International Journal of Ambient Computing and Intelligence
Volume 14 • Issue 1

Fusion of XLNet and BiLSTM-TextCNN for Weibo Sentiment Analysis in Spark Big Data Environment

Aichuan Li, College of Information and Electrical Engineering, Heilongjiang Bayi Agricultural University, China*
Tian Li, Heilongjiang Bayi Agricultural University, China

ABSTRACT

This article proposes a Weibo sentiment analysis method to improve traditional algorithms' analysis efficiency and accuracy. The proposed algorithm uses deep learning in the Spark big data environment. First, the input data are converted into dynamic word vector representations using the Chinese version of the XLNet model. Then, dual-channel feature extraction is performed on the data using TextCNN and BiLSTM. The proposed algorithm uses an attention mechanism to allocate computing resources efficiently and realizes feature fusion and data classification. Comparative experiments are conducted on two public datasets under identical experimental conditions. In the NLPCC2014 and NLPCC2015 datasets, the proposed model improves the precision and F1 metrics by at least 4.26% and 2.64%, respectively. In the weibo_senti_100k dataset, the proposed model improves the precision and F1 metrics by at least 4.66% and 2.69%, respectively. The results indicate that the proposed method has better sentiment analysis and prediction abilities than existing methods.

KEYWORDS

BiLSTM, Spark, Text Sentiment Analysis, TextCNN, XLNet

INTRODUCTION

With the rise of social media, microblogging has become a popular platform, drawing the attention and participation of many users. Users are free to express their opinions and share news and life moments on microblogs, which leads to a large amount of information emerging on such sites. This surge in information volume presents a significant hurdle regarding people's ability to efficiently access and process information (Jia & Han, 2020; Chen et al., 2023).

Information overload has become a common problem for microblog users. When faced with a large amount of microblog information, users often find it difficult to filter and understand the content quickly and effectively (Banik et al., 2023). This requires the use of data mining techniques to help users extract valuable information from massive microblog data. Data mining techniques can help users discover topics of interest, people of concern, and popular events by analyzing patterns and

DOI: 10.4018/IJACI.331744
*Corresponding Author

association rules in microblogging data, thus reducing the pressure caused by information overload (Dina et al., 2021; Pham et al., 2021).

Based on data mining, sentiment analysis techniques can help users understand microblogs' emotional tendencies and attitudes. Since microblogging provides a platform for users to express themselves freely, it contains rich emotional information. Sentiment analysis can be used to categorize microblog texts into positive, negative, or neutral sentiment categories through natural language processing and machine learning techniques. The results of such sentiment analysis can help users better understand the emotional information conveyed in microblogs. Consequently, users are empowered to gain more precise insight into the attitudes and emotions expressed by fellow users. Al-Shabi (2020) combined supervised machine learning and lexical knowledge methods for linguistic sentiment categorization. However, this approach requires substantial computational time and is unsuitable for handling complex tasks. Kumar et al. (2019) classified comment sentiment by evaluating hybrid features obtained by combining machine learning features with dictionary features. However, this approach is limited by the accuracy of the a priori parameters of the feature weights. Meanwhile, improvements in computing power have paved the way for the widespread adoption of distributed computing frameworks such as Spark. This trend has also provided efficient and reliable computational methods for sentiment analysis (Chebbi et al., 2018; Farhan et al., 2018), bringing significant impetus to developing the sentiment analysis field.

To address the limitations of traditional sentiment analysis methods, this paper proposes a text sentiment classification method with the Spark framework for Weibo sentiment analysis under abundant datasets. In contrast to traditional deep learning-based text sentiment analysis methods, the innovations of the proposed method lie in the following:

1. To thoroughly learn the semantic information contained in the text data of microblogs, this paper performs feature fusion operations with TextCNN and BiLSTM. This approach can not only effectively enhance the extraction efficiency of local and global features in text semantic information but can also effectively handle text information of various lengths. Hence, the feature fusion approach described in this paper addresses the inability of traditional text classification methods to handle text data of different lengths. The dropout regular function included in the model effectively solves the problem of model overfitting.
2. To highlight the emotional tendencies embedded in the microblog text data, the attention mechanism is selected to assign weights to different sentence features, effectively improving the efficiency of emotional tendency recognition.
3. Considering the limitations of single-machine serial in big data processing, this paper deploys the sentiment classification algorithm based on hybrid learning in the Spark platform. This memory-based iterative computing framework effectively improves computing efficiency while exhibiting superior versatility and fusibility.

RELATED WORKS

Sentiment analysis is a technology with a wide range of applications that can help us better understand and respond to people's emotional requirements. It also improves the effectiveness of communication and decision-making processes. The existing sentiment analysis methods can be divided into three categories: those based on sentiment dictionaries, those based on traditional machine learning (Mironczuk et al., 2018), and those based on deep learning.

Sentiment Analysis Method Using a Sentiment Dictionary

Sentiment analysis is an automated technique that employs computer algorithms to detect and analyze the emotional tendencies expressed in text (Van Atteveldt et al., 2021). In recent years, researchers have proposed various sentiment analysis methods and corresponding solutions based on sentiment dictionaries.

For example, Wang et al. (2021) proposed a sentiment dictionary expansion method called DW-PMI based on word distance and point mutual information. The DW-PMI added four types of words—degree adverbs, negatives, emoticons, and sentiment dictionaries—to the sentiment dictionary and expanded the film review sentiment dictionary. However, this method relies on an a priori vocabulary ontology database and is significantly affected by word frequency and sentimental polarity in the thesaurus. Jia and Li (2020) utilized the existing sentiment dictionary to select a reference word for each sentiment classification. They calculated the similarity between the sentiment and reference vectors and improved the semantic calculation rules. However, this method lacks preprocessing steps for the features of sentiment words, resulting in excessive calculation and affecting the efficiency of sentiment analysis. Wei-Dong et al. (2018) constructed a network matrix connecting internet users through forwarding relationships. They studied the interactive evolution mechanism between the subject and the environment in public opinion communication using social network analysis methods and sentiment mining analysis techniques, expanding the sentiment dictionary. However, this method's sentiment analysis accuracy is limited due to the inherent disorder of public opinion communication and the difficulty in ensuring its accuracy. J. S. Yang et al. (2019) quantitatively analyzed the overall polarity of emojis and new words, classifying them by comparing them to preset thresholds. However, the sentimental polarity of emojis is unclear and changes rapidly, limiting the universality of this method.

Sentiment Analysis Method Using Machine Learning

In contrast to sentiment dictionary methods that demand substantial time and human resources, the traditional machine-learning algorithm can efficiently process relatively large datasets (Van Atteveldt et al., 2021). The following outlines several machine learning-based sentiment analysis methods and corresponding solutions proposed in recent years.

