Research Article

Application Analysis of Emotional Learning Model Based on Improved Text in Campus Review and Student Public Opinion Management

Zhaofeng Weng

Fuzhou University Zhicheng College, Fuzhou, Fujian 350002, China

Correspondence should be addressed to Zhaofeng Weng; 02107078@fdzcxy.edu.cn

Received 1 March 2022; Revised 18 March 2022; Accepted 26 March 2022; Published 23 April 2022

Academic Editor: Man Fai Leung

Copyright © 2022 Zhaofeng Weng. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In an era of rapid mobile Internet development, students are increasingly expressing their views on specific events through campus comments. In the Web 2.0 era, the concept of public media participation is widely used by students, and it is important to effectively analyze online campus public opinion comments and present the findings in a rationalized form for the management of campus public opinion. In the new era of education governance modernization, the level of public opinion management in universities has become a key indicator to improve the standard of education management in higher education institutions; therefore, this paper focuses on proposing an innovative deep learning-based research method for topic management of university public opinion. Firstly, through the improved LDA module with sentiment discrimination learning capability, the sentiment of the main arguments in the campus commentary is extracted, and then the statistical sentiment intensity of the in-depth learning module is used to analyze the sentiment intensity of the thematic arguments of different events in time series, so as to achieve the long-term tracking of the trend of the sentiment intensity of the whole event.

1. Introduction

Public opinion, as a sociological term, refers to the subjective perception of public events in a certain context. Public opinion on campus networks refers to the attitudes and possibly related behaviors of the "subjects of public opinion,” with university students being the specific group of people, toward the events they care about on the Internet [1]. As "subjects of public opinion,” university students are significantly different from other groups in expressing their views on specific events due to their age, psychology, and cognition [2]. According to the 48th Statistical Report on the Development of China’s Internet released by China Internet Network Information Center (CNNIC) in Beijing on August 27, 2021, the number of Internet users in China reached 1.011 billion in June 2021, an increase of 21.75 million compared with December 2020, and the Internet penetration rate was as high as 71.6% [3]. This huge scale includes the most active group of students in colleges and universities. In the context of promoting the modernization of educational governance, the management of online public opinion in colleges and universities has become an important part of improving educational management in colleges and universities [4]. College administrators must pay attention to the influence of information technology on students’ thoughts and education and strengthen public opinion management through scientific and innovative methods, and it is on this basis that this paper proposes an innovative and feasible method.

Currently, for opinion analysis and management, from an algorithmic perspective, the most focus is on LDA probabilistic topic models and deep learning models. The literature [5] used LDA models with word2vec models to construct sentiment dictionaries for the temporal analysis of sentiment intensity of hot events of students’ concern. In the literature [6], for public opinion comments in complex contexts, the sentiment value measurement algorithm was constructed and incorporated into the improved LDA-ARMA model for dynamic presentation and fine-grained classification of the sentiment of event opinion. In the topic
detection task in the opinion analysis subtask, literature [7] proposed to extend the topic distribution obtained from the LDA model with word vectors trained by Skip-gram of word2vec as the text representation and used it as the features of the SKM algorithm for cluster analysis, while optimizing the cluster number selection of SKM with the first-order difference of distance cost function values to obtain multiple text clusters as the online opinion topics. Literature [8] used a more efficient sentiment classification model based on this, which was constructed by LSTM neural network model, and the accuracy rate was improved by about 10%. Literature [9] used dimensional sentiment analysis based on Valence-Arousal, and the sentiment classification effect was significantly improved. Literature [10, 11] used a hybrid algorithm based on LSTM. LSTM + Bi further reduced the influence of irrelevant words in long texts on the classification results, while LSTM + Bert + Bi had higher accuracy and effectively solved the sentiment classification problem in public opinion. In the literature [12, 13], a logistic curve model was used to classify the stages of public opinion, and graph attention networks were used to predict the evolution of public opinion after sentiment classification. The literature [14, 15] used softsign activation function instead of tanh activation function in the LSTM model, while using regularized LSTM input weights to effectively combine the advantages of their respective models to classify the tendency of public opinion into positive and negative sentiments for early warning purposes, and the experiments proved that the performance of the enhanced model improved significantly and effectively.

The aforementioned literature applies deep learning algorithm models to various stages of opinion management from different perspectives, but from a general perspective, the following shortcomings still exist: (1) LDA models, as unsupervised machine learning techniques, adopt the bag-of-words approach to transform textual information into digital information that can be easily modeled, but the bag-of-words approach does not consider the order between words and there are difficulties in evaluating the effect; (2) word2vec, as a static approach, has strong generality but cannot be dynamically optimized for a specific task, and the problem of multiple meaning words cannot be effectively solved because the generating vector and the words are in a one-to-one relationship. Based on the above considerations, this paper improves the existing IDA model and deep learning algorithm based on the existing LDA probabilistic topic model and deep learning model from the perspectives of analysis strategy, dynamic monitoring, and effect evaluation; then combines various hybrid algorithms; and applies them to the opinion management of campus reviews. Finally, the effectiveness of the algorithmic model for campus opinion management is demonstrated with experimental validation analysis.

