Review of Artificial Intelligence
ISSN: 2789-5505 (print)
Shuangqing Academic Publishing House
http://www.qingpress.com

Classification of Short Texts in Weibo Based on BERT Model

Chunyu Jiao
Central China Normal University, China

Abstract

With the gradual development of machine learning and deep learning techniques, scholars have made significant progress in classifying texts using well-performing neural networks. When conducting more in-depth research on short texts, finegrained sentiment analysis is the key to solving the current problem. As a result, multi-categorization text analysis at the comment text level has shifted to aspect-level judgments on statements in comments. In this project, aspect-level sentiment analysis is improved for the short text of Weibo comments based on the BERT model combined with an attention mechanism. The model proposed in this paper uses the attention mechanism to analyze the weights of the short text and precisely weight the summation and finally the category output of the short text can be obtained. For the characteristics of Weibo comments, such as few words and short sentences. This paper achieves good results by improving the lexical model. The algorithm in this paper conducts comparative experiments on the Chinese data set of Weibo users' comments. The experimental results show that the algorithm in this paper is better at aspect-level-achieve higher classification accuracy and recall rate than other algorithms on aspect-level sentiment classification tasks.

Keywords: Bert, NLP, Short Texts

1. INTRODUCTION

In the 21st century, people's communication is very fast-paced, and people are more willing to express their moods at the moment through short messages. Therefore, the analysis of short texts becomes more and more important. Weibo comment data are studied using natural language processing techniques. The purpose of this paper is to further improve the progress of short textbook analysis module in the field of natural language processing by proposing a new deep neural network to process short textbook data, which is used to change the difficult problems of existing research on short textbook sentiment analysis and further achieve optimization to achieve more accurate short textbook sentiment analysis in practice. The starting point of this study is to count the large number of comments made on Weibo, analyze their sentiment tendencies, and thus mine the sentiment tendencies of the majority of user comments [1]. Sentiment analysis is generally classified according to the size of the processed text granularity and can be divided into three approaches based on document-level sentiment analysis, sentence-level sentiment analysis, and aspect-level sentiment analysis. The aspect-level sentiment analysis is based on the BERT model we will study in this paper. As one of the tasks of the sentiment analysis task, it is a more fine-grained sentiment analysis method compared with the other two sentiment analysis methods, which is aimed at identifying the sentiment polarity of different aspects in the comment text. Currently, sentiment analysis methods in aspect-level-based sentiment analysis tasks are divided into the following three main categories: lexicon and rule-based sentiment analysis methods, traditional machine learning-based sentiment analysis methods, and deep neural network-based sentiment analysis methods. This paper mainly uses the deep learning approach to build the model [2].

2789-5505/© Shuangqing Academic Publishing House Limited All rights reserved.
Article history: Received Jan 10, 2022 Accepted Jan 22, 2022 Available online Jan 24, 2022
To cite this paper: Cuiyu Jiao (2022). Classification of Short Texts in Weibo Based on BERT Model. Review of Artificial Intelligence, Vol2, Issue1, Pages 1-12.

2. RELATED WORK

2.1 SENTIMENT CLASSIFICATIONN

The traditional sentiment analysis is first to construct a sentiment dictionary or sentiment collocation template by the lexicon and rule-based sentiment analysis method, then use it as a benchmark to compare the sentiment words or fixed collocations contained in the review text and finally calculate the sentiment tendency of the text. Later scholars improved it by adopting a machine learning-based approach to classify texts, mainly by extracting features and modelling the training data set with polarity labels, then learning them by machine learning algorithms to finally achieve the judgment of sentiment polarity. The former provides the feature set needed by the classification model to fit the data to the training text, while the latter affects the classification model's ability to the latter affects the ability of classification model to learn the feature set. This paper adopts a deep learning approach for text sentiment analysis. Deep neural networks can better capture the dependencies between aspects and contextual content, learn deep features, and break the limitations of machine learning. The application of deep learning methods in aspect-level sentiment analysis tasks consists of two main parts: the neural network language model and the neural network model to extract deep features [3]. A distributed representation approach is proposed, first to map words to a continuous low-dimensional vector space using a neural network language model, then calculate the cosine similarity between each word to capture the correlation between individual words. The word vectors are obtained by constructing a co-occurrence matrix of word context combination training, which considers local information and utilizes global corpus information. The target words are used as input features, stitched together with contextual features, and then input to the classification layer for sentiment polarity analysis. Meanwhile, in the sentiment analysis task, different words in the context are associated with the target word to different degrees, so the weights of different words, even different sentences, need to be calculated and analyzed using the attention mechanism. Deep learning-based sentiment analysis algorithms can be better applied to big data, and large-scale neural network models are highly capable of understanding the deep semantics of text. So a large number of researchers expect to use deep learning methods to build neural network models to automatically extract potential semantic features of text and apply them directly to the sentiment classification task to achieve a better. The goal is to achieve better classification results without human involvement [4].