Anwar et al. (2022) used DT to conduct a smoking sentiment analysis on searched words with specified sentimental properties. However, this study could not guarantee that the obtained global optimal decision was the optimal solution. AlBadani et al. (2022) combined "universal language model fine tuning" with a support vector machine (SVM) and proposed a new method, ULMFiT-SVM, for sentiment analysis using a deep learning architecture. However, the addition of inductive transfer learning places the solution at risk of negative transfer, which leads to a decline in the performance of the target task. Prastyo et al. (2020) employed SVM to predict new public opinion data labels based on the sentistrength_id labeling approach. However, this method's kernel function mapping dimension increases with the sample size, causing tremendous pressure on computing resources. Ruz et al. (2020) used Bayesian network classifiers to conduct sentiment analysis on two datasets and obtain the main characteristics of event dynamics. However, the classification effect is not good in cases with a large number of attributes or a significant correlation between attributes. Talha et al. (2020) used Python to conduct a behavior change analysis of particular groups and sustainable, intelligent groups based on Bayesian classifiers. However, the scalability of this method is weak due to the inherent property of Bayesian sensitivity to the expression form of input data. In summary, traditional machine-learning-based methods remain unable to identify multiple options and entail high storage service costs. These inherent problems must be solved urgently.

Sentiment Analysis Method Using Deep Learning

Deep learning technology improves the accuracy and reliability of sentiment analysis and promotes the widespread popularity of sentiment analysis applications, bringing increased convenience and value to people's daily lives and professional endeavors (Dang et al., 2020). Several deep learning-based sentiment analysis methods and corresponding solutions have been proposed recently.

Basiri et al. (2021) proposed a new model called ABCDM. Using two LSTMs and GRUs, the ABCDM obtains the context by considering the time in two directions but cannot capture the word order. Wang et al. (2019) proposed a CNN-LSTM model comprising a regional convolutional neural network (CNN) and a long short-term memory (LSTM) model. However, this method has a narrow application field, and its scalability needs to be improved. Rhanoui et al. (2019) proposed combining CNN and bidirectional long short-term memory (BiLSTM) models and realized opinion analysis of long texts through doc2vec embedding. Convolutional neural networks have excellent properties, such as weight sharing, local connection, and translation, which make them successful in many visual tasks. However, regarding the task of coordinate modeling, these advantages can transform into defects and potentially affect the final model performance. Agarwal et al. (2019) proposed four different recurrent neural network (RNN) variants to analyze speakers' utterances in videos. Although their framework provides more optimized sentiment classification accuracy, it does not deviate from the RNN framework system in the model. Monika et al. (2019) studied sentiment analysis using RNN and LSTM units. Their approach can deal with long-term dependence but will cause significant pressure on computing resources and affect the computational efficiency of the model. Shoryu et al. (2021) proposed a text sentiment prediction method (CNN-LSTM) combining CNN with LSTM. The CNN-LSTM first transforms the words into vectors based on their frequencies and then convolves them using a CNN to obtain the features they contain. The data obtained after the convolution of the CNN are then used for secondary feature extraction by LSTM to obtain more text information and achieve text classification. Experiments prove that the CNN-LSTM model can better extract the multidimensional features of user text. Wang et al. (2021) proposed a deep learning framework that combines XLNet with capsule networks to achieve character recognition in text information (XLNet-Caps). Experiments prove that XLNet-Caps can effectively classify personality features and have good average prediction accuracy. A short text sentiment analysis method (CNN-BiGRU) combining CNN with BiGRU was proposed by Gao et al. (2022) to realize the prediction of sentiment attributes using syntax and feature words in the corpus as vectors. Although the existing deep learning methods can better learn word vector representations of text data and make better recommendations, they still have low accuracy rates. These methods cannot extract dynamic character-level word vectors and often fail to consider global and local contextual feature information.

In recent years, several promising sentiment analysis methods have appeared. However, the existing methods have problems, such as low accuracy. Therefore, this paper proposes a Weibo sentiment analysis method based on deep learning. The proposed model is used in the Spark big data environment. To verify the effectiveness of the proposed model, it is compared with existing methods, including CNN-LSTM (Shoryu et al., 2021), XLNet-Caps (Wang et al., 2021), CNN-BiGRU (Gao et al., 2022), DW-PMI (Wang et al., 2021), DW-PMI (Wang et al., 2021), and ULMFiT-SVM (AlBadani et al., 2022).

TEXT SENTIMENT ANALYSIS METHOD BASED ON THE XAMBILSTM-TEXTCNN MODEL

The traditional microblog opinion event sentiment analysis methods lack deep semantic analysis, leading to low sentiment classification accuracy due to the sparse features and thin contextual relationships of short texts. To address these problems, this paper proposes a text sentiment analysis method based on the XAMBiLSTM-TextCNN model in the Spark big data environment to mine the deep-level semantic associations of sentiment feature words.

Definition of the Problem

The Weibo sentiment analysis task uses natural language processing and machine learning techniques to analyze the textual content in Weibo and extract the corresponding sentiment tendency or polarity. The input text data are $X = [X_1, X_2, \ldots, X_{le}] \in R$, where L is the sequence length. First, XLNET word vector modeling is performed to obtain $S = [S_1, S_2, ..., S_{le}]$, then BiLSTM modeling is performed to obtain $Q = [q_1, q_2, ..., q_{le}]$, and then $M = [m_1, m_2, ..., m_{le}]$ is obtained through the attention mechanism. At the same time, feature T is obtained through TextCNN, and then T' is obtained through attention. Finally, feature fusion is performed, and the output is the prediction value Y.

Model Framework of XAMBiLSTM-TextCNN

This paper mainly predicts the sentiment of microblog comments through the dual-channel feature extraction technique. This independent parallel structure makes it possible to extract semantic features independently and completely from multiple perspectives in text feature extraction, providing comprehensive text semantic information for the classification layer. First, words are used as the smallest input unit of microblog text, and word vectors with rich semantic information are generated using the word-based Chinese XLNet pretraining model. Then, a dual-channel feature extraction model is constructed by selecting TextCNN and BiLSTM as parallel structures. The lexical properties (e.g., adjectives, adverbs, and nouns) are extracted from the data utilizing the TextCNN model, which plays a more significant role in the expression of emotions. An attention mechanism is added after the TextCNN model to assign different weights to its output. This augmentation enhances the focus on keywords by further fusing word-level features. Although BiLSTM can efficiently capture temporal information features, it fails to consider the critical parts of text context information. Therefore, an attention layer is added before the output layer of the BiLSTM model, enabling the model to pay more attention to the essential parts of the context by assigning different weights. Finally, the feature vectors of the outputs of TextCNN-attention and BiLSTM-attention are spliced, and then the SoftMax function is utilized to obtain the sentiment probability distribution at the sentence level. The training structure of the proposed XAMBiLSTM-TextCNN model is shown in Fig. 1.