2. Related Work

As the essence of the research method proposed and applied in this paper is to use LDA models with deep learning models (word2vec, Bert), combined with sentiment dictionaries, to perform a temporal analysis of the intensity of sentiment in campus reviews, this section will focus on three parts: probabilistic topic models, neural network language models, and sentiment analysis based on sentiment dictionaries.

2.1. Probabilistic Topic Modeling. In probabilistic topic modeling, as one of the most common models in topic modeling methods, the core is to extract topic information from a large amount of textual information through statistical methods and theories, and it has a wide range of applications in the field of information retrieval [16]. The most initial text representation models, the TF-IDF model [16] and the spatial vector model [17], are simpler and cannot distinguish between polysemantic words. Latent Semantic Analysis (LSA) models are therefore proposed to achieve a high-dimensional to low-dimensional spatial mapping of document words [18]. Latent Dirichlet Allocation (LDA) is a more effective probabilistic topic model for identifying latent topic information in campus reviews, using a three-layer Bayesian framework, compared to the previous models [19]. However, as the traditional LDA model uses a bag-of-words approach that lacks consideration of interword order and text-to-topic links, the improved LDA model is used to extract and analyze thematic information from the text in order to optimize the results and improve the accuracy of sentiment analysis in campus reviews.

2.2. Neural Network Language Models. Statistical language models aim to learn the joint probability functions of words, and their main challenge is dimensional catastrophe. The emergence of neural network language models (NNLMs) has effectively addressed this problem. The first neural network language model (NNLM) [20] was systematized by Bengio. By optimizing the training model, it is possible to eliminate dimensional disasters and to understand a number of sentences with similar meanings. Two core components make up the neural network language model: distributed representation and word embedding [21]. Neural network models have evolved considerably over the last decade or so, and various models have been proposed using the NNLM as a template, including the CBOW model and the Skip-gram model [22, 23], which are simpler than the NNLM. In addition, hierarchical Softmax algorithms and negative sample algorithms have been developed to train models faster and more efficiently [24]. Google’s deep learning tool, word2vec, released in 2013, is a combination of these models and algorithms and provides a fast and effective way to obtain semantic associations between words. Based on the effectiveness of word2vec and its ability to accurately capture potential semantic similarities between words, the aim of this paper is to use an improved word2vec and Bert model to calculate the sentiment intensity of topic ideas extracted from campus reviews.

2.3. Emotional Analysis. Sentiment analysis is mainly implemented based on corpus, lexicon, and graph approaches. The corpus-based evaluation word extraction and
discrimination utilize the statistical properties of the corpus to mine and judge the polarity of corpus words, but the accuracy is more difficult to assess due to the limitations of the corpus. Lexicon-based evaluation word extraction and discrimination are mainly achieved through lexical links between words in the lexicon, and the HowNet sentiment lexicon is usually used for semantic similarity and semantic tendency of words based on semantic correlation fields. The calculation method, word-by-word matching for each comment text, finds all positive sentiment words and negative sentiment words in the text for sentiment calculation, and the sentiment scoring rule for each sentiment word is shown in the following equation specifically.

\[ S = (-1)^n V_s W. \]  

The sentiment score of the sentiment word in the time period is obtained by adding the sentiment scores of the same sentiment word appearing in different comment texts in the same time period, where \( S \) is the sentiment score of the sentiment word; \( n \) is the number of negative words; \( V_s \) represents the sentiment lexicality, with a value of 1 indicating a positive sentiment word and \(-1\) indicating a negative sentiment word; and \( W \) is the degree weight value.

Graph-based sentiment analysis is implemented by introducing an attention mechanism, which can solve the problem of information loss and accuracy degradation caused by the increase of coding intermediate vectors as the input text length increases when methods such as LSTM and word2vec are used for text classification. In this paper, a self-attentive mechanism is used to reduce the dependence on external information and improve the ability to capture data features, which in turn improves the accuracy of text de-emphasis analysis. The method is implemented by acquiring sentiment words for a certain time period of comment text and associating the comment text with the sentiment words to construct a graph structure, as shown in Figure 1.

