Hindawi
Security and Communication Networks
Volume 2021, Article ID 5567991, 11 pages
https://doi.org/10.1155/2021/5567991

Research Article

Social Network Spam Detection Based on ALBERT and Combination of Bi-LSTM with Self-Attention

Guangxia Xu, Daiqi Zhou, and Jun Liu

School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China

Correspondence should be addressed to Guangxia Xu; xugx@cqupt.edu.cn

Received 4 February 2021; Revised 15 March 2021; Accepted 24 March 2021; Published 8 April 2021

Academic Editor: Hao Peng

Social networks are full of spams and spammers. Although social network platforms have established a variety of strategies to prevent the spread of spam, strict information review mechanism has given birth to smarter spammers who disguise spam as text sent by ordinary users. In response to this, this paper proposes a spam detection method powered by the self-attention Bi-LSTM neural network model combined with ALBERT, a lightweight word vector model of BERT. We take advantage of ALBERT to transform social network text into word vectors and then input them to the Bi-LSTM layer. After feature extraction and combined with the information focus of the self-attention layer, the final feature vector is obtained. Finally, SoftMax classifier performs classification to obtain the result. We verify the excellence of the model with accuracy, precision, F_1-score, etc. The results show that the model has better performance than others.

1. Introduction

Online social network platforms (OSNs) provide users with convenient communication and interactive tools, which can instantly share various content related to life and work, including text, pictures, and videos. Because of the support of wireless communication and computer, OSNs have become very popular than before [1]. However, a great quantity active user and the convenient conditions for publishing content have attracted a lot of spammers on OSNs. The release of spam by spammers has caused great troubles to normal users and platforms. According to reports, spam that had little or limited impact in the past can now take advantage of OSNs to cause a huge distributed impact [2]. OSNs disclose all basic user information and provide follow-up functions, which enables spammers to easily and accurately send spam to potential target users and promote dissemination [3]. For example, a social platform message embedded with a URL may spread to thousands of users within a few minutes.

Spam is a kind of information actively sent by spammers, and its purpose is to deceive, spread lies, and advertise for profits [4]. Spam will cause problems such as resource occupation, extended communication time, and bandwidth waste [5]. A report showed that on most OSNs, the growth rate of spam exceeded the rate of ordinary reviews [6]. It also showed that 15% of spams contain links to malicious content, pornography, or malware. Although a lot of related research has been done, there are still a large number of spams on social networks. And the spammer that manufactures and sends spam will disguise spam by observing the platform filtering strategies. It shows that there are still deficiencies in these methods.

In this paper, we mainly study a spam detection model in the Sina Weibo social network. This model combines ALBERT and Bi-LSTM network based on self-attention mechanism. The contribution of this work can be summarized as follows:

(i) In view of the huge amount of social network tweet data, introducing the ALBERT model to embedding the text to improve the efficiency of model classification while ensuring the accuracy of the model.

(ii) Aiming at the situation where spammer hides the spam camouflage in the normal text, introducing the Bi-LSTM to the spam tweet recognition method,

which can fully consider the context and semantics to capture the core of tweets text.

(iii) Because of the limited content of a single tweet (mostly less than or equal to 140 characters), which cannot provide a large amount of effective information, introducing the self-attention mechanism to the Bi-LSTM model to further obtain key words that affect the classification results of tweet characteristics.

The rest of this paper is organized as follows. We present background and related work in Sections 2 and 3, respectively. Section 4 introduces the architecture and details of the model. The implementation of the experiment is described in Section 5, and the results and analysis is discussed in Section 6. Section 7 concludes this paper and makes an outlook for future work.

2. Background

At present, there have been many related studies on the spam detection in social networks. Among them, part of the research has laid the foundation. Gupta et al. [7] conducted a security survey on pervasive online social networks, mainly exploring the trust management mechanism and anomaly detection mechanism of social network platforms, and raised current problems in anomaly detection, among which spam belongs to one. Branitskiy et al. [8] used machine learning methods to process social network data in order to explore the sensitivity of young generation to stimulating information on social networks. This research will help us distinguish between ordinary users and malicious users. Social network-based relationship graph methods are usually proposed when studying social networks, but the data are huge and not easy to calculate. Kolomeets et al. [9] proposed a reference architecture for storing and analysing graph data and provided visualization. Xu et al. [10] proposed a weighted algorithm for social network topology graphs, which can help researchers find smaller communities. In addition to the research on network topology, there is also a certain research foundation on the content of social network platforms. Jeong et al. [11] proposed a method to calculate text similarity-based edge weights. Experiments show that it can effectively find similar text data. Xu et al. [12] extracted audio and text from videos on social networks and used 3DCLS (3D Convolutional-Long Short-Term Memory) hybrid framework to analyse user emotions. In addition, there have been many studies for spam detections. After reviewing the relevant literature, it is found that the research can be classified to 2 types: traditional machine learning method and complex neural network method.

2.1. Traditional Machine Learning Method. Traditional machine learning method has been widely used in spam detection. These approaches build a classification model based on the characteristic attributes of the spam text. Most of these attributes include the number of URLs, key words, etc. [13]. Amayri and Bouguila [14] applied SVM to spam detection. They focused on the impact of SVM kernels. The results showed that string kernels are better than distance-based kernels. Vangelis et al. [15] made many different improvements to naive Bayes. They proposed five improved models based on Naive Bayes (NB), including multivariate Bernoulli NB, TF multinomial NB, boolean multinomial NB, flexible Bayes, and multivariate Gauss NB. The experiments show that flexible Bayesian and multinomial NB with Boolean attributes have a leading role. Soiraya et al. [16] took advantage of the J48 decision tree to analyse the Facebook message text. They detected the keywords, the average number of words, the length of the text, and the number of links contained in the information. The accuracy and recall rates of the research results were 61% and 63%, respectively. Johnson et al. [17] compared traditional machine learning algorithms with deep learning framework algorithms and found that random forests has better performance in text URL detection.