XLNet Model

As an improvement over the bidirectional encoder representations from transformers (BERT) model, Z. Yang et al. (2019) proposed the XLNet presenting permutation language modeling (PLM). The PLM enables the model to consider preceding and subsequent information when predicting each target word. Moreover, it solves the problem of errors both in the fine-tuning and pretraining data. These errors are caused by the unavailability of the actual masked word at the fine-tuning stage in BERT and the problem of independence among masked words. Based on the AR language model, the location information of the target word is added, and the double-flow attention mechanism is proposed.

The PLM proposed in XLNet is based on the AR language model and introduces the idea of unordered natural automatic distribution estimation (NADE), which not only preserves the advantages of the AR language model but also uses context information to predict the target.

Supposing that the sequence length is *Le* and the total number of sorting methods is $m = Le!$, the calculation method of XL's full array model is as follows:

$$\max(\lambda) A_{p \sim p_T} [\sum_{i=1}^{Le} \lg R_{\lambda}(x_{p,i} \mid X_{p<i})] \tag{1}$$

where A represents a sequence set; $p \sim P_{\mathrm{Le}}$ are all possible text arrangements; $x_{p,i}$ is the current word, $X_{p<i}$ is the previous word of $i-1$; R is the probability that the prediction result is the current word; and λ is a parameter.

The XLNet introduces the two-stream self-AM composed of content stream self-AM and query stream self-AM to solve the issue of location information omission in the PLM. Fig. 2 shows the XLNet recurrent mechanism.

The learning method of the transformer is as follows:

Figure 1. Model training structure of XAMBiLSTM-TextCNN

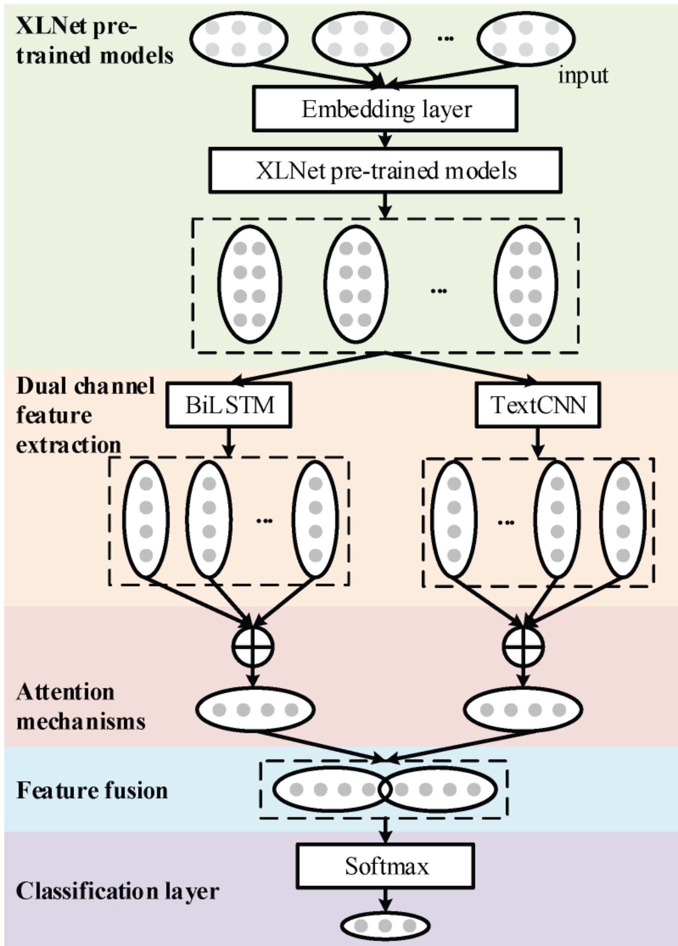
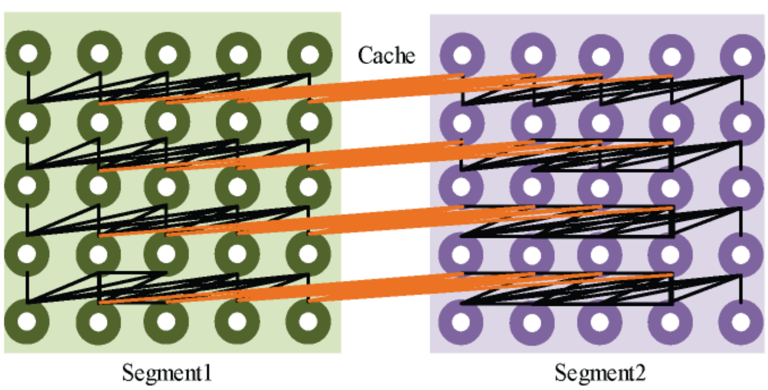


Figure 2. XLNet recurrent mechanism



$$S_{le} = f(S_{le-1}, W_{pd} + L_{1:l}) \tag{2}$$

where S_{le} is vector encoding at le; W_{pd} is the position encoding of the text vector of the current segment, and $L_{1:l}$ is location encoding, being the same in different segments.

BiLSTM Model

In LSTM, the internal correlation of text sentiment is strong. Therefore, LSTM bidirectional layers are used to build a BiLSTM network. BiLSTM stores past and future text sentiments through forward and backward LSTM, respectively. The single-layer BiLSTM network architecture is shown in Fig. 3.

The output vector of the BiLSTM at t is:

$$q_t = \mu^{t1} Y_t^1 + \mu^{t2} Y_t^1 + b^0 \tag{3}$$

where $\mu^{ti}(i \in \{1,2\})$ is the weight of the hidden layer, and $b^i(i \in \{0,1,2\})$ is the offset vector. After two layers of BiLSTM, the sentimental feature matrix Q of serialized text is obtained.