The vertices in Figure 1 are the sentiment words that appear in that time step, and if there is a sentiment word \( w \) in the comment text, the comment \( s \) is attributed to the corresponding vertex \( v \). A vertex can contain multiple comments, and a comment can be assigned to multiple vertices. The mapping relationship also implies that the more the public comment texts associated with two sentiment words, the closer the sentiment expressed. In this paper, we propose a method for constructing edges between vertices based on the union of text structure and text content. The TF-IDF text similarity of the comment text associated with vertex \( v_i \) and vertex \( v_j \) is \( T_{ij} \). If there are public comments on the associated comment text and the number of public comments is \( N \), then an edge \( e_{ij} \) is added between these two vertices with the weight \( w_{ij} \):

\[ w_{ij} = \frac{a + N}{b + N}, \]  

where \( a \) and \( b \) are the numerator and denominator, respectively, after conversion to the simplest fraction. The

\[ (1) \]

\[ (2) \]
threshold value of text similarity \(a\) is set to 0.3. If there is no common comment between two vertices but their text similarity \(T_{ij}\) is greater than or equal to \(a\), an edge \(e_{ij}\) is similarly added between \(v_i\) and \(v_j\) with the weight

\[
w_{ij} = T_{ij},
\]

2.4. Gated Circulation Units. GRU is the same improvement to recurrent neural networks as LSTM. The GRU model contains two gate structures, update gates and reset gates. The reset gate determines how new input information is combined with previous memory, and the update gate defines the amount of previous memory saved to the current time step. By having a more streamlined structure, the GRU model has a speedup in training compared to the LSTM model and is able to better characterize and model the text. Let the input be \(x_t\) and the output of the GRU hidden layer be \(h_t\) at moment \(t\). The computation is shown as follows.

\[
z_t = \sigma(W_z[h_{t-1}, x_t]),
\]

\[
r_t = \sigma(W_r[h_{t-1}, x_t]),
\]

\[
\gamma = \tanh(W[r_t \ast h_{t-1}, s_t]),
\]

\[
h_t = 1 - z_t \ast h_{t-1} + z_t \ast \gamma,
\]

where \(w\) is the weight matrix connecting the two layers, subscripts \(r\) and \(z\) denote the reset and update gates, and \(\sigma\) and \(\tanh\) are activation functions.

The model in this paper uses a two-way gate to control the cyclic unit for data extraction, which fully considers the sequence information of the text and improves the accuracy of sentiment judgment.

3. Research Methodology and Framework

To effectively study the changing trends of sentiment intensity of topical events in campus comments, this study combines Latent Dirichlet Allocation (LDA) and deep learning. Considering the need for high timeliness in managing public opinion in campus comments, we design a sentiment intensity analysis framework based on probabilistic topic modeling and deep learning. The framework uses LDA models to discover information related to a specific topic event from campus comments; annotates the information by topic; mines students’ views and opinions on the event; implements sentiment analysis through word2vec word embedding and LSTM networks, so as to obtain the sentiment intensity of the information and apply it to the topic distribution; and finally builds up the sentiment intensity through changing it under each topic by attention networks and gated recurrent units combined to form a predictive model based on graph neural networks.

3.1. Potential Dirichlet Allocation Model. Probabilistic topic models (PTMs) have been used for good effect in text classification, information retrieval, and other related fields. The basic principle of a probabilistic topic model is that a document is a mixture of probability distributions of several topics, each of which is a mixture of probability distributions of words, and can be regarded as a generative model of a document. Among the various approaches to probabilistic topics, the Latent Dirichlet Allocation (LDA) model is one of the most effective [16].

The traditional LDA-based topic modeling relies on a word cooccurrence model, which is necessarily less effective for short texts such as reviews. Therefore, this paper will improve the IDA to enable adaptive and short textual adaptation of the topic model.

3.1.1. Model Generation Process. The LDA is a generative probabilistic model of a 3-layer Bayesian network with the graphical model shown in Figure 2.

Each campus review is assumed to be a mixture of topics; each topic is a probability distribution over a set of words, and can be regarded as a generative model of a document. Among the various approaches to probabilistic topics, the Latent Dirichlet Allocation (LDA) model is one of the most effective [16].

The traditional LDA-based topic modeling relies on a word cooccurrence model, which is necessarily less effective for short texts such as reviews. Therefore, this paper will improve the IDA to enable adaptive and short textual adaptation of the topic model.

The generation process is as follows:

1. Sample a word distribution \(\Phi_z \sim \text{Dir}(\beta)\) for any topic \(z\).
2. For each comment \(d_m\), sample a topic distribution \(\theta_d \sim \text{Dir}(\alpha)\).
3. For the word \(w_{mi}\) in comment \(d_m\), traverse a, b.

   (a) Select a subject \(z_{mj}, z_{mj} \sim \text{Multi}(\theta_d)\).
   (b) Choose a word \(w_{mi}, w_{mi} \sim \text{Multi}(\Phi_k)\).
\( M \) denotes the number of documents \( d \) in the corpus; \( z \) denotes the topic in the corpus; \( w \) denotes the words in the corpus; \( \theta_k \) denotes the word distribution of topic \( z_k \); \( \theta_d \) denotes the topic distribution of document \( d \); \( w_m, i \) denotes the \( i \)-th word of document \( d_m \); \( z_m, j \) denotes the \( j \)-th topic associated with document \( d_m \); \( N_d \) denotes the number of words in document \( d \); \( z_d \) denotes the topic of document \( d \); and \( \alpha \) and \( \beta \) are hyperparameters, empirical values: \( \alpha = 50/k, \beta = 0.01 \).