In addition to the classification model based on the single method, researchers also try to combine multiple methods to achieve better classification effect. The authors in [18] applied the Markov clustering (MCL) algorithm and allocated the probability of each node in the network by using the weighted graph as the input of the algorithm. In [19], a hybrid machine learning spam detection model is proposed based on support vector machine, and the experiments were carried out on four language datasets (Arabic, English, Spanish, and Korean). It showed that the accuracy of the model was better than other algorithms.

2.2. Complex Neural Network Method. In recent years, the model of deep neural network has shown strong ability in the field of natural language processing. It includes word representation learning, sentence and document representation, grammar analysis, machine translation, and sentiment classification. In the field of spam detection, many methods have been proposed. Tien and Nur [20] explored an artificial neural network language model to distinguish different users' short text writing styles, so as to distinguish and identify users, and then detect spam users. Ma et al. [21] established a microblog rumor detection system using LSTM and GRU. Ruan and Tan [22] introduced a technique based on three-layer back propagation neural network and feature construction based on concentration. In [23], a text classification model based on word2vec is constructed to solve the high-dimensional problem of traditional methods. In addition, LSTM is added to extract the key information of the text. Finally, this method is applied to the classification of patent texts. Experiments show that the accuracy of classification is 93.48%. Luan and Lin [24] proposed a text classification model called CNN-COIF-LSTM. The experimental results show that the combination of CNN and LSTM has higher accuracy without activation function or its variants. Recently, self-attention mechanism has attracted people's attention and achieved state-of-the-art results. Dong et al. [25] combined a self-interaction attention mechanism with label representation and used the Bert model to solve the problem of text feature extraction. In this method, joint word representation and label representation

are proposed to improve the efficiency of the model. Experiments also prove the correctness of the method.

3. Related Work

3.1. BERT and ALBERT. BERT (bidirectional encoder representations from transformers) was proposed by Devlin et al. [26] of Google in 2018, and immediately it showed a strong ability in the NLP field. The structure of Bert is shown in Figure 1.

BERT uses transformer [27] structure as the main framework. It converts the distance of two words at any position into one, which effectively solves the long-term dependency problem in NLP. The training process of BERT includes two parts: MLM (masked language model) and NSP (next sentence prediction). In the first part of experiment, 15% of the words were randomly masked, 80% of the words were replaced by (mask), 10% were replaced by any other words, and 10% were kept in the original state. In the second part, the model randomly extracts two consecutive sentences, 50% of which retain the extracted two sentences, their relationship is labeled IsNext, the other 50% of the second sentence is randomly extracted from the corpus, and their relationship is labeled NotNext.

The structure of ALBERT refers to BERT, and it still uses transformer and GELU activation function. However, at the same time, there are some innovations compared with BERT, including the following three points:

(1) Factored embedding parameterization: it reduces the word embedding dimension of embedding layer and adds a project layer between word embedding and hiding layer. Suppose the size of thesaurus is L, H represents the dimension of hidden layer, and the dimension of word embedding is V.

The calculation formula of the parameters of the BERT model is

$$P_{\text{bert}} = L \times H. \tag{1}$$

The calculation formula of the parameters of the ALBERT model is

$$P_{\text{albert}} = L \times V + V \times H. \tag{2}$$

In the ALBERT model, the dimension of word embedding is the same as the hidden layer. When V is large and V is far less than H, the number of parameters will be reduced after factorization of word embedding.

(2) Cross-layer parameter sharing: the parameters in each layer of transformer are relatively independent, including self-attention and full connection, which will lead to a significant increase of parameters when the layers increase. In order to decrease the number of parameters and promote the stability of the model, ALBERT tries to share all the parameters.

(3) Inter-sentence coherence loss: ALBERT has changed the NSP of BERT into a sentence order prediction (SOP). In SOP, the construction of positive example is consistent with NSP, but the negative example is to reverse the two sentences.

(4) Dropout was canceled.

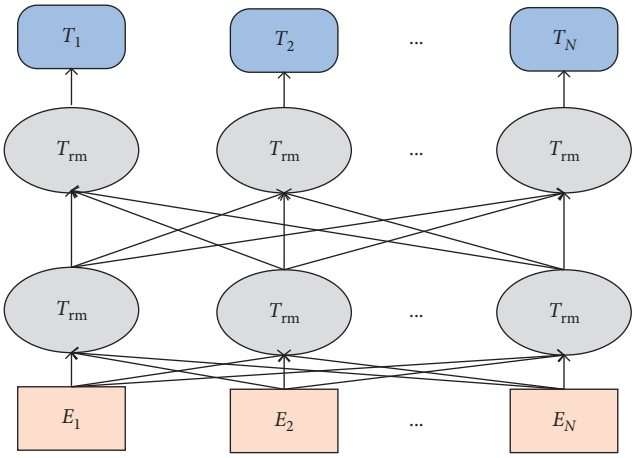

Figure 1: The structure of BERT.