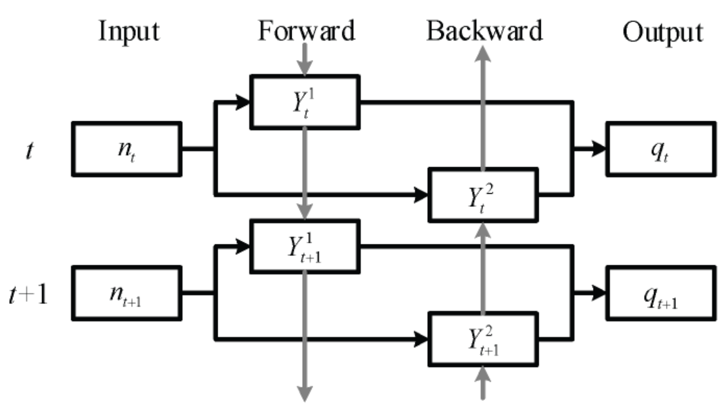
The input at t is n_t, while the hidden layers at t are H_t^1 and H_t^2.

TextCNN Model

The TextCNN model obtains the feature map representation by designing convolution kernels (CKs) for the convolution operation and further refines the extracted feature maps using the pooling layer. To resolve the problem that the maximum pooling method only retains unique elements and ignores the word order information of the text, this paper introduces the K-Max Pooling method to pool the feature map obtained by the convolution operation.

Unlike the square CK often used in the image field, the CNN uses a CK of size $z * k$ to perform convolution to obtain the local features of the text, where z is the height of CK and k is the dimension of the word vector. Setting the width of the CK equal to the dimension serves to preserve the integrity of the word vector. This dimension represents the smallest granularity of the text in the CK operation. Assuming that the word vector matrix obtained through the word embedding layer is $C \in R$, the word vector matrix c_i, composed of consecutive z words, is calculated from top to bottom along the direction of row C with the CK γ of size $z * k$.

Figure 3. Single-layer BiLSTM network architecture



After each CK scans the word vector matrix C, it generates a feature map $T \in R^{l-z+1}$, in which $T = [T_1, T_2, ..., T_{l-z+1}]$. In practical applications, multiple CKs of different types are employed for convolution operations to extract local features of different scales from multiple angles.

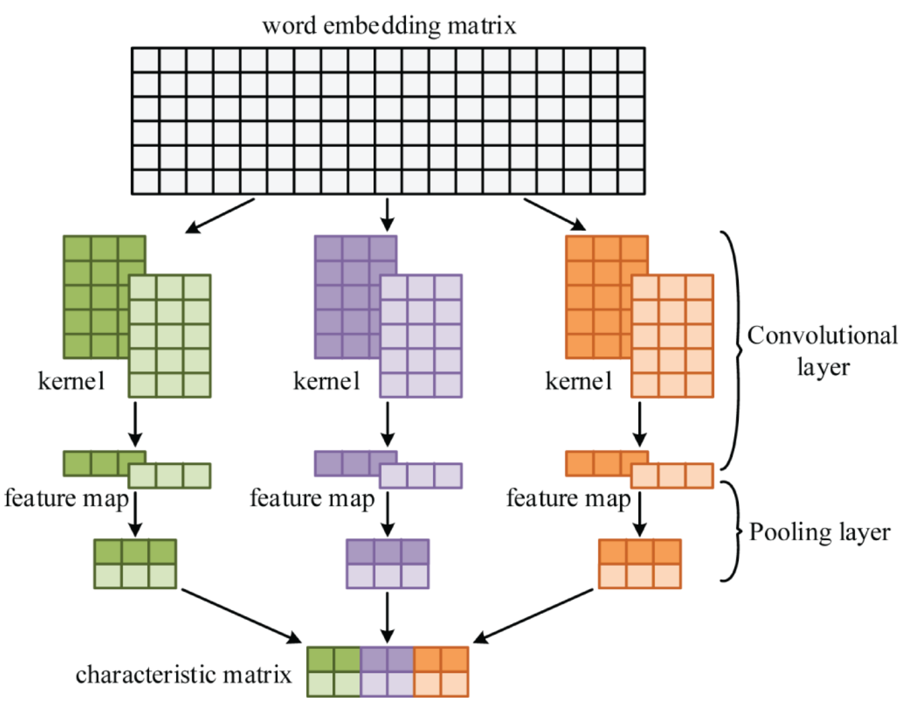
The commonly used pooling methods are max and average pooling. The calculation of both is shown in Equation 4:

$$\begin{cases} T_{\max} = \max(T_1, T_2, ..., T_{l-z+1}) \\ T_{avg} = \dfrac{(T_1 + T_2, +... + T_{l-z+1})}{l-z+1} \end{cases} \tag{4}$$

Max pooling retains the largest feature; average pooling is the arithmetic mean of all feature values in the feature map. Both pooling methods can reduce the feature map T to one dimension, considerably reducing the dimension of feature extraction and the subsequent calculation. However, the order of words is another sentence feature. The two abovementioned pooling methods only retain the unique elements, lacking the extraction of location features. Therefore, the K-Max Pooling method is employed in the TextCNN model. For the feature map T, the relative position of each feature is unchanged, and its maximum value γ is retained. This increases the consideration of other essential features.

After the word vector matrix is calculated using the TextCNN model, a new feature matrix $T \in R^{f*n}$ is obtained, where f is the sum of the feature numbers obtained after the convolution and pooling operations using CKs of different sizes, and n is the amount of each CK. Fig. 4 illustrates the TextCNN channel calculation process. It is assumed that γ is 2 in K-Max Pooling; that is, the

Figure 4. TextCNN channel calculation process



first two largest features of the six feature maps calculated by the convolution layer are retained to form the feature vectors. The features obtained by the CK operation of the same size are spliced along the column direction to obtain three feature matrices T_i. Then, these three T_i are spliced along the row direction to obtain the output matrix T of the TextCNN model. The row dimension of the matrix is 6(2*3), while the column is the amount of each CK, which is 2.

Attention Mechanism

The solution process of the attention value is shown in Fig. 5.