2.2. ATTENTION MECHANISM NETWORKS

$$C_j = \sum_{i=1}^{T} a_{ij} h_i$$

The attention mechanism is the attention to the input weight assignment. The first use of the attention mechanism is in the encoder-decoder, where the attention mechanism obtains the input variables of the next layer by weighting the hidden states of the encoder over all time steps: the general expression of the attention mechanism can be written as O=soft max(QK)VO=soft max(QK)V. The attention mechanism can be broadly understood as a one-layer perceptron (soft max summation) consisting of the query term matrix, the corresponding key term, and the value term to be weighted averaged [5]. The attention module in the network structure is responsible for automatically learning the attention weights a_{ij}, which automatically capture the correlation between h_i (the encoder hidden state, which we call the candidate state) and s_j (the decoder hidden state, which we call the query state). These attention weights are then used to construct the content vector C, which is passed to the decoder as input. At each decoding position j, the content vector c_j is a weighted sum of all hidden states of the encoder and their corresponding attention weights.

In the field of natural language processing, the current mainstream attention mechanism is mainly applied to the Seq2Seq problem. In the traditional RNN model structure, the input and output of the model are a fixed-length sequence. At the same time, in general application scenarios, such as machine translation, the length of the sentences after machine translation does not depend on the length of the original sentences, so the standard RNN model cannot solve such problems, which The Encoder-Decoder framework emerged to solve the Seq2Seq problem and the attention mechanism is often attached to this framework. The following figure shows the abstract representation of the Encoder-Decoder framework [6].

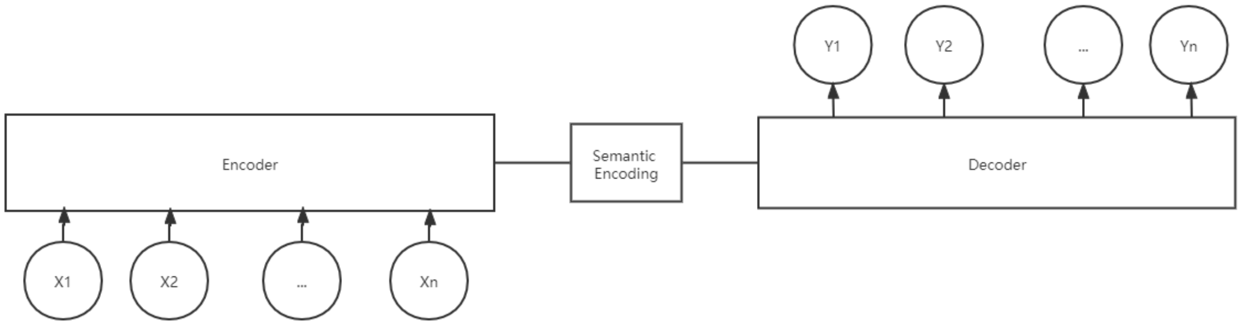

FIG. 1 Encode-Decoder framework diagram

It can be seen from the figure that the intermediate semantic representation C_i of the Encoder-Decoder framework with the introduction of the attention model is different, so the process of generating the target word sequence can be expressed by the following equation.

In Encoder, by encoding the input sequence $\{x_1,x_2,...,x_n\}$, the encoding models such as RNN,LSTM, etc. Models are used to generate the contextual encoding vector c and then Decoder is used to decode the semantic encoding vector c. The structure of the common decoding model is similar to the encoding model and the output sequence $\{y_1,y_2,...,y_k\}$ is finally generated y_k.

3. SHORT TEXT FEATURE EXTRACTION ALGORITHM

3.1 DEPENDENCY RELATIONS

The purpose of dependency analysis is to obtain the most likely set of dependencies in a sentence. We construct a series of dependency syntax rules based on the constructed inter-word dependencies to further process these

$$y_1 = f(C_1)$$

$$y_2 = f(C_2, y_1)$$

$$y_3 = f(C_3, y_1, y_2)$$

dependencies. For example, when the core predicate of a sentence is a verb, the object, aspect and entity of evaluation maybe in the subject position, object position and complement position of the object and the lexical nature of the subject of evaluation is usually a noun phrase, verb phrase and the emotional phrase is usually found in the rhetorical. Therefore[7]we construct a dependency syntax rule to extract the phraseological features of the text according to these evaluation objects and the possible positions of the sentiment phrases, the following is the calculation of the dependency probability in the specific dependency syntax rule. In the same sentence of the training set, the probability of a dependency relationship between w_i and w_j is