3.2. LSTM and Bi-LSTM. Although RNN can support information persistence, it cannot achieve significant effect for some complex scenarios. For example, we try to predict the following: "I grew up in Sichuan I say "xxx" fluently". The RNN will give a name of language for field in, but the answer is wrong most of the time because the word "Sichuan" that is helpful for prediction is too far away. However, in the case of increasing the interval, we will not be able to learn the information between them. LSTM, a special RNN, was proposed by Hochreiter et al. [28] in 1996, and it can stably learn long-term dependent information. Having a chained network module is a feature of all RNNs. In a normal RNN, this repeating module has a very simple structure, such as a tanh layer. There is the same structure in LSTM, but the internal of the structure is different from RNN. The difference is that it has four modules that play different roles.

The structure of LSTM includes input gate i_t, forgetting gate f_t, output gate o_t, and cell state update vector c_t. The structure of LSTM is shown in Figure 2.

The forgetting gate at LSTM selects what information to discard from the cellular state. The formula of this part is as follows:

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right), \tag{3}$$

where W_f and b_f represent the weight matrix and bias matrix of forgetting gate, respectively. σ is the activation function, h_{t-1} represents historical information, and x_t is based on the current input of new information to determine which old information to forget. In this formula, the output h_{t-1} of the previous stage is calculated and combined with x_t. The formula will output a value between 0 and 1. 0 indicates that the old information is completely discarded, and 1 means that the old information is completely retained.

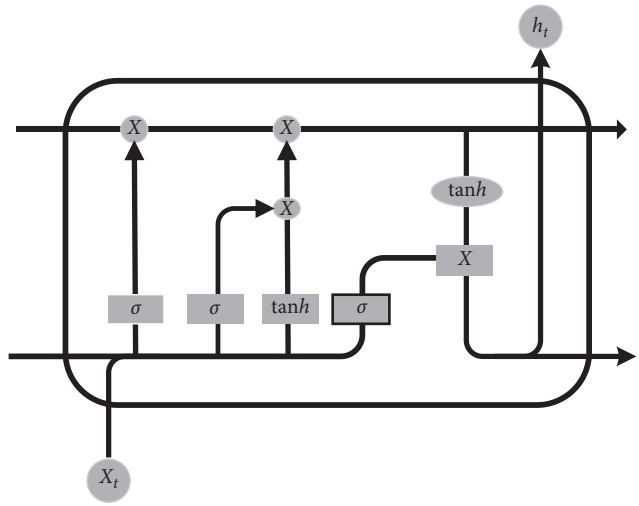

Figure 2: The structure of LSTM.

Choosing what sort of information to store in cell is the main responsibility of input gate. The input gate contains two parts. In the first part, the sigmoid layer (input gate layer) decides what value to update. In the second part, the tanh layer will build a new candidate value vector $\widetilde{C}_t$ and put it in the state. After the coefficients of the input gate and the forget gate are obtained, the current cell state is updated. The formula is as follows:

$$\begin{aligned} i_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right), \\ \widetilde{C}_t &= \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right), \\ C_t &= f_t * C_{t-1} + i_t * \sim C_t, \end{aligned} \tag{4}$$

where W_i and b_i, respectively, represent the weight of the input gate and tanh is the activation function.

The output gate determines what information will be output at the current stage. In the first step, it calculates a sigmoid layer to decide for which section of the cell state will be output. Then, we will get a value between −1 and 1 through the tanh layer and multiply it with the value of the first step output. Finally, we only output the part that we decided. The formula is as follows:

$$\begin{aligned} O_t &= \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right), \\ h_t &= o_t * \tanh\left(C_t\right), \end{aligned} \tag{5}$$

where W_o and b_o represent the weight matrix and the offset matrix of the input gate, respectively.

Bi-LSTM is a combination of a forward LSTM and a backward LSTM. The common forward LSTM has a sequence of information processing. Usually, it only considers the preceding text and ignores the following, so it cannot synthesize the context information to output. The Bi-LSTM structure can obtain enough context information, and both models have a common output layer, as shown in Figure 3.

3.3. Self-Attention Mechanism. In the process of spam detection, we usually faced the situation about the number of texts is limited, especially for the situation that the content information of different users' tweets varies greatly, it is difficult to obtain more effective semantic information. However, through the comparison, it is found that some key words in tweets can help to identify the types of tweets more quickly. For example, words such as "promotion" and "discount" can help to quickly identify the advertising intention of the product tweeted by spammers. At the same time, different words play different roles in classification. Extracting key words helps optimize the feature extraction process. The introduction of attention mechanism can increase the efficiency and improve the classification accuracy. Compared with the attention mechanism, self-attention only computes attention within the sequence, looking for the internal connection of the sequence. The calculation formula is as follows:

$$\text{Attention}\left(Q, K, V\right) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{6}$$

where Q, K, V are three matrices obtained from the same input and different parameters. First, calculate the multiplication of Q, K matrix and divide by $\sqrt{d_k}$. Finally, the softmax operation is used to normalize the result into probability distribution and multiply it by matrix V to get the result. The structure of self-attention is shown in Figure 4.