In Fig. 5, k_word_i represents the keyword, s_que_i represents the sequence query, $M(k,s)$ represents the similarity function, ω_i represents the weight, m_i represents the correlation calculation value between k_word_i and s_que_i, α_i and β_i are the weight coefficients, and Attention Value is the final calculated attention value. The three stages of AM are as follows:

First, the correlation between s_que_i and k_word_i is obtained. The common $M(k,s)$ methods are as follows:

1. Dot multiplication:

$$m_i = s\_que_i^{T} \cdot k\_word_i \tag{5}$$

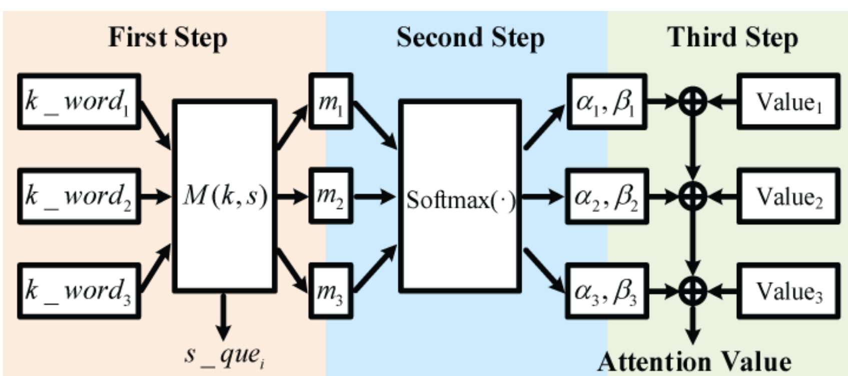
2. Matrix multiplication:

$$m_i = s\_que_i \times k\_word_i \tag{6}$$

3. Cos similarity:

$$m_i = \frac{s\_que_i \cdot k\_word_i}{\left\| s\_que_i \right\| \cdot \left\| k\_word_i \right\|} \tag{7}$$

Figure 5. Attention value calculation process



4. Series connection:

$$m_i = [s\_que_i; k\_word_i] \tag{8}$$

5. Multilayer perceptron:

$$m_i = \tanh(\alpha_i s\_que_i + \beta_i k\_word_i) \tag{9}$$

The second stage is normalization. The weights of important elements are highlighted to obtain the corresponding weight coefficient α_i:

$$\alpha_i = Soft\max(m_i) = \frac{e^{m_i}}{\sum_{j=1}^{L} e^{m_i}} \tag{10}$$

In the third stage, the final attention value is obtained by weighted summation:

$$Attention(k\_word_i, s\_que_i) = \sum_{i=1}^{L} \alpha_i \omega_i \tag{11}$$

EXPERIMENTS AND ANALYSIS

In the experiments, the four deep learning network models, TextCNN, XLNet, Bi-LSTM, and XAMBiLSTM-TextCNN, were trained on the combined dataset of NLPCC2014 and NLPCC2015 competitions and the weibo_senti_100k (Li et al., 2022) dataset. The accuracy (Acc), precision (Pre), recall (Re), and F1 of the trained models were compared and analyzed. Moreover, the experiment used data analysis to provide an intuitive evaluation of the performance of the sentiment analysis algorithm. This analysis was used to identify and assess the algorithm's limitations and shortcomings.

Experimental Environment

To measure the execution efficiency and scalability, all algorithms were run in the same software and hardware environment. The specific environment settings and parameter settings are shown in Tables 1 and 2, respectively.

Datasets

To prove the effectiveness of the hybrid learning model based on the sentiment dictionary and Bi-GRU in the analysis of Weibo sentiment tendency, the combined data of the NLPCC2014 and NLPCC2015 competitions were used.

NLPCC is an annual conference that attracts natural language processing enthusiasts from universities across China. The conference primarily focuses on natural language processing projects for Chinese texts. The above two competition data sets are related to Sina Weibo blog posts. All blog posts were labeled with sentimental tendencies in advance and divided into eight categories: no sentiment, surprise, fear, disgust, sadness, like, anger, and happiness. The data included a training set and a test set, and 60,000 pieces of data were finally sorted out.

This paper combines "like" and "happiness" as positive sentiments and "surprise," "fear," "disgust," "sadness," and "anger" as negative sentiments. It considers no sentiment as neutral.

Table 1. Experimental environment

Name	Related configuration
Development language	Java, JavaScript, Python
Server	CentOS7
JDK environment	JavaJDK 1.7.0_67
Database tools	MySQL5.1.27
Development tool	IDEA+PyCharm+Java under the framework SpringBoot, React+AntDesign
Hardware environment	Three computers: Intel(R) Core (TM) i5-4430s CPU@2.70 GHz, 4GB RAM

Table 2. Model parameter settings

Parameter	Value
Embedding size	256
Hidden Size of TextCNN	128
Hidden Size of BiLSTM	128
Activation	ReLU
Learning rate	0.000 1
Loss	categorical_crossentropy
Optimizer	Adam
Epoch	10
Bach_size	16

Evaluation Index

This section provides an effective evaluation and facilitates model comparison by analyzing the experimental results from different perspectives and employing various quantitative indicators. The main quantitative metrics are Acc, Pre, Re, and F1. These assessment metrics mainly comprise TP, TN, FP, and FN. TP and TN denote the numbers of correctly predicted positive and negative categories, respectively. FP and FN denote the numbers of incorrectly predicted positive and negative categories, respectively.

In the experiment, the positive and negative categories were used as positive and negative category samples to evaluate the model's accuracy in categorizing emotions. Taking the actual emotion label "happy" as an example, the specific distributions of TP, FP, TN, and FN are shown in Table 3, where A~H represent the eight emotion tendencies of no emotion, surprise, fear, disgust, sadness, liking, anger, and happiness, respectively.

Table 3. Example of experimental index division

	Predicted results A	Predicted results B~H
Actual category A	TP	FN
Actual category B~H	FP	TN

The specific calculation formulas of the evaluation indexes are as follows:

1. Acc:

$$\mathrm{Acc} = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

2. Pre:

$$\mathrm{Pre} = \frac{TP}{TP + FP} \tag{13}$$

3. Re:

$$\mathrm{Re} = \frac{TP}{TP + FN} \tag{14}$$

4. F1:

$$F1 = \frac{2Pre * Re}{Pre + Re} \tag{15}$$

Model Training

The dataset used in the model training part was NLPCC2014&NLPCC2015. Compared with the shallow neural network, the key advantage of the deep neural network is that it can learn, extract features, and constantly modify the model through continuous iteration. However, the overall performance is affected if the number of iterations is too high or too low. In a task, if the number of iterations surpasses a certain threshold, the features cannot be learned perfectly, limiting the performance. Conversely, if the number of iterations is less than a certain threshold, computational efficiency might suffer due to prolonged execution times.

Consequently, the selection of iteration times is vital in the task. Fig. 6 shows the relationship between simulation precision and iteration times in the experiment.

It can be seen from Fig. 6 that after the models have converged and stabilized, the prediction accuracy of the proposed XAMBiLSTM-TextCNN model stabilizes at 94.8%, which is 18.06%, 10.36%, and 2.93% higher than that of the TextCNN (80.3%), XLNet (85.9%), and BiLSTM (92.1%) methods, respectively. Hence, the proposed XAMBiLSTM-TextCNN model has better prediction accuracy than the other three models. In terms of convergence speed, the proposed XAMBiLSTM-TextCNN model can complete the convergence of the model when the number of iterations is 50, exhibiting a faster convergence speed than the other models. Therefore, the proposed algorithm has a higher operational efficiency, which saves more computational resources.