In the topic model, \( w_m(d) \) is the known variable, \( \alpha \) and \( \beta \) are the two prior parameters of the given Dirichlet distribution, and \( z_m \) is the potential topic, which is also the generating variable, so it is the document-topic distribution \( \theta_{mk} \) and the topic-word distribution \( \phi_k \) that need to be estimated. Parameter estimation methods currently include the EM (expectation maximization) algorithm, GS (Gibbs sampling), variational Bayesian estimation, message passing algorithm, mean-field variational expectation maximization, and expectation propagation algorithm.

This paper focuses on the collapsed Gibbs sampling algorithm for LDA. "Collapsing" in Gibbs sampling refers to the use of statistical relationships in the form of integrals to achieve avoidance of direct computation of the implied parameters. Based on the model in Figure 2 and the generation process in Section 3.1.1, the joint probability distribution of the model can be calculated.

\[
p(w, z | \alpha, \beta) = p(w | z, \beta) p(z | \alpha),
\]

\[
p(w | \alpha, \beta) = \int p(w | z, \Phi) p(\Phi | \beta) d\Phi = \prod_{m=1}^{M} \frac{\Delta(n_{m, \beta})}{\Delta(\alpha)},
\]

\[
p(z | \alpha) = \int p(z | \Theta) p(\Theta | \alpha) d\Theta = \prod_{m=1}^{M} \frac{\Delta(n_{m, \alpha})}{\Delta(\alpha)},
\]

\[
n_z = \{n_z(t)\}_{t=1}^{V},
\]

\[
n_m = \{n_m(k)\}_{k=1}^{K},
\]

\[
\Delta(\alpha) = \prod_{k=1}^{K} \frac{T(\alpha_k)}{T(\sum_{k=1}^{K} \alpha_k)}.
\]

\( n_z(t) \) is the number of times a word \( t \) appears in topic \( z_k \); \( n_m(k) \) is the number of times a word in document \( d_m \) appears in topic \( z_k \); \( \Theta, \theta \) are a parameter space consisting of \( \alpha, \beta \).

\( \alpha = (\alpha_1, \ldots, \alpha_K) \),

\( \alpha_1 = a_2 = \cdots = a_K, \beta = (\beta_1, \ldots, \beta_K), \beta_1 = \beta_2 = \cdots = \beta_K. \)

Calculate the conditional posterior probability using the joint probability distribution calculated in (1).

\[
p(z_i = k | z_{i-1}) = \frac{p(w, z)}{p(w, z_{i-1})}
\]

\[
p(w, z) = \frac{p(w, z_{i-1}) p(w_i, z_i) p(z_i)}{p(w, z_{i-1})} = \frac{p(w, z_{i-1})}{p(w_i, z_i)} \frac{1}{p(w_i, z_i)} \Delta(n_z + \beta) \Delta(n_m + \alpha) \Delta(n_z - i - 1)\Delta(n_m - i - 1)
\]

\[
\sum_{k=1}^{K} n_k^{(i)} + \beta_1 \sum_{k=1}^{K} n_k^{(i)} + \beta_2 - 1
\]

Using the corollary of the prior distribution of the polynomial distribution, the Dirichlet distribution, and (10), the probability distribution of the target parameter can be obtained, and finally the estimation of the parameter is achieved.

\[
p(\theta_m | w, z, \alpha) = \frac{\prod_{n=1}^{N_m} p(z_{m,n} | \theta_{m,n}) p(\theta_{m,n})}{\int \prod_{n=1}^{N_m} p(z_{m,n} | \theta_{m,n}) p(\theta_{m,n}) d\theta_m},
\]

\[
\prod_{[i,z]} p(w_i | \Phi_k) p(\Phi_k | \beta) = \text{Dir}(\Phi_k | n_k + \beta),
\]

\[
\Phi_k = \frac{n_k^{(i)} + \beta_1}{\sum_{k=1}^{K} n_k^{(i)} + \beta_2}
\]

\[
\theta_{mk} = \frac{n_m(k) + \alpha_k}{\sum_{k=1}^{K} n_m(k) + \alpha_k}
\]

\[
\prod_{n=1}^{N_m} p(z_{m,n} | \theta_{m,n}) p(\theta_{m,n}) d\theta_m = \text{Dir}(\theta_{m,n} | n_m + \alpha).
\]