4. The Proposed Model

In traditional spam detection tasks, it only needs to count the malicious words. However, with the continuous optimization of spam, spammer replaces malicious words with other words, but it can transmit the same information. We need a model which can better understand the sentence and pay attention to the order of words in the sentence, and the information expressed after they are related. ALBERT and Bi-LSTM can better understand the short text information and the association information between words. In the model, we propose a hybrid model with ALBERT, Bi-LSTM, and Self-attention. The model uses ALBERT to extract and

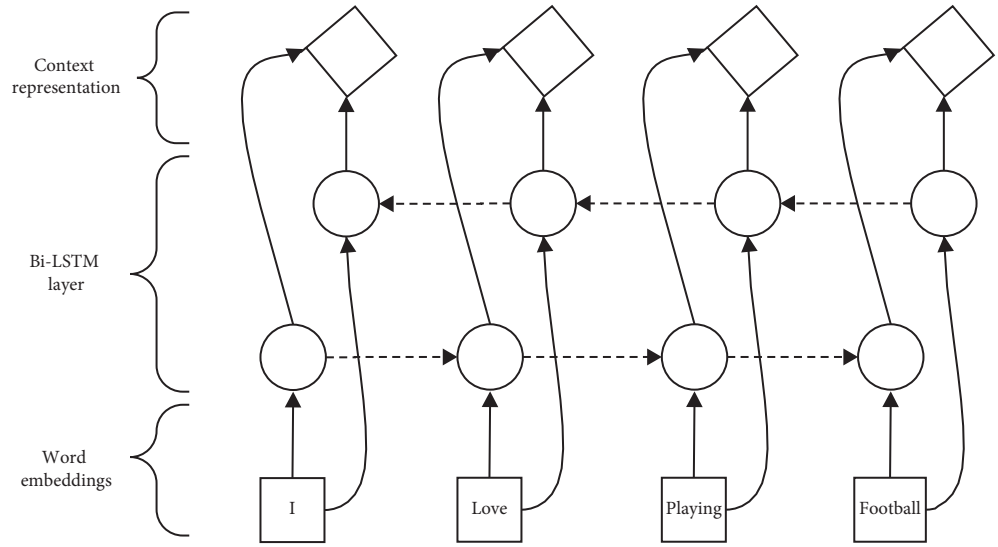

Figure 3: The structure of Bi-LSTM.

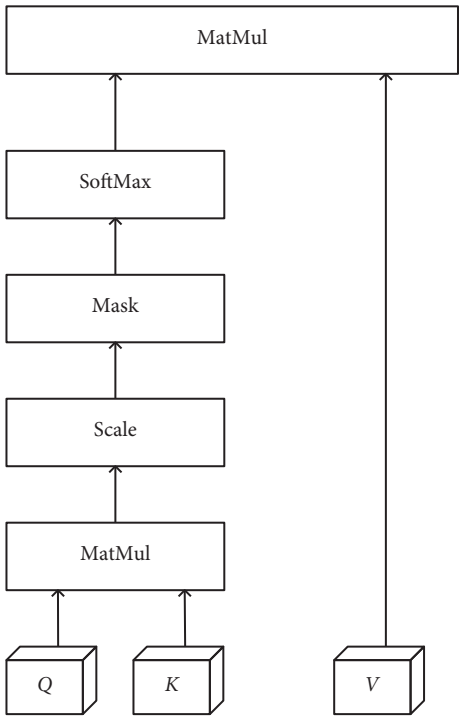

Figure 4: The structure of self-attention.

understand the semantic features of the original text for the first time. Self-attention is combined with Bi-LSTM to detect spam. Figure 5 shows the framework and steps of the model.

4.1. Embedding Layer. Before passing the data into the model, we need to perform some preprocessing operations on the data. For example, remove some stop words and delete emoticons. Then, the embedding layer will serialize the input preprocessed data and convert each word in the text data into a corresponding number in the dictionary. The feature representation will be output through the multilayer bidirectional transformer encoder. The formula is as follows:

$$T = (T_1, T_2, \ldots, T_{N-1}, T_N), \tag{7}$$

where T_i represents the feature vector of the i word in the text.

4.2. Bi-LSTM Self-Attention Layer. The ALBERT model has serialized the data and tried to understand the relationship between words. However, to understand a short text is a tough work for ALBERT. We use a combination of self-attention and Bi-LSTM to perform spam detection. In this layer, the text feature data output by the ALBERT layer will be trained, and the text features will be input into the forward LSTM and the backward LSTM, respectively; then, two text vector representations will be obtained, which will be calculated together to get the final output. Finally, we perform SoftMax normalization on the output result to get result.

5. Experiment

5.1. Dataset. There are two datasets for our experiment. One of the datasets is microblogPCU. It comes from the uci dataset website, which contains the user's basic attribute information (such as gender, number of fans, and number of followers) and the content posted by the user. In this paper, we extracted part of the posted content data. In order to simulate the real social network environment, we set the ratio of the number of spam and nonspam to 2 : 8. There are a total of 2000 data, including 1600 nonspam and 400 spam.