The relationship between the simulation F1-score and the number of iterations in the experiment is illustrated in Fig. 7.

Regarding the F1-score, all algorithms reach a balance when the number of iterations reaches 50~60. When the number of iterations is 80, the F1-score of the proposed XAMBiLSTM-TextCNN is the highest, reaching 96.5%. This is because the proposed

Figure 6. The relationship between precision and iteration time

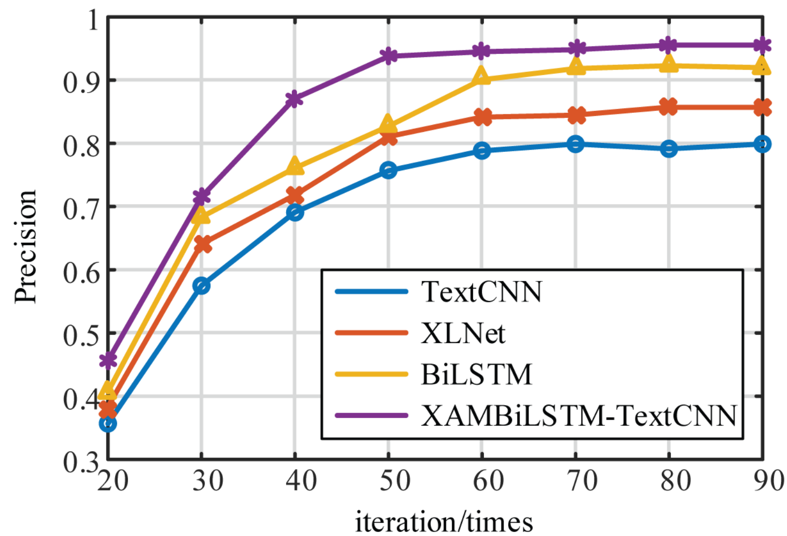
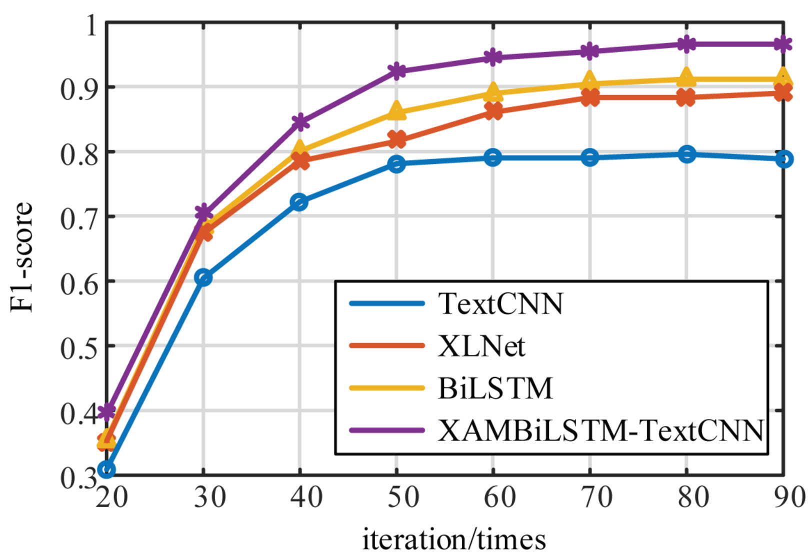


Figure 7. The relationship between the F1-score and iteration time



algorithm adopts a dual-channel feature extraction structure, which improves the efficiency and effectiveness of the algorithm. The F1-score of the TextCNN algorithm is the lowest at 80.1%. This is because the network structure of TextCNN is shallow. Much location information is lost after the text passes through the convolution and pooling layers. Consequently, the F1 evaluation index of comprehensive precision and recall is relatively poor and may converge to the local optimal solution.

Based on the above analysis, the number of experimental iterations was selected as 70 to ensure the stability and reliability of the detection results.

Experimental Comparison and Analysis

To further validate the effectiveness of the proposed XAMBiLSTM-TextCNN model in text sentiment analysis, NLPCC2014&NLPCC2015 and weibo_senti_100k were used uniformly, and comparative experiments were conducted under the respective optimal parameters of the models. The comparison models include DW-PMI (Wang et al., 2021), ULMFiT-SVM (AlBadani et al., 2022), CNN-LSTM (Shoryu et al., 2021), XLNet-Caps (Wang et al., 2021), and CNN-BiGRU (Gao et al., 2022). The accuracy and F1 scores of each model in the microblog text analysis task are listed in Table 4.

It can be observed from Table 4 that the XAMBiLSTM-TextCNN model improves the precision rate and F1 metrics by at least 4.26% and 2.64% in the NLPCC2014 and NLPCC2015 datasets, respectively. Although both CNN-LSTM and XLNet-Caps combine convolutional and recurrent neural networks in the model structure, they are not accurate enough in word vector transformation to generate context-rich information. XLNet-Caps combines XLNet and capsule networks to generate dynamic subvectors rich in contextual semantic information but cannot effectively capture global and local features. To verify the effectiveness of the proposed model more comprehensively, experiments were also conducted on the weibo_senti_100k dataset under the same experimental conditions. The results demonstrate that the model proposed in this paper improves the precision and F1 metrics by at least 4.66% and 2.69%, respectively.

Ablation Experiments

To verify the role played by the models XLNet, Text CNN, BiLSTM, and Attention in the proposed model, ablation experiments were designed to analyze the effect of each module. The experimental analysis was conducted on two datasets. Table 5 shows the experimental results, where “No XLNet” is the model input layer that uses only embedding for word vector representation, and “No TextCNN” is the model that removes the TextCNN module. “No BiLSTM” is the model that removes the BiLSTM module, and “No Attention” is the model that removes the attention module.