### 3.2. Word2vec

As a deep learning-based tool, word2vec provides a more efficient way of representing the semantic distance between words by calculating the cosine distance between the word vectors and obtaining the vector expressions of each word in the corpus, based on the corpus and optimization of the training model. Word2vec includes two word vector training models, CBOW and Skip-gram, both of which include an input layer, a projection layer, and an output layer, but the CBOW model predicts the current word from its context, whereas the Skip-gram model predicts the current word from its context. Similarly, word2vec includes two word vector optimization models, the Hierarchical Softmax model and the Negative Sampling model. The Hierarchical Softmax optimization method constructs a binary tree using the frequency of words in the corpus as weights and maps the leaf nodes to all words, while the Negative Sampling optimization method uses relatively simple resampling to improve the speed of word vector training. By combining the two training models with the two optimization methods, a total of four frameworks for training word vectors can be obtained. As a tool for training word vectors, it can quickly and efficiently represent words in a corpus as vectors and capture the semantic features and
similarities between words for use in other related application studies. The sentiment intensity of the text data is obtained and used in the correlation analysis of the change of sentiment intensity over time for each topic.

3.3. Specific Study Process. In this paper, we investigate a method for sentiment intensity analysis (shown in Figure 3), using probabilistic topic modeling and deep learning to crawl text data from campus forums based on selected hot events and clean the data. Then, we train the model to analyze selected topics that are meaningful or need attention. Based on explicit sentiment words, we use word2vec to obtain implicit sentiment words in the text data and calculate the sentiment intensity of each text. Finally, we perform sentiment intensity time-series analysis under the selected topics.

1. Data crawling and preprocessing. The two main tasks are text data capturing and text data preprocessing. The crawler collects and crawls time-series data related to hot events. After crawling the data, the crawled data is preprocessed by word separation, spam filtering, and deactivation of words.

2. Topic modeling and filtering. After acquiring the data and preprocessing it, probabilistic topic modeling is carried out using the LDA model.

3. Sentiment intensity calculation. The calculation of sentiment intensity is carried out in parallel with the LDA theme modeling. Sentiment intensity was calculated by importing the data into word2vec for word vector cosine distance calculation and then by SO-SD.

3.3.1. Deep Learning Sentiment Analysis Methods Incorporating Self-Attention Mechanisms. This paper further incorporates deep learning sentiment analysis with a self-attentive mechanism based on traditional methods, which can effectively improve the accuracy of temporal analysis.

The model in this paper is divided into six main layers: the word embedding layer, the BGRU layer, the self-attentive layer, the multi-granularity convolutional layer, the attention-based pooling layer, and the fully connected and classified layer, as shown in Figure 4.

The comment text is transformed into word sequences using a word separation library; i.e., the text data is pre-trained to achieve word vector mapping.

Implementing a word vector to a bidirectional gated cyclic unit to learn serialization features, the operation can be combined via a forward GRU and a reverse GRU, followed by the output, as shown in the following equations:

\[ h_t^+ = f_{\text{GRU}}(h_{t-1}, v_t), \]

\[ h_t^- = f_{\text{GRU}}(h_{t-1}, v_t), \]

\[ h_t = [h_t^+, h_t^-]. \]

The self-attentive mechanism is used to perform an initial screening of features after sequence analysis and extract features with high task relevance. The specific form is as follows: global information is considered, word weight values \( \alpha \) are calculated, and the weights are weighted and summed with the features at each moment to obtain the highlighted new focus feature \( H_t \). This formula is shown as follows:

\[ H_t = \sum \alpha_i h_i, \]

where \( \alpha_i \) is the feature weight and the condition \( \sum \alpha_i = 1 \) is satisfied.

This layer receives the output features from the attention layer and selects convolutional kernels of different sizes for further feature extraction using the ReLU function, which
speeds up the training convergence and effectively avoids problems such as gradient explosion and disappearance. The feature extraction process is shown as follows:

\[ c_i = f_{\text{ReLU}}(w \cdot x_i + h - 1 + b), \quad (17) \]

where \( w \) is the convolutional kernel weight, \( h \cdot m \) denotes the convolutional kernel window granularity, and \( f_{\text{ReLU}} \) is the activation function.

After the model has obtained the feature map, pooling is used to reduce the amount of training data and model parameters to further find out the factors that have the greatest impact on the final sentiment polarity classification results, and this paper uses the attention mechanism instead of the traditional pooling layer to improve the feature extraction capability, as shown in the following equation:

\[ P_i = \sum a'_i c_i, \quad (18) \]

where \( a'_i \) is the feature weight and satisfies \( \sum a'_i = 1 \).

The feature map of the data obtained in the convolution layer is extracted by self-attentive dimensionality reduction to obtain a local feature sequence. This layer feeds the feature sequence into the fully connected layer for feature fusion and is classified by the classification layer for emotional polarity. The specific operation process is as follows: the sequence is processed through the fully connected layer to obtain the feature sequence \( D_{\text{final}} \); the classification layer performs feature fusion analysis through Softmax and transforms \( D_{\text{final}} \) to derive the probability distribution of the two emotional polarities. The calculation formula is shown in the following equation:

\[ p_i = \text{Softmax}(W \cdot D_{\text{final}} + b), \quad (19) \]

where \( p_i \) is the probability distribution of sentiment polarity, \( W \) is the weight matrix of Softmax, and \( b \) is the offset.