Another dataset is a self-collected Weibo dataset by us. The dataset contains the basic information and tweets of 985 users who have been labeled, of which 403 are marked spammers and 582 are nonspammers. It contains 95,385 Weibo twitters in total. We randomly selected 1000 Weibo

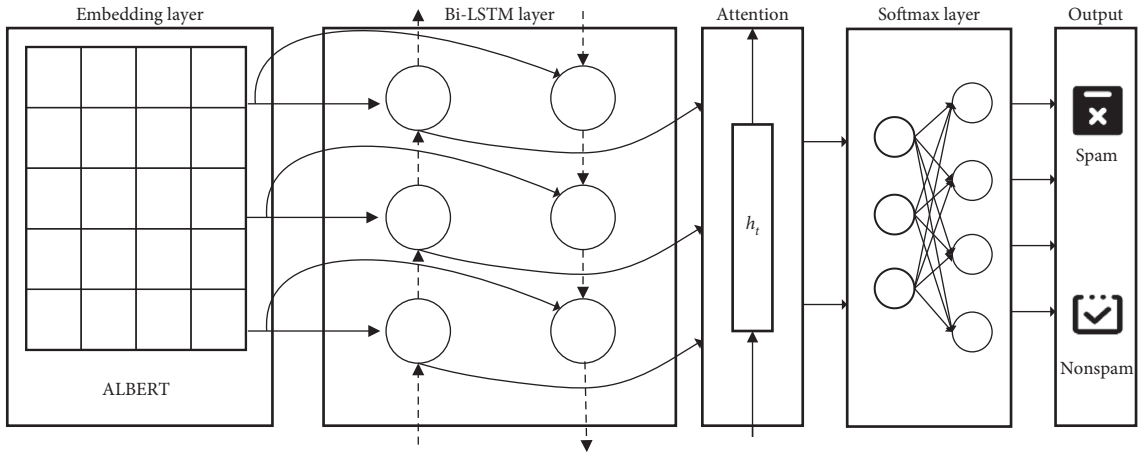

Figure 5: Spam detection model based on ALBERT and self-attention Bi-LSTM.

contents from spammer and nonspammer, respectively, checked, and labeled each one.

5.2. Input Preparation. The input to Bi-LSTM network is the user's Weibo text, but the length of the text is inconsistent. However, the network we built can only accept fixed length text, so we first analysed the length of the input network text, as shown in Table 1. We take 75% of the length quantile 80 as the maximum length value of the input model. For texts with length less than 80, we perform filling operation, and for texts with length greater than 80, we clear the subsequent text. In this way, the user's Weibo content will be retained and will not produce excessively filled meaningless content. And the model will also get enough text to extract semantic information. The processed text is then entered into the model.

5.3. Evaluation Metrics. We calculate precision, recall, F_1-score, and accuracy to judge the performance of the model. Accuracy is the probability of spam in the selected spam text, recall rate is the probability of correctly predicting as spam, F_1-score is the harmonic mean of precision and recall, and accuracy is the rate that the predicted correct text accounts for the total text. The formula is as follows:

$$\begin{aligned} \text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{recall} &= \text{TPTP} + \text{FN}, \\ F_1 - \text{score} &= 2 \times \text{precision} \times \text{recall}\,\text{precision} + \text{recall}. \end{aligned} \tag{8}$$

The relationships among TP, FP, FN, and TN are given in Table 2.

6. Results and Analysis

In this section, we use 2 datasets to verify the effectiveness of the model and compared it with other approaches. The experimental results are presented in the form of Table 3 and Figure 6 for dataset weiboData and Table 4 and Figure 7 for dataset microblogPCU. In the experiment, we use logistic regression, naive Bayes, and SVM combined with ALBERT to construct the spam detection model, respectively, and prove the superiority of LSTM neural network in this task. The results show that all the methods of constructing neural networks using LSTM performance better than the traditional methods of machine learning. From Tables 3 and 4, it shows the performance of the microblogPCU dataset is better than the weiboData dataset, which may be related to the ratio of spam to nonspam in the dataset.

Table 1: Analysis of user tweet length.

Measurement method	Value
Mean	58.5
Std	36.6
Min	7
25% of text length quantiles	27
50% of text length quantiles	54
75% of text length quantiles	**80**
Max	157

Table 2: Confusion matrix.

		Predicted	
		Spam	Ham
Actual	Spam	TP	FN
	Ham	FP	TN

Table 3: The results of model experiments (weiboData).

Model	Precision	Recall	F_1
AB + LR	0.868	0.829	0.848
AB + NB	0.810	0.814	0.811
AB + SVM	0.846	0.835	0.840
W2V + Bi-LSTM	0.844	0.841	0.842
AB + Bi-LSTM	0.889	0.862	0.875
AB + SA Bi-LSTM	**0.903**	**0.863**	**0.882**

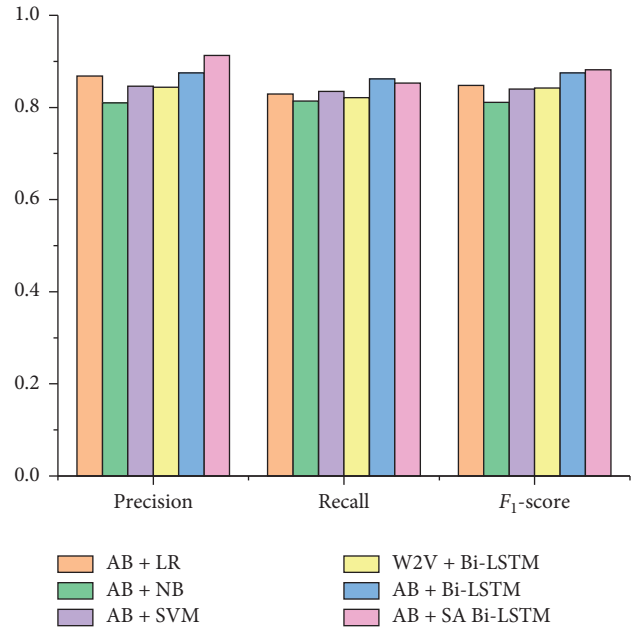

Figure 6: Comparison of model experiment results (weiboData).