Table 5 shows that the accuracy (%) and F1-score (%) of the “No XLNet” model in the NLPCC2014 and NLPCC2015 datasets are 5.13% and 5.04% lower than those of the proposed XAMBiLSTM-TextCNN model, respectively. Meanwhile, the “No TextCNN” and “No BiLSTM” models exhibit lower accuracy (%) and F1-score (%) than the proposed XAMBiLSTM-TextCNN model. Specifically, there are four differences: 07% and 4.7% for accuracy and 2.31% and 2.57% for

Table 4. Comparative experimental results of model evaluation indexes

Datasets	Model	Precision (%)	F1-score (%)
NLPCC2014 & NLPCC2015	DW-PMI	83.43	84.54
	ULMFiT-SVM	86.43	87.74
	CNN-LSTM	88.92	92.37
	XLNet-Caps	90.23	93.19
	CNN-BiGRU	89.31	92.96
	XAMBiLSTM-TextCNN	94.49	95.83
weibo_senti_100k	DW-PMI	82.86	83.98
	ULMFiT-SVM	85.78	86.96
	CNN-LSTM	88.67	90.94
	XLNet-Caps	89.43	90.79
	CNN-BiGRU	91.57	92.38
	XAMBiLSTM-TextCNN	93.93	94.37

Table 5. Model ablation experiment comparison results

Datasets	Model	Precision (%)	F1-score (%)
NLPCC2014 & NLPCC2015	No TextCNN	90.42	91.13
	No XLNet	89.36	90.79
	No BiLSTM	92.18	93.26
	XAMBiLSTM-TextCNN	94.49	95.83
weibo_senti_100k	No TextCNN	88.67	90.94
	No XLNet	89.43	90.79
	No BiLSTM	91.57	92.38
	XAMBiLSTM-TextCNN	93.93	94.37

F1-score. The best results in sentiment analysis achieved by the model containing all the modules verify the model's effectiveness in this paper.

Discussion

This paper focuses on the multiangle experimental scheme to verify the effectiveness of the proposed model. Compared with the existing models, including DW-PMI, ULMFiT-SVM, CNN-LSTM, XLNet-Caps, and CNN-BiGRU, the proposed XAMBiLSTM-TextCNN model shows an improved performance in terms of precision (%) and F1-score (%) metrics by at least 4.26% and 2.64%, respectively, indicating that the proposed model can more accurately predict users' sentiment tendencies. The DW-PMI model based on the sentiment lexicon relies too much on an a priori lexicon ontology database. It is highly influenced by word frequency and sentiment polarity in the synonym dictionary. On the other hand, the machine learning-based sentiment analysis model ULMFiT-SVM suffers from an inability to handle large-scale data and is prone to overfitting. Deep learning-based sentiment analysis methods, including CNN-LSTM, XLNet-Caps, and CNN-BiGRU models, suffer from the problems of low accuracy, inability to extract dynamic character-level word vectors, and inability to consider global and local contextual feature information. The proposed XAMBiLSTM-TextCNN model effectively improves the extraction efficiency of local and global features in textual semantic information and adeptly manages textual information of different lengths. The proposed model efficiently highlights the emotional tendencies embedded in microblogging text data, facilitating improved comprehension and analysis of emotional tendencies in large amounts of textual data.

CONCLUSION

Given the bottleneck problem of enhancing the performance of sentiment analysis algorithms under massive data, this paper proposed a fusion model using XLNet and BiLSTM-TextCNN in the Spark environment. The experiments validated the effectiveness of the method. Based on the above analysis and description, the following conclusions can be drawn: 1) The fusion of TextCNN and BiLSTM models not only effectively improves the extraction efficiency of both local and global features in text semantic information but also effectively handles text information of various lengths, solving the problem faced by traditional text classification methods, which struggle to handle text data with varying lengths. 2) To highlight the emotional tendency embedded in the microblog text data, the attention mechanism is chosen to assign weights to different features in the sentence, emphasizing their importance and effectively improving the efficiency of emotional tendency recognition. 3) Considering the limitations of stand-alone serialization in big data processing, this paper deploys a hybrid learning-based sentiment classification algorithm in the Spark platform. This memory-based iterative computing framework effectively improves computational efficiency and boasts enhanced generalization and integrability capabilities.

The shortcomings of the proposed algorithm are mainly its high computational complexity and its limitations based on the experimental conditions. Future research will focus on reducing the complexity of the deep learning algorithm to improve fault tolerance. Additionally, the sentiment-labeled text will be expanded to include smaller granularity, considering the sentiment tendency of the information content of the pictures and videos. This expansion aims to enhance the scalability and adaptability of the work. The ultimate objective is to develop a more reliable deep learning model that offers high-quality sentiment analysis predictions.

ETHICS APPROVAL AND CONSENT TO PARTICIPATE

This manuscript does not involve research of human participants, data, or tissue. Hence, the manuscript does not require a statement of ethical approval and ethical consent.

CONSENT FOR PUBLICATION

This manuscript does not contain any person's data in any form. Thus, the consent of others is not needed.

DATA AVAILABILITY

The data included in this paper are available without any restrictions.

COMPETING INTERESTS

The authors of this publication declare there are no competing interests.

FUNDING

This research was supported by the NSFC Youth Science Fund Project (No. 32201655) and Heilongjiang Agricultural Reclamation Administration Projects (No. HKKY190201-02).

ABBREVIATIONS

Not applicable.

ACKNOWLEDGMENT

We wish to express our appreciation to the reviewers for their helpful suggestions, which significantly improved the presentation of this paper.

REFERENCES

Agarwal, A., Yadav, A., & Vishwakarma, D. K. (2019). Multimodal sentiment analysis via RNN variants. In *Proceedings of IEEE International Conference on Big Data, Cloud Computing, Data Science & Engineering (BCD)*. IEEE.

Al-Shabi, M. A. (2020). Evaluating the performance of the most important Lexicons used to sentiment analysis and opinions mining. *IJCSNS*, *20*(1), 1–12.

AlBadani, B., Shi, R., & Dong, J. (2022). A novel machine learning approach for sentiment analysis on Twitter incorporating the universal language model fine-tuning and SVM. *Applied System Innovation*, *5*(1), 13–25. doi:10.3390/asi5010013

Anwar, M. K., Yusoff, M., & Kassim, M. (2022). Decision tree and naive Bayes for sentiment analysis in smoking perception. In *Proceedings of IEEE 12th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*. IEEE. doi:10.1109/ISCAIE54458.2022.9794558

Banik, D., Satapathy, S. C., & Agarwal, M. (2023). Advanced weighted hybridized approach for recommendation system. *International Journal of Web Information Systems*, *19*(1), 1–18. doi:10.1108/IJWIS-01-2022-0006

Basiri, M. E., Nemati, S., Abdar, M., Cambria, E., & Acharya, U. R. (2021). ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. *Future Generation Computer Systems*, *115*(1), 279–294. doi:10.1016/j.future.2020.08.005

Chebbi, I., Boulila, W., & Mellouli, N. (2018). A comparison of big remote sensing data processing with Hadoop MapReduce and Spark. In *Proceedings of 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. IEEE. doi:10.1109/ATSIP.2018.8364497