$$P_c(l_{ij}) = \frac{C_c(w_i, w_j, R)}{C_c(w_i, w_j)}$$

where $C_c(w_i, w_j, R)$ is the number of times that there is a dependency relationship between w and R in the same sentence in the training set C. $C_c(w_i, w_j)$ is the number of times that there is a dependency between w_i and w_j in the same sentence in training set C, and C( ) is the number of times that both w_i and w_j occur in a sentence. The meaning of $P_D(l_{ij})$ is similar to that of $P_c(l_{ij})$, except that in the specific calculation, a single document is used instead of a document set. Single document instead of the document set. However due to the data sparsity problem, a smoothing operation must be performed, first in the calculation of $P_D(l_{ij})$ is calculated by using linear interpolation to smooth the individual documents by the document set C document D.

$$P_D{}'(l_{ij}) = (1-\lambda)P_D(l_{ij}) + \lambda P_C(l_{ij})$$

In the above equation, the λ is the interpolation coefficient. Then the final formula for the probability of dependence $P_D(l_{ij})$ and $P_C(l_{ij})$ is as follows.

$$\boldsymbol{P = \lambda_1 E_1 + (1 - \lambda_1)(\lambda_2 E_{23} + (1 - \lambda_2) E_4)}$$

Where λ_1 and λ_2 are the smoothing parameters, defined as follows.

$$\lambda_1 = \frac{\delta_1}{\delta_1 + 1}, \lambda_2 = \frac{\delta_2 + \delta 3}{\delta_2 + \delta 3 + 1}$$

3.2 TEXT FEATURE EXTRACTION BASED ON WORD CO-OCCURRENCE

Phrase feature extraction based on dependent syntactic rules reflects the text's semantic information by extracting the text's phrase features. Although this method considers the semantic connection between words in the text, it requires additional dependent syntactic analysis tools to perform dependent syntactic analysis on the text. In addition, the

$$d_{co}(w_i, w_j) = \frac{N_{doc}(w_1, w_2)}{N_{doc}(w_1)}$$

establishment of dependent rules requires human participation, thus greatly reducing the efficiency of the text feature extraction process [8]. In addition, short texts, due to their characteristics, such as short word count, colloquialism, and omitted sentences, phrase feature extraction methods based on dependent syntactic rules cannot effectively solve the problem of missing short texts. The absence of short texts has a great impact on their semantic understanding. In this section, we propose a word co-occurrence feature extraction method based on semantic similarity weights to solve the above problems. First, we construct a word co-occurrence matrix based on the words that appear in the data set under a certain topic, such as the Phone data set, after text pre-processing and based on word frequency; second, we introduce the Word2Vec word vector model and add semantic similarity weights to the word co-occurrence matrix. Finally, we select the most relevant words for each word w_i in the word co-occurrence matrix with semantic similarity weights w_i' and $< w_i, w_i >'$ is used as the feature of the text [9]. The following figure shows the experimental steps of the word co-occurrence feature extraction method based on semantic similarity weights. Text is an ordered combination of several words and the word co-occurrence matrix is mainly used to describe the relevance between words. When the probability of two words appearing in the same document at the same time is high, it can be considered that the stronger the relevance of these two words, so we mainly use word frequency to measure the degree of association between words in the document. Where $d_{co}(w_i, w_j)$ denotes the co-occurrence similarity between words w_i and w_j, $N_{doc}(w_i, w_j)$ denotes the number of words w_i and w_j occurring simultaneously in the documents in the data set, and $N_{doc}(w_i)$ denotes the number of words w_i occurring in the documents. In this paper, we convert the text data into a text-word matrix by dividing the data set into words and deactivating words. Then count the number n of all words appearing in the text-word matrix to construct an $n \times n$ word-word matrix and calculate the co-occurrence similarity between each two words in the word-word matrix using the above formula.