Table 4: The results of model experiments (microblogPCU).

Model	Precision	Recall	F_1
AB + LR	0.865	0.821	0.842
AB + NB	0.817	0.826	0.821
AB + SVM	0.861	0.842	0.850
W2V + Bi-LSTM	0.850	0.853	0.851
AB + Bi-LSTM	0.899	0.870	0.884
AB + SA Bi-LSTM	**0.912**	**0.891**	**0.901**

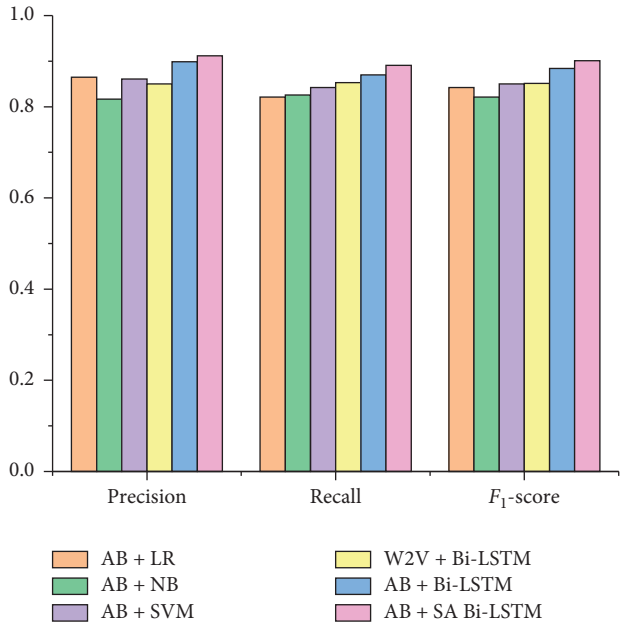

Figure 7: Comparison of model experiment results (microblogPCU).

To select the leading model in word embedding, word2vec (W2V) and Bi-LSTM are used to build the model. In Figures 6 and 7, it can be seen that ALBERT + Bi-LSTM is 2%–4% ahead of word2vec + Bi-LSTM in precision and F_1-score.

Finally, the effectiveness of self-attention is proved by comparing the AB + Bi-LSTM model and others. The experimental results are shown in Table 3. AB + SA Bi-LSTM is all ahead of other comparison models in the results and is 1% ~ 2% ahead of AB + Bi-LSTM in precision.

Figures 8 and 9 show the performance comparison of the word2vec (W2V) Bi-LSTM, ALBERT (AB) Bi-LSTM, and ALBERT + self-attention (AB + SA) Bi-LSTM. When the dataset size is the same and the parameters are set to the same value, the accuracy of the model rises with the increase of epoch. Figure 8 shows that the accuracy of W2V Bi-LSTM increases steadily with rounds before the epoch 10, fluctuates between the epoch 10–28. The accuracy of AB Bi-LSTM has been fluctuating and rising. The accuracy of the AB + SA LSTM model also increases with the increase of epoch and is stable higher than the values of the other two models after epoch 20. It can be seen from Figure 8 that the performance of AB + SA Bi-LSTM is not excellent when there are fewer rounds. This may be related to the self-attention mechanism requiring more data features for reference. It is noticed that along with the increase of training epoch, the model we proposed gradually shows its superiority. Figure 9 shows that the accuracy of the microblogPCU dataset with the increase of epochs is close to the same as the weiboData dataset. The difference is that the AB + SA Bi-LSTM model leads the other two models when the epoch is 25 on the microblogPCU.

Figures 10 and 11 show the accuracy of the model as the amount of data raises, assuming that the training epoch is

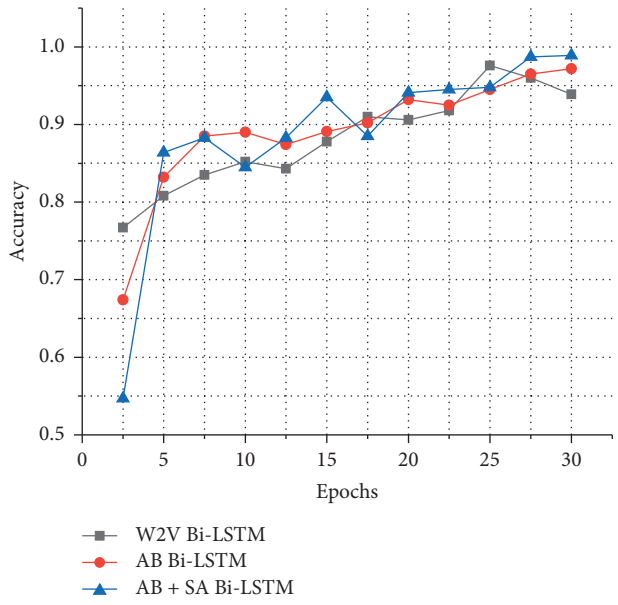

Figure 8: Accuracy as the number of epochs increases (weiboData).

