t = f(w_1x_t + w_2h_{t-1}) \quad (9)$$

$$h'_t = f(w_3x_t + w_5h'_{t+1}) \quad (10)$$

$$O_t = g(w_4h_t + w_6h'_t) \quad (11)$$
5.2 Bi-LSTM-based sentiment analysis

In the field of sentiment classification, the CNN method has also achieved good results, but CNN involves the need to build a lot of feature engineering for improving the accuracy of sentiment classification, which will take a lot of time. When using the LSTM method, this problem is avoided, and there is no need to consider the semantic relationship between words. For a long text, LSTM can effectively connect the semantics of the context and learn sentence-level text features. Bi-LSTM can simply be understood as LSTM in two directions. The results obtained before and after can be combined to solve the problem of sentiment classification, and the effect is better in more complex sentences. In the research of sentiment classification, first, the user’s comments are vectorised. LSTM neural network will selectively retain the information affecting the neural network through the unique gate structure, and update the cell state in real-time. For example, when predicting the text published by ‘In my point of view, The two singers performed badly, But listen more deeply and find that it’s not as bad as that. It’s intriguing’, the LSTM network will turn the previous dissatisfaction with the singer’s attitude into satisfaction through the forgetting gate of its cell unit. The analysis result of user comment text finally uses the sigmoid function to display the output result as 1 (positive) or 0 (negative).
6 Result analysis

6.1 Experimental data

This experiment uses the data set sentiment140 (user comments provided by Twitter), which has 800,000 positive and 800,000 negative sentiment data elements. The model is trained and tested based on this data set. The obtained data sets were randomly divided into 1,280,000 training sets and 320,000 test sets based on the ratio of 4:1. Each data element is a user’s English text comment. The model is obtained from the training set, and the accuracy is obtained from the test set.

6.2 Experimental process

1. First, we process the data set, by resetting the header ‘sentiment’ and ‘text’ for the data set and discarding useless columns, as shown in Figure 7.

2. For the processing of text, the downloaded English stop thesaurus and the extraction of English stem (for example, present participle of English will become the backbone vocabulary) are used. The regularised expression is also used to process the special symbols in a text. In the process of data cleaning, the uppercase letters are also converted into lowercase letters for processing. The processing of one piece of data is shown in Figure 8.

3. We divide the data set into training set and test set according to a certain proportion. ‘English’ itself is a word. We index each word and fix the maximum length of the text for training. We use the pre-trained word vector GloVe to represent the words with feature vectors to obtain the word embedding matrix that can be recognised by the computer. As a specific operation, we set the downloaded GloVe model as a word vector of 300 dimensions. In this way, words with similar meanings will have similar vector representation. The result of word vector representation in words is shown in Figure 9.

4. We build the model, add the dropout method, add convolution layer, add optimiser and set parameters to train the data.
5. We write the test function, input the test text, compare the experimental results on the trained model and select the optimal deep learning model.

6.3 Experimental results (Table 1)

The results of text sentiment analysis method based on LSTM are shown in Figure 10. The results of text sentiment analysis method based on Bi-LSTM are shown in Figure 11. In the field of sentiment classification, the CNN method has also achieved good results, but CNN involves the
need to build a lot of feature engineering in order to improve the accuracy of sentiment classification, which will consume a lot of time. When using the LSTM method, this problem is avoided, and there is no need to consider the semantic relationship between words. For long text, LSTM can effectively connect the semantics of the context and learn sentence-level text features. Bi-LSTM can simply be understood as LSTM in two directions. The results obtained before and after can be combined to solve the problem of sentiment classification, and the effect is better in more complex sentences. In the research of text sentiment classification, first, the user’s comments are vectorised. LSTM neural network will selectively retain the information affecting the neural network through the unique gate structure, and update the cell state in real-time. For example, when predicting the text published by the user ‘in my point of view, the two singers performed badly, but listen more deeply and find that it’s not as bad as that. It’s intriguing’, the LSTM network will turn the previous dissatisfaction with the singer’s attitude into satisfaction through the forgetting gate of its cell unit. The analysis result of user comment text finally uses the sigmoid function to display the output result as 1 (positive) or 0 (negative).

Fig. 10 Experimental results of LSTM sentiment analysis.

Fig. 11 Experimental results of Bi-LSTM sentiment analysis.
6.4 Experimental analysis

The experimental results show that with the increase of model training times, the accuracy of the model increases steadily and the loss rate decreases gradually, as shown in Figure 12. The accuracy of test set and training set is also increasing, and the loss rate is decreasing. In the sentiment analysis of user comments, when the same text is input, both can accurately predict whether the text encompasses a positive or negative sentiment. However, when looking at the more accurate probability value of the model, the prediction accuracy of the Bi-LSTM-based method is up to about 76.8%, and the prediction accuracy of the LSTM-based method is up to about 75.4%. In some texts, the prediction accuracy of the Bi-LSTM-based method is much higher than that of the LSTM-based method. Therefore, the Bi-LSTM-based method has a higher prediction accuracy of sentiment analysis than the LSTM-based method in dataset Sentiment140.

7 Conclusions

According to the experimental results of this paper, there will be overfitting between LSTM model and Bi-LSTM in the process of training. Overfitting often occurs in deep learning, and thus the dropout method is applied to the LSTM model and the spatial dropout method is applied to the Bi-LSTM model to alleviate the problem of overfitting by preventing the synergy of neuronal features using fixed or random methods. Finally, the experiment is trained based on 1.6 million user comments. The experimental results show that the accuracy of the Bi-LSTM model is about 77% after training 10,000 data elements on each occasion for 10 times, and the accuracy of the LSTM model is about 75.0% after training 10,000 data elements on each occasion for 10 times. The final experimental results show that the Bi-LSTM model has certain advantages over the single LSTM model in the sentiment140 data set. Of course, some texts can better reflect the advantages of Bi-LSTM, and the difference between the two results may be larger.

Acknowledgments

This work is sponsored by: (1) Training project of top scientific research talents of Nantong Institute of Technology under Grant No. XBJRC2021005; (2) the science and technology planning project of Nantong City under Grant No. J C2021132, MS22021028; and (3) the key project of college students’ innovation and entrepreneurship training programme of Jiangsu Province in 2021 under Grant No. 202112056003Z.

References

Sentiment analysis using bidirectional long short-term memory networks