3.3 Calculation of Semantic Similarity Weights

The maximum log-likelihood function is used as the objective function in the Word2Vec model, as shown in the following equation.

$$max(\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log P(w_{t+j}|w_t))$$

Where c is the words with upper and lower windows between 2, T is the length of the text, and in addition, $p(w_{t+j} \mid w_t)$ is calculated using the softmax function as shown in the following equation.

$$p(w_O|w_I) = \frac{exp({v_{w_o}}'^T v_{w_I})}{\sum_{w=1}^{W} exp({v_{w_o}}'^T v_{w_I})}$$

The Word2Vec language model maps all the words in the text into 300 dimensional real vectors, which can be converted into vectors to quantitatively analyze the semantic connections between words and the similarity between words can be calculated. For example, the word 'king' + 'woman'= 'queen', so we use the cosine similarity to calculate the semantic connection between two words, as shown in the following formula.

$$d_{semantic}(w_i, w_j) = \frac{w_i \cdot w_j}{\|w_i\| \cdot \|w_j\|}$$

We use the semantic similarity between words $d_{semantic}(w_i, w_j)$ as the weight along with the co-occurrence similarity $d_{co}(w_i, w_j)$ as the evaluation function to measure the degree of association between words and based on this evaluation function as the weight Weight(w_{ij}) in the word co-occurrence matrix. The following equation is shown.

$$Weight(w_{ij}) = d_s emantic(w_i, w_j) \times d_{co}(w_i, w_j)$$

Based on the word co-occurrence matrix of the constructed data set, for a word w_i in the text, we select the word w_j with the highest weight w_{ij} among all candidate words associated with the word in the word co-occurrence matrix and use w_i, w_j together as the word co-occurrence feature of the text.

3.4 SENTIMENT ANALYSIS EXPERIMENT

In order to verify the effectiveness of the text feature extraction algorithm based on word co-occurrence on the extraction of text implicit semantic features and the influence of different feature extraction algorithms on text sentiment classification, we conducted sentiment classification experiments to compare the text feature extraction based on word co-occurrence and the phrase feature extraction algorithm based on dependent syntactic rules. In addition, we also conducted sentiment classification experiments for word features, lexical features. Then analyzed and verified the effect of the feature sets constructed by different text feature extraction algorithms by performing sentiment classification experiments, in which the classification model used the LSTM model. The experimental evaluation indexes in this paper mainly include Accuracy, Precision, Recall and F1-score etc. The experimental results are evaluated by combining several indexes. Accuracy is the ratio of correctly classified sentiment categories to all sentiment categories. Accuracy rate refers to the ratio of correctly classified in all positive samples. The recall rate refers to the ratio of all correctly classified sentiment categories that are actually positive. The F1 value is a combined indicator of the accuracy rate and recall rate. The specific evaluation metric parameters are shown in the table below.

TABLE 1 COMPARISON EXPERIMENTAL RESULTS OF DIFFERENT FEATURE EXTRACTION ALGORITHMS FOR SENTIMENT CLASSIFICATION

Method	Accuracy	Precision	Recall	F1 values
Word features	0.7136	0.7048	0.6975	0.7258
Word features +lexical features	0.7396	0.7396	0.7145	0.7305
Word co-occurrence	0.7436	0.7401	0.6996	0.7396
Word co-occurrence + lexical features	0.7498	0.7706	0.7683	0.7351

The above table shows the comparison results of different feature extraction algorithms for sentiment classification. We can see that the text feature extraction algorithm based on word co-occurrence has the best effect on sentiment classification, with an accuracy of 74.95% in the data set. The latter strongly relies on the syntactic structure of the text itself. Still, short texts have the characteristics of non-standard grammar and colloquialism, so the effect may be poor in the face of short texts with irregular expressions. In contrast, the text feature extraction algorithm based on word co-occurrence can supplement the implicit features of the text through the word co-occurrence matrix to solve the problem of missing information in short texts. In addition, the combination of word features + word features and word co-occurrence features + word features slightly improves the performance than word features and word co-occurrence features alone due to the addition of lexical features, which also indicates that the lexical features of the text can reflect the sentiment tendency of words to a certain extent and thus help to predict the sentiment tendency of the text.

In addition to the sentiment classification comparison experiments with different feature extraction algorithms, the following table shows the results of the sentiment classification comparison experiments based on different neural network models. This also shows that for text sequences, the grammatical structure of the text itself is conducive to mining the semantic features of the text, while the sequence model can better preserve the sequence features of the text and characterize the semantics of the text based on the position of words in the text. The sequence model can better preserve the sequence features and characterize the text semantics based on the position of words in the text. In addition, the LSTM network and GRU network outperform the RNN network in the sequence model, which is also due to the design of the gating circuit in the LSTM network and GRU network that can mine the semantic features more relevant to the sentiment analysis. The LSTM network has the best effect on sentiment classification, which also fully illustrates the effectiveness of the LSTM network in characterizing the potential semantic information of the text.