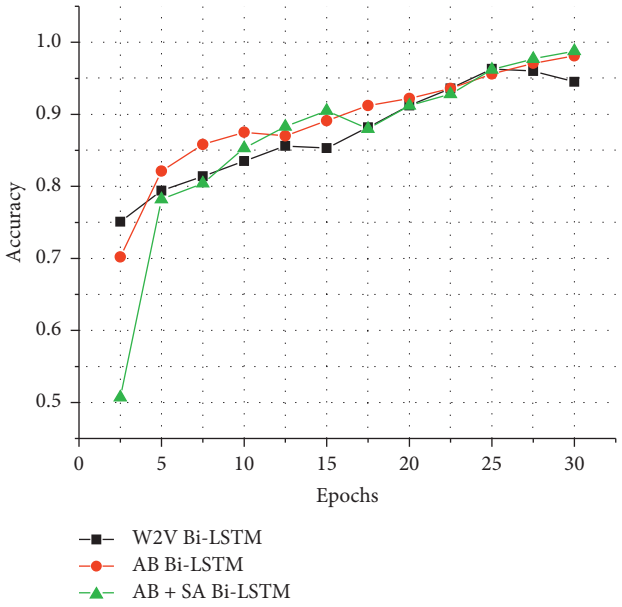

Figure 9: Accuracy as the number of epochs increases (microblogPCU).

fixed to a uniform size of 30. The figures show that the accuracy increases with the data size increasing. Figure 10 shows that the AB + SA Bi-LSTM model leads the other models at 1700 data volume, which shows that our proposed model has better performance when the data volume is larger. Figure 11 shows that the AB + SA Bi-LSTM model on the microblogPCU dataset always ahead of the other two models, which proves that the model is more sensitive to the quality of the dataset and outperforms other models on a better-quality dataset.

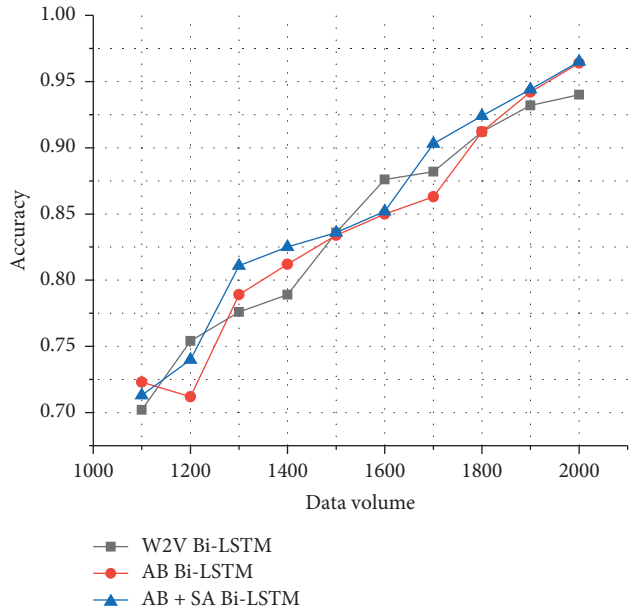

Figure 10: Accuracy as the data volume increases (weiboData).

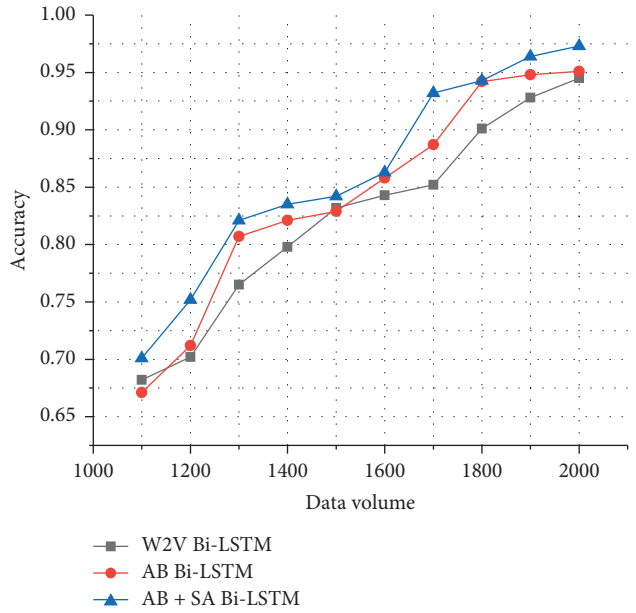

Figure 11: Accuracy as the data volume increases (microblogPCU).

7. Conclusion and Future Work

The detection of social network spam is a difficult task. The main problem in this field is that spammers have been upgrading and iterating spam text based on detection strategies, resulting in a decrease in the accuracy of detection. In response to this problem, we propose a detection method that fully considers the contextual information of text and, at the same time, take advantage of the self-attention mechanism for the shortness of text. We compared the difference between the machine learning methods with Bi-LSTM. Experiments show that Bi-LSTM performs better on spam detection task. At the same time, we also compared the effect of word2vec and ALBERT applied to the model. The experimental results show that ALBERT performs better because it considers more context information. We also proved the effectiveness of the self-attention mechanism in the model through comparative experiments. In general, according to the experimental results, our proposed model is 1% to 4% ahead of other models.

However, at the same time, the addition of the self-attention mechanism increases the computational time and

computational resources required by the model. In the future, we will continue to explore how to improve the efficiency of detection with the addition of a self-attention mechanism.

Data Availability

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

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation (grant nos. 61772099, 61772098, and 61802039); the Science and Technology Innovation Leadership Support Program of Chongqing (grant no. CSTCCXLJRC201917); the University Outstanding Achievements Transformation Funding Project of Chongqing (grant no. KJZH17116); the Innovation and Entrepreneurship Demonstration Team Cultivation Plan of Chongqing (grant nos. CSTC2017kjrc-cxcytd0063); and the Technology Innovation and Application Development Projects of Chongqing under grant nos. CSTC2019jscx-fxydX0086, cstc2019jscx-fxydX0089, and cstc2018jscx-mszd0132.