The sentiment intensity calculation is performed simultaneously with the LDA modeling, which is more complex compared to the topic modeling.

The processed data were transformed into word vectors, the sentiment lexicon was combined with word2vec and LSTM networks to label sentiment word lexicality, and finally SO-SD was applied to calculate sentiment intensity. The specific calculation formula is shown as follows:

\[ SO - SD(\text{word}) = \sum_{\text{pwords}} SD(\text{word}, \text{pword}) - \sum_{\text{nwords}} SD(\text{word}, \text{nword}), \quad (20) \]

where \( \text{pword} \) denotes a positive dominant sentiment word that is highly correlated with word; \( \text{Powords} \) denotes a positive dominant sentiment word that is most correlated with word; \( \text{nword} \) denotes a negative dominant sentiment word that is highly correlated with word; and \( \text{Nwords} \) denotes a negative dominant sentiment word that is most correlated with word. The specific formula is shown as follows:

\[ SD(\text{word1}, \text{word2}) = \frac{\sum_{k=1}^{n} x_{1k} x_{2k}}{\sqrt{\sum_{k=1}^{n} x_{1k}^2} \sqrt{\sum_{k=1}^{n} x_{2k}^2}} \quad (21) \]

where \( n \) is the word vector dimension, \( x_{1k} \) is the \( k \)-dimension value of the first word vector, and \( x_{2k} \) is the \( k \)-dimension value in the second word vector.

Once the SO-SD values are obtained, \( p \) and \( q \) are used as thresholds to make judgments as follows:

\[ SO - SD(\text{word}) \begin{cases} > p & \text{The Word is a positive implicit emotion word,} \\ \in [p, q] & \text{The word is neutral,} \\ < q & \text{The word is a negative implicit emotion word.} \end{cases} \quad (22) \]
All positive and negative sentiment words are represented as +1 and −1, respectively. Finally, according to the emotional words and their corresponding emotional intensity, the evaluation is carried out in the campus comment text.

3.3.2. Building Predictive Models. Existing prediction models are mainly based on graph neural networks, using them as feature extractors, embedding nodes in vectors, and implementing sequence learning with recurrent neural networks. These methods require information about the nodes over the entire time span and have limited applicability to scenarios where the nodes change frequently. Therefore, recurrent neural networks are usually used for dynamic injection into graph convolutional network parameters to form evolving sequences to solve this problem.

This paper introduces an attention mechanism into graph neural networks on this basis, enabling adaptation to different neighbor weights shared across all edges in the graph, without relying on pre-access to the global graph structure and node features. The prediction model is based on a graph attention network and gated recurrent units, where the graph attention network is the basis and the gated recurrent network implements network parameter updates. The node features are convolved with the graph attention network to obtain a temporal node embedding matrix, and the weight matrix evolves over time through the gated cyclic unit, with the dynamic evolution of the $l$-th level of the weight parameter matrix being shown as follows:

$$W_{t}^{(l)} = \text{GRU}(H_{t}^{(l)}, W_{t-1}^{(l)}),$$

where $W_{t}^{(l)}$ and $W_{t-1}^{(l)}$ are the weight parameter matrices for time period $t$ and time period $t-1$, respectively; $H_{t}^{(l)}$ is the time period $t$, $l$-layer node embedding matrix, updated in the manner shown in the following equation:

$$H_{t}^{(l+1)} = \text{GAT}(A_{t}, H_{t}^{(l)}, W_{t}^{(l)}),$$

where $A_{t}$ is the corresponding adjacency matrix for time period $t$, which is used to store the inter-vertex relationship, and the node embedding matrix for time period $t$ and layer $l+1$ is obtained by calculation. The model can predict the possible sentiment words and the corresponding scores in the next time period by the graph structure in the time period, through the multilayer perceptron MLP, with $x$ graph structure information, and finally all scores are summed up as the sentiment words derived from the time period prediction.

4. Experiments and Analysis

4.1. Experimental Setup

4.1.1. Experimental Environment. The model was trained using Ubuntu 18.04 operating system, Python 3.8 TensorFlow 1.14 deep learning framework was used for stability considerations, and the hardware configuration is shown in Table 1.

<table>
<thead>
<tr>
<th>Emotional tendency</th>
<th>Amount</th>
<th>Average length</th>
<th>Mark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>7,000</td>
<td>112.11</td>
<td>1</td>
</tr>
<tr>
<td>Negative</td>
<td>3,000</td>
<td>169.06</td>
<td>0</td>
</tr>
</tbody>
</table>

4.1.2. Experimental Dataset. To test the ability of this model to classify sentiment polarity in the Chinese corpus, two datasets, the online_shopping_10_cats provided by Chinese NLP corpus and the collected campus review dataset, were used for this experiment.