TABLE 2 EXPERIMENTAL RESULTS OF EMOTION CLASSIFICATION BY DIFFERENT NEURAL NETWORK MODELS

Method	Accuracy	Precision	Recall	F1 values
Word co-occurrence features+CNN	0.6838	0.6997	0.6699	0.7034
Word co-occurrence features+RNN	0.7043	0.7063	0.7101	0.7039
Word co-occurrence features+GRU	0.7345	0.7066	0.7016	0.7016
Word co-occurrence features +LSTM	0.7409	0.7340	0.7390	0.7350

4. ASPECT-LEVEL SENTIMENT ANALYSIS BASED ON THE ASPECT-ATTENTION MEMORY NETWORK MODEL

4.1 CONSTRUCTION OF ASPECT-ATTENTION MEMORY NETWORK

This chapter designs an automatic classification model for genre based on sequential features to address these difficulties. The model incorporates the advantages of several basic deep learning models and the overall structure can be divided into an input layer, a feature extraction layer, a fully connected layer and an output layer. The model structure is shown in the figure below. First the model uses the BERT model in the input layer to obtain the sentence vector representation. The BERT model learns the deep semantic features in the pre-training stage, so the BERT pre-training model can obtain the sentence vector with deep semantic information[10]. Then in the feature extraction part of the model, as we learned from the theoretical introduction in the previous chapter, common deep learning models such as LSTM can capture the long-term dependencies of the utterances and effectively process the sequence information of the text, to better extract the overall features of the text. The Aspect-Attention memory network model can extract local text information as a complement to features, while the Attention mechanism enables the model to retain key information in a focused manner while ensuring comprehensiveness. Therefore, LSTM is used as the basis for the feature extraction layer. Then the extracted features are integrated in the fully connected layer part and a fully connected operation is performed. Finally, for the classifier part, this model directly uses the Softmax function directly connected at the model's output.

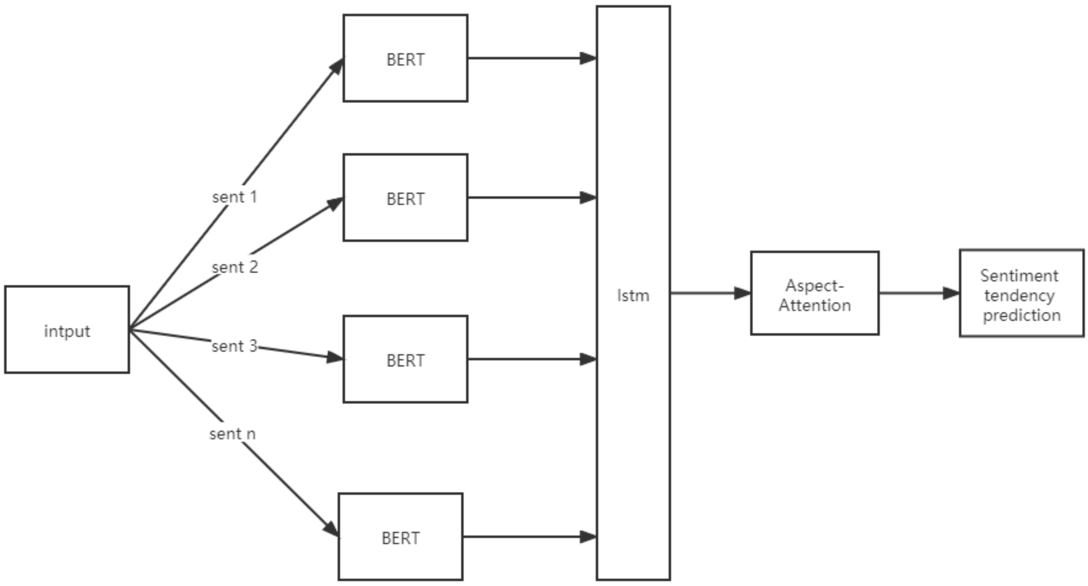

FIG. 2 ASPECT-ATTENTION MEMORY NETWORK

The structure and functions of each layer are described in detail below.

1. Input layer

The purpose of the input layer is to pre-process the text and perform sentence vectorization representation. The text is first broken and then input to the BERT model to obtain the sentence vector representation as input to the feature extraction layer. The specific process of sentence vector representation has been described in the previous subsection, so we will not repeat it here. Suppose the input short text is $Q_m=(W_1, W_2, ... ,W_m)$ and after breaking the sentence for Q_m , assume that (E_1, E_2, E_n) denotes a sentence after the short text is broken, the BERT pre-training model vectorizes the sentence and after the Transformer model analyzes the semantic structure and the relationship between words in the sentence from multiple perspectives, the sentence representation vector $[T_1, T_2, ..., T_n]$.