Chen, M., Luo, X., Shen, H., Huang, Z., Peng, Q., & Yuan, Y. (2023). A Chinese nested named entity recognition approach using sequence labeling. *International Journal of Web Information Systems*, *19*(1), 42–60. doi:10.1108/IJWIS-04-2023-0070

Dang, N. C., Moreno-García, M. N., & De la Prieta, F. (2020). Sentiment analysis based on deep learning: A comparative study. *Electronics (Basel)*, *9*(3), 483. doi:10.3390/electronics9030483

Dina, N., Yunardi, R., & Firdaus, A. (2021). Utilizing text mining and feature-sentiment-pairs to support data-driven design automation massive open online course. [iJET]. *International Journal of Emerging Technologies in Learning*, *16*(1), 134–151. doi:10.3991/ijet.v16i01.17095

Farhan, M. N., Habib, M. A., & Ali, M. A. (2018). A study and performance comparison of MapReduce and Apache Spark on Twitter data on Hadoop cluster. [IJITCS]. *International Journal of Information Technology and Computer Science*, *10*(7), 61–70. doi:10.5815/ijitcs.2018.07.07

Gao, Z., Li, Z., Luo, J., & Li, X. (2022). Short text aspect-based sentiment analysis based on CNN+ BiGRU. *Applied Sciences (Basel, Switzerland)*, *12*(5), 2707. doi:10.3390/app12052707

Jia, K., & Li, Z. (2020). Chinese micro-blog sentiment classification based on sentiment dictionary and semantic rules. In *Proceedings of International Conference on Computer Information and Big Data Applications (CIBDA)*. IEEE.

Jia, L., & Han, X. (2020). Tracing Weibo (2009–2019): The commercial dissolution of public communication and changing politics. *Internet Histories*, *4*(3), 304–332. doi:10.1080/24701475.2020.1769894

Kumar, H. M., Harish, B. S., & Darshan, H. K. (2019). Sentiment analysis on IMDb movie reviews using hybrid feature extraction method. *International Journal of Interactive Multimedia & Artificial Intelligence*, *5*(5), 257–268. doi:10.9781/ijimai.2018.12.005

Li, Y., Guo, H., Zhang, Q., Gu, M., & Yang, J. (2018). Imbalanced text sentiment classification using universal and domain-specific knowledge. *Knowledge-Based Systems*, *160*(12), 1–15. doi:10.1016/j.knosys.2018.06.019

Li, Y., & Li, N. (2022). Sentiment analysis of Weibo comments based on graph neural network. *IEEE Access : Practical Innovations, Open Solutions*, *10*, 23497–23510. doi:10.1109/ACCESS.2022.3154107

Mironczuk, M. M., & Protasiewicz, J. (2018). A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications*, *106*(3), 36–54. doi:10.1016/j.eswa.2018.03.058

Monika, R., Deivalakshmi, S., & Janet, B. (2019). Sentiment analysis of US airlines tweets using LSTM/RNN. In *Proceedings of IEEE 9th International Conference on Advanced Computing (IACC)*. IEEE. doi:10.1109/IACC48062.2019.8971592

Pham, B., Jovanovic, J., Bagheri, E., Antony, J., Ashoor, H., Nguyen, T. T., Rios, P., Robson, R., Thomas, S. M., Watt, J., Straus, S. E., & Tricco, A. C. (2021). Text mining to support abstract screening for knowledge syntheses: A semi-automated workflow. *Systematic Reviews*, *10*(1), 1–11. doi:10.1186/s13643-021-01700-x PMID:34039433

Prastyo, P. H., Sumi, A. S., Dian, A. W., & Permanasari, A. E. (2020). Tweets responding to the Indonesian government's handling of COVID-19: Sentiment analysis using SVM with normalized poly kernel. *Journal of Information Systems Engineering and Business Intelligence*, *6*(2), 112–122. doi:10.20473/jisebi.6.2.112-122

Rhanoui, M., Mikram, M., Yousfi, S., & Barzali, S. (2019). A CNN-BiLSTM model for document-level sentiment analysis. *Machine Learning and Knowledge Extraction*, *1*(3), 832–847. doi:10.3390/make1030048

Ruz, G. A., Henríquez, P. A., & Mascareño, A. (2020). Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers. *Future Generation Computer Systems*, *106*(2), 92–104. doi:10.1016/j.future.2020.01.005

Shoryu, T., Wang, L., & Ma, R. (2021). A deep neural network approach using convolutional network and long short term memory for text sentiment classification. In *Proceedings of IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE. doi:10.1109/CSCWD49262.2021.9437871

Talha, I. M., Salehin, I., & Debnath, S. C. (2020). Human behaviour impact to use of smartphones with the python implementation using naive Bayesian. In *Proceedings of 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE. doi:10.1109/ICCCNT49239.2020.9225620

Van Atteveldt, W., Van der Velden, M. A., & Boukes, M. (2021). The validity of sentiment analysis: Comparing manual annotation, crowd-coding, dictionary approaches, and machine learning algorithms. *Communication Methods and Measures*, *15*(2), 121–140. doi:10.1080/19312458.2020.1869198

Wang, J., Yu, L. C., & Lai, K. R. (2019). Tree-structured regional CNN-LSTM model for dimensional sentiment analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*(3), 581–591.

Wang, Q., Zhu, G., Zhang, S., Li, K., Chen, X., & Xu, H. (2020). Extending emotional lexicon for improving the classification accuracy of Chinese film reviews. *Connection Science*, *33*(2), 153–172. doi:10.1080/09540091.2020.1782839

Wang, Y., Zheng, J., Li, Q., Wang, C. J., Zhang, H., & Gong, J. (2021). XLNet-Caps: Personality classification from textual posts. *Electronics (Basel)*, *10*(11), 1360. doi:10.3390/electronics10111360

Wei-Dong, H., Qian, W., & Jie, C. (2018). Tracing public opinion propagation and sentimental evolution based on public emergencies in social networks. *International Journal of Computers, Communications & Control*, *13*(1), 129–142. doi:10.15837/ijccc.2018.1.3176

Yang, J. S., Ko, M. S., & Chung, K. S. (2019). Social sentimental opinion decision with newly coined words and emoticon polarity of social networks services. *Future Internet*, *11*(8), 165–179. doi:10.3390/fi11080165

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc.

Aichuan Li, a Ph.D. in Electrical Engineering and Associate Professor, graduated from Heilongjiang Bayi Agricultural University in 2017 and worked in the Heilongjiang Bayi Agricultural University. His research interests include the Internet of Things and mechatronics engineering.

Tian Li is a master of Agricultural Engineering and Information Technology and is studying at Heilongjiang Bayi Agricultural University. His research interests include agricultural engineering and information technology.