Computer Life
ISSN: 1819-4818 | Vol. 11, No. 2, 2023

A Microblog Rumor Detection Model Incorporating Multivariate Features

Bai Li, Yujun Zhang *

School of Computer and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China
* **Corresponding author**: Yujun Zhang

Abstract: The wide spread of microblog rumors has seriously disturbed the network public order and greatly affected people's lives. The traditional microblog rumor detection model only focuses on semantic information and has insufficient generalization ability. Aiming at this problem, this study constructs a novel microblog rumor detection model by using rumor propagation pattern features and rumor propagation user features, combined with semantic information. The experimental results show that the model achieves an accuracy of 96.3% on the microblog rumor public dataset, and all other evaluation indexes also perform well.

Keywords: Weibo Rumors; Feature Fusion; Deep Learning.

1. Introduction

《The Blue Book of the World Internet Conference 2021》 points out that the number of Internet users in China has reached 1.011 billion, and the Internet penetration rate is 71.6%. The wide application of the Internet and the rapid development of related technologies have had a profound impact on people's way of learning, living and thinking. People's access to information has transitioned from traditional newspapers and television to today's social media platforms. Over the decades, many excellent social platforms have emerged at home and abroad. In China, after years of development, Sina Weibo has become one of the most popular social platforms for Chinese people. Social platforms such as Weibo have become a tool for people to obtain daily information and engage in communication. Due to the wide coverage of microblogging users, some users may post factually incorrect content, thus triggering microblogging rumors. If these rumors are not controlled and managed, they can trigger public panic and social unrest.

2. Domestic and International Research Status

The current research on rumor detection at home and abroad is mainly divided into two stages, and the previous stage mainly uses machine learning algorithms for rumor detection. With the development of arithmetic power, deep learning methods have become popular, and the research at this stage is mainly based on traditional neural networks and improved models of these networks for rumor detection.

Earlier rumor detection was done by manually extracted features and then using machine learning algorithms for rumor detection. Qazvinian et al[1] in 2011 used tweet content features , network features, and symbolic features to construct Bayesian classifiers for rumor detection. Castillo et al[2]. in 2011 proposed four types of features: content features, user features, topic features, and propagation features that Including features such as the length of the content, the number of symbols, the number of user followers, the percentage of tweets with hashtags, etc., a decision tree algorithm was used to carry out the rumor detection work.

With the development of arithmetic power, rumor detection at this stage is based on deep learning methods. In 2016, Ma et al[3] used recurrent neural networks to learn hidden representations to capture temporal changes in contextual information in relevant posts. Ajao et al[4] proposed a model combining convolutional neural networks and long-short-term memory networks to automatically extract semantic features of Twitter. real-time rumor detection was achieved.

3. A Microblog Rumor Detection Model Incorporating Multivariate Features

The rumor detection model is depicted in Figure 1. The first layer serves as the input layer, responsible for receiving the extracted ten features and the textual features of Weibo events. The second layer is the feature extraction layer, which is divided into two extraction methods. The first one extracts for user features and rumor propagation features; the second one extracts for text features of microblogging events. The third layer is to fuse the features extracted in both ways to form the multifusion features.The fourth layer is to feed the multivari -ate fusion features into the classification layer for rumor discrimination.

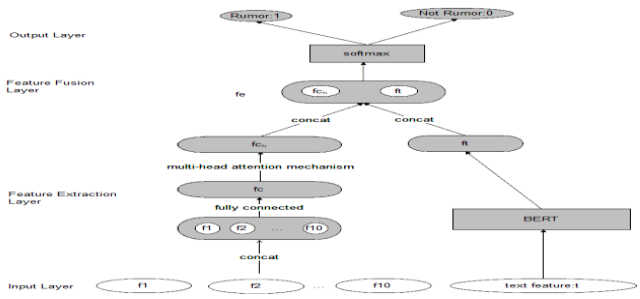

Fig.1 Rumor Detection Model Diagram

Input Layer

Studies have shown that there are some differences in the basic characteristics of rumor users and non-rumor users. For example, women are more likely to spread rumors than men, and non-verified users are more likely to spread rumors than

verified users. There is also a certain pattern in the spread of rumors. The life cycle of a rumor is divided into four stages, namely, the outbreak stage, the diffusion stage, the peak stage and the decline stage. 0-6 hours after the appearance of a rumor is the outbreak stage, 6-12 hours is the diffusion stage, 12-18 hours is the peak stage, and 18-24 hours is the decline stage. Based on these differences, a total of 10 relevant features are extracted in this paper, as shown in Table 1 below:

Table 1. Feature Table

Feature Descriptions	Feature Representations
Fans_count(f1)	Specific Value
Followers_count(f2)	Specific Value
Fans_count – Followers_count(f3)	Specific Value
Gender(f4)	m:0，f:1
Verified(f5)	no:0，yes:1
Statuses_count(f6)	Specific Value
Outbreak_count(f7)	Specific Value
Diffusion_count(f8)	Specific Value
Peak_count(f9)	Specific Value
Decline_count(f10)	Specific Value

Feature Extraction Layer

The feature extraction layer is divided into two types: text feature extraction, user features and propagation fe atures extraction. The process is shown in the followin g equation:

$$fc = sigmoid(W_c(concat(f1, f2, \dots, f10)) + b_c) \quad (1)$$

Where f1,…,f10 are the numerical values of the ten extracted features,W_c is the parameter matrix of the fully connected layer and b_c is the bias of the fully connected layer.