2.Feature extraction layer

The purpose of the feature extraction layer is to extract the overall features and local features of the short text by performing feature extraction on the sentence vector representation output from the input layer.

Since the short text studied in this chapter has the feature of contextual logical relevance, we choose a bi-directional long and short-term memory network (LSTM) to learn this kind of sequence information. LSTM is composed of a LSTM model. The first layer of the LSTM network is to fuse the information within the sentence and the second layer of the LSTM network is to fuse the information of the context, inputting each sentence into the forward LSTM and backward LSTM. Using the hidden layer state of the last time step as the feature representation of the text to form a feature representation with contextual logical relevance. Suppose the forward LSTM network obtains the hidden layer vectors $[h_{L1}, h_{L2}, h_{L3}, ..., h_{Lm}]$, and the backward LSTM network obtains the hidden layer vector $[h_{R1}, h_{R2}, h_{R3}, ..., h_{Rm}]$. Finally, the forward and backward hidden layer vectors are stitched to obtain a text vector representation with inter sentence logical sequence information [hL1, hR1], [hL2, hR2], ..., $[h_{Lm}, h_{Rm}]$. This enables the feature fusion of the information of the upper and lower connected sentences and effectively avoids the loss of contextual information. Then the representation vector of each utterance is filtered using the attention mechanism, which aims to emphasize the different influences of different input data on the output data. Based on the idea of Attention, the attention mechanism will increase the weight of important statements in the text and enhance the attention of that part, thus optimizing the feature vector of the text and improving the classification performance. For example, in the comment text, the beginning and the end are likely to be the key summary sentences of the text.

So the feature extraction should pay attention to the beginning and the end parts of the comment in addition to the logical relationship of the context. The computation of Attention mechanism mainly consists of two parts of the computation process: the computation process of the attention probability distribution, and the other is the final feature computation process. The input at each moment is the output generated by each sentence through the LSTM model. In the model of this chapter, the output attention probability at n moments of time, is calculated by the following equation. Where N denotes the number of input sequence elements. U is the weight matrix, and F represents the summation of the final hidden layer state values in each independent direction in the LSTM[11].

$$a_n = \frac{exp(h_n{}')}{\sum_{i=1}^{N} exp(h_i{}')}$$

$$h_n{}' = h_n \cdot^T UF$$

The summation of the hidden layer state values at time n is calculated as shown below.

$$y = softmax(F_i{}') = \frac{exp(F_i{}')}{\sum_{j=1}^{T}} exp(F_j{}')$$

$$F' = V \cdot F$$

In the above equation, W_n represents the input data of the model at the moment of t_n and h_{n-1} represents the state of the hidden layer of the model at the previous moment of n_t. U and V represent the weight parameter matrices of the model's input layer and hidden layer, respectively. In this model, the final feature F_a, based on the attention distribution, is calculated as shown in the following equation.Where T is the number of category labels and V denotes the weight matrix of the model output layer. F_i' denotes the component value in vector F' and the vector length is equal to the number of classification labels. After the soft max function classification, the probability distribution of the text category can be obtained, and the cross entropy loss is obtained with the real category distribution Y, as shown in the following equation.

$$E(Y, y) = -Y \log(y)$$

Where Y denotes the probability distribution of the true category and y denotes the probability distribution of the category predicted by the model.

4.2 Analysis of results

The following table shows the Accuracy, Precision, Recall and F1-score of the aspect-level sentiment analysis experiments of different algorithms in the data set. Then see the table that the LSTM without the attention mechanism has the worst effect on the aspect-level sentiment analysis, indicating the attention mechanism. This also indicates that the attention mechanism can tap the aspect information of the text to increase the effectiveness of aspect-level sentiment classification. It is worth noting that ATAE-LSTM is stronger than TD-LSTM and IAN in the data set, where ATAE-LSTM is an improved version of TD-LSTM and the difference is that the former involves the randomly generated aspect vectors in learning the semantic features of text context, so ATAE-LSTM can highlight the semantic parts of text context words related to aspect information, while IAN can highlight the semantic parts of text context words related to aspect information. While IAN adopts a different granularity of attention mechanism, it still aims to explore the semantic importance of the words in the text but ignores the more important aspectual information. In addition, the Aspect-Attention memory network proposed in this paper takes advantage of the LSTM expertise in processing sequential data, adopts a bidirectional LSTM to learn co-occurrence features of text words and then characterize aspectual features of text. Finally, constructs an attention mechanism between contextual semantic features and aspectual features to fully explore the potential connection between contextual semantics of text and aspectual features of text. This also makes the experimental results of aspect-level sentiment analysis by the Aspect-Attention memory network better than the other algorithms in the comparison experiments. In addition, the data above shows that the F1 values of the models designed in this chapter are higher than those of the LSTM, TD-LSTM, ATAE-LSTM, and IAN models[12].