The dataset contains 2 sentiment tendency labels, 0 for negative tendency sentiment and 1 for positive tendency sentiment. The experiments used 52,783 training data items and 10,000 test data items, which were balanced so that each has 5,000 positive and 5,000 negative data items. Specific examples are shown in Table 2.

The campus review dataset contains 7,000 and 3,000 positive and negative disposition sentiment data items, respectively, with 9,000 training data items and 1,000 test data items being set in this experiment; the specific parameters are shown in Table 3.

4.1.3. Parameter Settings. The use of moderate dimensional word embeddings preserves the deeper meaning of words and reduces the training overhead and computation. In this paper, a trained high-dimensional word vector model, Chinese Word Vectors, modified by word2vec, is used.

In order to fully extract the text features, four different convolutional kernels are used for contextual information extraction, and a dropout mechanism is added to avoid the phenomenon of fitting; the specific parameter configuration is shown in Table 4.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word vector dimension</td>
<td>300</td>
</tr>
<tr>
<td>Size of convolution kernel</td>
<td>(2, 3, 4, 5)</td>
</tr>
<tr>
<td>Layer of BGRU</td>
<td>128</td>
</tr>
<tr>
<td>Batch size</td>
<td>50</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.00012</td>
</tr>
<tr>
<td>Dropout rate</td>
<td>0.5</td>
</tr>
<tr>
<td>Optimizer</td>
<td>Adam</td>
</tr>
</tbody>
</table>

4.1.4. Model Parameter Configuration. The use of moderate dimensional word embeddings preserves the deeper meaning of words and reduces the training overhead and computation. In this paper, a trained high-dimensional word vector model, Chinese Word Vectors, modified by word2vec, is used.
4.2. Assessment Criteria and Comparison Experiments

4.2.1. Assessment Methods. In this paper, accuracy (ACC) and cross-entropy loss (CEL) functions are used to evaluate the parameters.

The formula for calculating ACC is as follows:

\[
\text{ACC}(i) = \frac{T_i}{N},
\]

(25)

where \( T_i \) is the number of correctly classified data entries and \( N \) is the total number of data entries.

The GEL is calculated as follows:

\[
\text{GEL} = \frac{1}{N} \sum [y \ln a + (1 - y) \ln (1 - a)],
\]

(26)

where \( a \) is the true output, \( y \) is the desired output, and \( N \) is the total number of texts.

4.2.2. Comparative Experiments. To demonstrate the validity of the model, this paper also compares it with other better-performing sentiment analysis models.

(1) Text sequences were extracted using LSTM, and the comparison experiments were based on the results of the network proposed by Zaremba et al. with partial adaptations.

(2) Bi-LSTM [25] combines contextual information to improve feature extraction of text sequences.

(3) GRU improves model training speed by simpler structure.

(4) Bidirectional GRU (BGRU) enhances the combination of contextual information.

(5) ATT-BiLSTM uses an attention mechanism to differentiate the importance of the output of the BiLSTM network.

(6) TextCNN uses CNN networks to extract local features from text.

4.3. Results and Analysis

4.3.1. Test Results. The accuracy of each model is shown in Table 5.

Because the same dataset was used, some of the algorithms used results from the existing literature.

4.3.2. Accuracy Analysis. As shown in Table 5, the model in this paper achieves 92.75% accuracy on the hotel review dataset, which is the best result for the reformulation group. The experimental results prove that the model in this paper has better performance in the Chinese corpus. At the same time, considering the reasonableness test of the model results, this paper also conducts ablation experiments, and the experimental results prove that the multi-granularity convolutional neural network has certain advantages in improving the model classification, especially with long text data. This paper concludes that convolutional neural networks can effectively reduce the dimensionality of data and improve the local feature extraction ability of the model.

The review of the experimental data revealed that the LSTM network structure with complex results and multiple parameters is prone to overfitting in training, reducing accuracy. For recurrent neural networks, the multi-granularity approach improved the accuracy by 0.76% and 1.2% over the single convolutional kernel approach for both datasets. The results demonstrate the effectiveness of multi-scale convolutional kernel feature extraction in sentiment classification tasks, and the self-attentive mechanism further improves the model feature extraction capability and classification accuracy compared to the traditional approach.

4.3.3. Analysis of Callback Parameters. The analysis of each parameter of the training process of the model in this paper, as shown in Figure 5, shows that the loss function decreases rapidly with increasing number of training rounds and tends to converge.

On the online_shopping_10_cats dataset, after 6 rounds of training, the loss of the training model will decrease. After 13 rounds, the loss function decreased more slowly, and finally after 18 rounds, the loss function stabilized at 0.009. In the campus comments dataset, the loss function of the model decreased faster and converged after the 8th round. After the 15th round, the loss function dropped to below 0.0001, and the best fit of the model to the dataset was achieved.