References

[1] G. Jain and M. Sharma, "Social media: a review," *Advances in Intelligent Systems and Computing*, vol. 433, pp. 387–395, 2016.

[2] J. f. Alqatawna, A. Madain, and A. M. Al-Zoubi, "Online social networks security: threats, attacks, and future directions," *Social Media Shaping E-Publishing and Academia*, pp. 121–132, 2017.

[3] S.-M. Al-Sayyed, W. C. Ao, and P.-Y. Chen, "On modeling malware propagation in generalized social networks," *IEEE Communications Letters*, vol. 15, no. 1, pp. 25–27, 2011.

[4] J. Chen and L. Bing, "Opinion spam and analysis," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pp. 219–230, Palo Alto, CA, USA, 2008.

[5] H. Atefeh, M. Tavakoli, N. Salim et al., "Detection of review spam: a survey," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3634–3642, 2015.

[6] Nexgate, *State of Social Media Spam Report*, https://go.proofpoint.com/%20nexgate-social-media-spam-research-report, 2013.

[7] T. Gupta, G. Choudhary, and V. Sharma, "A survey on the security of pervasive online social networks (POSNs)," *Journal of Internet Services and Information Security*, vol. 8, no. 2, pp. 48–86, 2018.

[8] A. Branitskiy, D. Levshun, N. Krasilnikova et al., "Determination of young generation's sensitivity to the destructive stimuli based on the information in social networks," *Journal of Internet Services and Information Security*, vol. 9, no. 3, pp. 1–20, 2019.

[9] M. Kolomeets, A. Benachour, D. E. Baz et al., "Reference architecture for social networks graph analysis," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 10, no. 4, pp. 109–125, 2019.

[10] G. Xu, X. Wu, and J. Liu, "A community detection method based on local optimization in social networks," *IEEE Network*, vol. 34, no. 4, pp. 42–48, 2020.

[11] S. Liu, J. H. Yim, H. J. Lee et al., "Semantic similarity calculation method using information contents-based edge weighting," *Journal of Internet Services and Information Security*, vol. 7, no. 1, pp. 40–53, 2017.

[12] G. Xu, W. Li, and J. Liu, "A social emotion classification approach using multi-model fusion," *Future Generation Computer Systems*, vol. 102, pp. 347–356, 2020.

[13] B. Enrico and A. Bryl, "A survey of learning-based techniques of email spam filtering," *Artificial Intelligence Review*, vol. 29, pp. 63–92, 2008.

[14] O. Amayri and N. Bouguila, "A study of spam filtering using support vector machines," *Artificial Intelligence Review*, vol. 34, no. 1, pp. 73–108, 2010.

[15] M. Vangelis, L. Androutsopoulos, and G. Paliouras, "Spam filtering with naive bayes-which naive bayes?" in *Proceedings of the Third Conference on Email and Anti-Spam (CEAS 2006)*, Mountain View, CA, USA, July 2006.

[16] M. Soiraya, S. Thanalerdmongkol, and C. Chantrapornchai, "Using a data mining approach: spam detection on Facebook," *International Journal of Computer Applications*, vol. 58, no. 13, pp. 26–31, 2012.

[17] C. Johnson, B. Khadka, R. B. Basnet et al., "Towards detecting and classifying malicious URLs using deep learning," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 11, no. 4, pp. 31–48, 2020.

[18] F. Ahmed and M. Abulaish, "An MCL-based approach for spam profile detection in online social networks," in *Proceedings of the 2012 IEEE 11th International Conference on IEEE Trust, Security and Privacy in Computing and Communications (TrustCom)*, Liverpool, UK, 2012.

[19] A. Zoubi, H. Faris, J. Alqatawna et al., "Evolving support vector machines using whale optimization algorithm for spam profiles detection on online social networks in different lingual contexts," *Knowledge Based Systems*, vol. 153, pp. 91–104, 2018.

[20] P. Tien and Z. Nur, "User identification via neural network based language models," *International Journal of Network Management*, vol. 29, no. 3, 2018.

[21] J. Ma, W. Gao, P. Mitra et al., "Detecting rumors from microblogs with recurrent neural networks," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Palo Alto, CA, USA, July 2016.

[22] G. Ruan and Y. Tan, "A three-layer back-propagation neural network for spam detection using artificial immune concentration," *Soft Computing*, vol. 14, no. 2, pp. 139–150, 2010.

[23] L. Z. Xiao, G. Z. Wang, and Y. Zuo, "Research on patent text classification based on Word2Vec and LSTM," in *Proceedings of the 2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, December 2018.

[24] Y. D. Luan and S. F. Lin, "Research on text classification based on CNN and LSTM," in *Proceedings of the 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Dalian, China, March 2019.

[25] Y. Dong, P. Liu, Z. Zhu, and Q. Zhang, "A fusion model-based label embedding and self-interaction attention for text classification," *IEEE Access*, vol. 8, pp. 30548–30559, 2020.

[26] D. Wang, M. W. Chang, L. Kenton et al., "BERT: pre-training of deep bidirectional transformers for language understanding," arXiv, 2017.
[27] V. Ashish, S. Noam, P. Niki et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.
[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 7, pp. 1735–1780, 1997.