TABLE 3 ASPECT-LEVEL SENTIMENT ANALYSIS RESULTS OF DIFFERENT ALGORITHMS

Method	**Accuracy**	**Precision**	**Recall**	**F1 values**
LSTM	0.7231	0.7043	0.7095	0.7231
TD-LSTM	0.7195	0.7240	0.7360	0.7003
ATAE-LSTM	0.7246	0.7389	0.7435	0.7209
IAN	0.7431	0.7640	0.7499	0.7589
Aspect-Attention Memory Network	0.7838	0.8195	0.7735	0.7505

In general, compared with the commonly used LSTM models, the innovations of the algorithm model in this chapter. Firstly in the text representation, the conventional word vector representation is improved to sentence vector representation, which makes the text representation vector of short text rich in semantic information. Secondly, in the feature extraction, the Attention layer is added based on the LSTM network structure, which makes the model more capable of mining. Then in terms of feature extraction, the Attention layer is added to the LSTM network structure, which makes the model more capable of extracting key information in sentences, and then combined with the aspect layer, the model extracts local information. The final experimental results prove that the algorithm can effectively improve the effect of short text textual sentiment classification.

5. FULL ARTICLE SUMMARY

With the rapid development of social media, people are sharing their views and expressing their opinions on the Internet, and a large amount of user data is generated from the Internet every day. Unlike traditional blog posts, news, and other long texts, these user-generated short text data are short, colloquial and the content is mostly commented on a specific event and a specific product. Sentiment analysis of these short texts can help users make purchase decisions, and companies can also use user feedback to understand what users like and dislike about their products. In short text sentiment analysis, chapter-level and sentence-level sentiment analysis have achieved good results. However, there are still some difficulties in aspect-level sentiment analysis, mainly because of the short text content, semantic understanding difficulties and aspect information extraction. Traditional machine learning-based sentiment analysis methods cannot extract the deep semantic meaning of short texts and further explore the aspectual features of texts. In contrast, deep learning-based methods can effectively characterize the semantic meaning of texts and do not require a human feature extraction process. The attention mechanism enables the neural network model to focus on the more relevant words to the target function when understanding the semantic meaning of the text or the aspectual information. In addition, since the attention mechanism enables neural network models to focus on those words and sentences that are more relevant to the target function when understanding text semantically, it is important to design more effective aspect-level sentiment analysis models that ignore the order relationship between words when generating word vector representations and single-layer attention can only obtain the weight assignments of different words in a sequence, ignoring the weight assignments of different sentences, resulting in the loss of deep sentiment features. The analysis model is a very important work at present. Therefore this paper proposes a short text sentiment analysis algorithm with an improved attention mechanism based on the BERT model and its main work is as follows: (1)A text feature extraction method based on word co-occurrence is proposed to solve the problem of missing content in short texts, by constructing a word co-occurrence matrix for the whole review data set and introducing the Word2Vec word vector model adding semantic. We also introduce the Word2Vec word vector model and add semantic similarity weights to the word co-occurrence matrix to highlight the semantically linked words but less frequently co-occurring in the short text. For each word in the text, the most relevant candidate word is selected from the word co-occurrence matrix based on the semantic similarity weights and used as the word co-occurrence feature of the text together with the word. The results show that word co-occurrence features can achieve better sentiment classification results than other features. (2)An aspect-level sentiment analysis algorithm based on the Aspect-Attention memory network model is proposed to address the difficulty of extracting textual aspect information. The final text semantic output is formed by weighted summation with the text context semantics and aspect features. Since the Aspect-Attention mechanism can extract the semantic features related to the textual aspect information in the textual context semantics, it improves the effect of aspect-level sentiment classification. (3)Since the Aspect-Attention mechanism cannot effectively characterize the sentiment tendency between the contextual semantics of text and aspectual features. We propose a tensor neural network based on multi-feature fusion, introduce lexical features, and build a tensor neural network for different text features, including lexical features, contextual semantic features, and aspectual features to explore the potential semantic connections between different features. We use the tensor decomposition technique to reduce the redundant information of tensor weights and the number of parameters of the model. The experimental results show that the tensor neural network based on multi-feature fusion can effectively improve the accuracy of aspect-level sentiment classification.

ACKNOWLEDGMENT

Thanks to Central China Normal University.