From Figure 6, it can be seen that on the online_shopping_10_cats dataset, the test accuracy of the model is
relatively short and the model fits the training data better as the number of experimental rounds increases, but the test accuracy does not change significantly. The test accuracy did not change significantly.

As shown in Figure 7, for the campus review data, the model's test accuracy gradually improved from the first five rounds, and in the 11th round the model performance stabilized.

As shown in Figure 8, the model uses the graph neural network and the adaptive weighting method to effectively improve the performance of evolutionary prediction and achieves good results in the evolution of emotional trends of campus comments. Leveraging the campus review text's semantic relations, structural properties, transforming the text into a graph structure, and then extracting features for predictive modeling through a graph convolutional network. The final result comparison graph proves that the improved scheme is effective.

The above experimental results show that the model can be stabilized with fewer training rounds, with fast convergence, high accuracy of sentiment prediction, and better performance.

To further validate the effectiveness and superiority of the model, this paper randomly divides the dataset into a training set and a test set in a ratio of 8:2. For each dataset, the experiment is repeated three times, and the average of accuracy, precision, recall, and AUC is taken as the final result. The main configuration parameters of the LSTM model contain the maximum number of training rounds (epoch = 10) and word embedding size (embeddingSize = 100).

Comparing this method with SVM and textCNN, as shown in Table 6, we find that textCNN is outstanding in shallow text feature extraction but is not effective in long text areas due to modeling limitations and discourse order insensitivity. LSTM, on the other hand, can capture sequence information and is more effective in sentiment analysis. In this dataset, LSTM achieves accuracy, precision, recall, and AUC of 96.1%, 84.2%, 88.9%, and 0.904 (threshold = 0.7), which are 3.2%, 0.9%, 3.3%, and 0.053 higher than textCNN and 7.2%, 4.8%, 7.7%, and 0.0821 higher than SVM, respectively. The above data demonstrate the superiority of using LSTM models for solving text sentiment analysis problems.

In this paper, we analyze the influence of different word vector dimensions on the performance of the model. As shown in Figure 9, word vector data with dimensions of 50, 100, 150, and 200 are processed, and the accuracy rate can reach 89.3%, 89.9%, 88.9%, and 88.6%, respectively, by comparing the LSTM model with the annotated text, and the results show that the accuracy is higher when the word vector dimension is 100. Therefore, the model is trained with a word vector dimension of 100.

In this paper, we also consider the influence of the maximum number of training rounds of the LSTM model on the results, as shown in Figure 10, the maximum number of training rounds epoch is a key parameter affecting the performance. Too many times will cause the predicted value of the fitted loss rate available estimation model to be inconsistent with the true value. The experiments were carried out with 5, 10, 15, 20, and 25 dimensions. The results showed that with the increase of dimensions, the loss rate decreased first and then increased. When the dimension was 10, the
loss rate reached the best value of 16.8%. Therefore, the maximum number of training rounds in this paper was set to 10.

Overall, it can be found that the sentiment intensity analysis method based on probabilistic topic model and deep learning model proposed in this paper can obtain different views and student evaluation in public opinion based on campus comments more accurately and can reasonably and effectively analyze the change of sentiment intensity of each opinion and event over time. This method not only effectively makes up for the shortcomings of traditional opinion analysis methods in terms of not being able to obtain trends of changes in sentiment and make effective predictions for managing public opinion, but also allows for targeted sentiment analysis based on students’ different views and opinions and can calculate and analyze students’ public opinion and their sentiment more accurately, which is of great significance and value for monitoring and managing public opinion. The method also has a certain degree of significance in guiding students and university opinion management workers in public opinion, helping them to handle and respond to public opinion in a reasonable manner and carry out more scientific and effective opinion management.

5. Conclusion

For campus opinion analysis and management, which is essentially a text sentiment analysis task based on Chinese corpus, this paper proposes a deep learning sentiment analysis method incorporating a self-attention mechanism. The model uses a recurrent neural network structure to extract the sequence features of the text; then uses the self-attentive mechanism to highlight the sentiment classification features, which are fed into a multi-granularity convolutional neural network for higher-level feature extraction; uses the self-attentive mechanism to implement the pooling function; and finally achieves the sentiment polarity determination of the text. Experiments on two typical datasets demonstrate that the model has improved the accuracy of sentiment polarity prediction compared with the current mainstream methods. In addition, this paper demonstrates the soundness of the model structure through ablation experiments. By using a combination of convolutional neural networks and recurrent neural networks, the model achieves good results on both datasets. However, in reality, there is often a lack of high-quality training data in the target domain, and how to improve the ability of deep learning models to solve cross-domain problems from existing training datasets is worthy of further research.

Data Availability

The experimental data used to support the findings of this study are available from the author upon request.

Conflicts of Interest

The author declares no conflicts of interest regarding this work.

References


J. Lu and X. Li, “Neural network language model,” *Communication World*, vol. 24, no. 9, pp. 94-95, 2017.