REFERENCES

[1] Ge Bin, He Chunhui, Zhang Chong, and Hu Yanli. Classification algorithmof chinese sentimentorientation based on dictionary and lstm. In Proceedings of the 2nd International Conference on Big Data Research, ICBDR 2018, page 119–126, New York, NY, USA, 2018.Association for Computing Machinery. ISBN 9781450364768. doi: 10. 1145/3291801.3291835. URL https://doi.org/10.1145/3291801.3291835.

[2] Jie Gao. Chinese sentiment classification model based on pre-trained bert. In *2021 2nd International Conference on Computers, Information Processing and Advanced Education*, CIPAE 2021, page 1296–1300, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450389969. doi: 10.1145/3456887.3457511. URL https:// doi.org/10.1145/3456887.3457511.

[3] Xiaoqing Gu, Kaijian Xia, Yizhang Jiang, and Alireza Jolfaei. Multi-task fuzzy clustering–based multi-task tsk fuzzy system for text sentiment classification. ACM Trans. Asian Low-Resour. Lang. Inf. Process., 21(2), nov 2021. ISSN 2375-4699.doi: 10.1145/3476103. URL https://doi.org/ 10.1145/3476103

[4] Yong Huang and Siwei Liu. An efficient model for text sentiment analysis. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, ACAI 2019, page 479–484, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450372619. doi: 10.1145/3377713.3377796. URL https:// doi.org/10.1145/3377713.3377796.

[5] Jilei Lin, Yipei Huang, Ying Gao, and Rongjuan Chen. Predicting information popularity: A study of sina weibo. In *Proceedings* of the *2017 2nd International Conference on Communication and Information Systems*, ICCIS 2017, page

335–339, NewYork, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450353489. doi: 10. 1145/3158233.3159327. URL https://doi.org/10.1145/3158233.3159327.

[6] Jingang Liu, Chunhe Xia, Xiaojian Li, Haihua Yan, and Tengteng Liu. A bert-based ensemble model for Chinese news topic prediction. In *Proceedings of the 2020 2nd International Conference on Big Data Engineering*, BDE 2020, page 18–23, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450377225. doi: 10. 1145/3404512.3404524. URL https://doi.org/10.1145/3404512.3404524.

[7] Kurland, Philip B., and Ralph Lerner, eds. The Founders' Constitution. Chicago: University of Chicago Press, Lattimore, Richmond, trans. The Iliad of Homer. Chicago: University of Chicago Press, Na Pang, Hairong Lu, and Li Qian. The entity analysis of social networks in weibo with suicidal tendencies based on bert. In The *2021 3rd International Conference on Big Data Engineering*, BDE 2021, page 125–130, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450389426. doi: 10.1145/3468920.3468938.URL https://doi.org/10.1145/3468920.3468938.

[8] Suresh Venkatasubramanian, Ashok Veilumuthu, Avanthi Krishnamurthy, Veni Madhavan C.E, Kaushik Nath, and Sunil Arvindam. A nonsyntactic approach for text sentiment classification with stop words. In Proceedings of the *20th International Conference Companion on World Wide Web*, WWW '11, page 137–138, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450306379. doi:10.1145/1963192.1963262.URL https://doi.org/10.1145/1963192.1963262.

[9] Nan Xiang, Qian qian Jia, and Yue dong Wang. Sentiment Analysis of Chinese Weibo Combining BERT Model and Hawkes Process, page 59–65. *Association for Computing Machinery*, New York, NY, USA, 2021. ISBN 9781450390163. URL https://doi.org/10.1145/3480001.3480007.

[10] Lyle Luo Yanlong, Philip Lei Iat Seng, and Rita Tse Tan Sim. Evaluating macao's gaming industry using sentiment analysis on weibo tweets. In Proceedings of the *2015 2nd International Conference on Electronic Governance and Open Society: Challenges in Eurasia*, EGOSE '15, page 139–144, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450340700.doi: 10.1145/2846012.2846018.URL https://doi.org/10.1145/2846012.2846018.

[11] Zhengchun Zhang and Qingsong Yu. Chinese relation extraction based on lattice network improved with bert model. In Proceedings of the 2020 5th International Conference on Mathematics and Artificial Intelligence, ICMAI 2020, page 98–102, New York, NY, USA, 2020a. Association for Computing Machinery. ISBN 9781450377072. doi: 10. 1145/3395260.3395276. URL https://doi.org/10.1145/3395260.3395276.

[12] Xiaotong Zhao. Book rating model based on self-attention and lstm. In *2020 6th International Conference on Robotics and Artificial Intelligence*, ICRAI 2020, page 153–156, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450388597.doi:10.1145/3449301.3449327. URLhttps://doi.org/10.1145/3449301.3449327.