In order to emphasize the importance of different features, the fc vector obtained from the fully connected layer is passed through a multi-head attention mechanism.The specific process is illustrated by the following equations:

$$Q_i = fc * W_i^Q \quad (2)$$

$$K_i = fc * W_i^K \quad (3)$$

$$V_i = fc * W_i^V \quad (4)$$

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{dk}}\right)V \quad (5)$$

$$head_i = Attention(Q_i, K_i, V_i) \quad (6)$$

$$MultiHeadAttention = concat(hard_1, \dots, head_n)W^o \quad (7)$$

$$fc_h = MultiHeadAttention \quad (8)$$

Where W_i^Q, W_i^K, W_i^V are the parameter matrices of the ith head and n is the number of heads. fc_h is the extracted feature.

Text features are extracted using the pretrained model Bert.

$$ft = BERT(t) \quad (9)$$

Feature Fusion Layer

The two extracted features are fused.

$$fe = concat(ft, fc_h) \quad (10)$$

Classificition Layer

$$output = softmax(W_o * fe + b_o) \quad (11)$$

W_o is the parameter matrix of the fully connected layer and b_o is the bias of the fully connected layer.

4. Experiment

Dataset

The dataset used in this paper comes from a range of microblogging platform-based data collected and used in the paper by Ma et al[3]. The dataset treats the source tweet and the comments and retweets below it as a microblogging event and contains a total of 4664 microblogging events.

Experimental Evaluation Metrics

The model evaluation metrics used in this article are consistent with those used in previous literature, including accuracy, precision, recall, and F1 score. For binary classification problems, the predicted results and actual outcomes can exhibit the four scenarios as shown in Table 2 below:

Table 2 The outcomes of binary classification problems

Real situation	Predicted situation	
	Positive Prediction	Negative Prediction
Positive Actual	True Positive (TP)	False Negative (FN)
Negative Actual	False Positive (FP)	True Negative(TN)

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (12)$$

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (15)$$

Analysis Of Experimental Results

This article has designed the following several models for comparison, in order to verify the effectiveness of the proposed method in this article.

BERT: A BERT model is used and retrained to extract text features and feed them into a classifier for classification.

LSTM-1[3]:A layer of LSTM network is used to obtain text features and finally classification is done by softmax.

GRU-1[3]:The model replaces the recurrent units in traditional recurrent neural networks with gated units to capture the current features, as well as incorporate the features from the previous moment, and finally classify them by softmax.

MUL-BERT: The trained BERT model is used and fine-tuned for extracting text features, which are used to fuse with the ten features proposed in this paper, and the fused vectors are fed to the final classifier.

MUL-ATT-BERT: In order to better emphasize the importance of different features, the ten extracted features are fed into the multi-attention mechanism and then after fusion with the text features extracted by BERT, they are fed into the classifier.

Table 3 Comparision of Model Results

	Category	Accuracy	Precision	Recall	F1-score
LSTM-1	R	0.907	0.878	0.934	0.905
	N		0.938	0.882	0.909
GRU-1	R	0.914	0.916	0.911	0.913
	N		0.913	0.916	0.914
BERT	R	0.915	0.903	0.915	0.909
	N		0.926	0.915	0.921
MUL-BERT	R	0.954	0.946	0.963	0.954
	N		0.962	0.946	0.954
MUL-ATT-BERT	R	0.963	0.948	0.980	0.964
	N		0.979	0.945	0.961

The experimental results from the table show that using LSTM and GRU can partially extract text features, but the performance is not very good. Using the BERT model can effectively extract text features, achieving an accuracy of 91.5%. Fine-tuning the pre-trained BERT model with the 10 features proposed in this study further improves the accuracy to 95.4%, demonstrating the effectiveness of the features proposed in this research. Building upon the MUL-BERT model and incorporating a multi-head attention mechanism to focus on the importance of different features, the accuracy reaches 96.3%, with improvements in other evaluation metrics as well.

5. Conclusion

This study introduces a BERT-based multi-feature fusion model that effectively addresses the discrimination of rumors in Weibo events. By utilizing the BERT pre-trained model to extract features from Weibo text and incorporating ten additional features for assistance in detection, along with the implementation of a multi-head attention mechanism to capture the significance of various features, the challenge of limited Weibo text data is overcome. Through experimental comparisons, the proposed model in this study demonstrates performance enhancement.

Acknowledgements

The authors sincerely thank all friends who provided assistance.

References

[1] Qazvinian V, Rosengren E, Radev D, et al. Rumor has it: Identifying misinformation in microblogs [C]// Proceedings of the 2011 conference on empirical methods in natural language processing. 2011: 1589-1599.

[2] Castillo C, Mendoza M, Poblete B. Information credibility on twitter[C]//Proceedings of the 20th international conference on World wide web. 2011: 675-684.

[3] Ma J, Gao W, Mitra P, et al. Detecting rumors from microblogs with recurrent neural networks[J]. 2016.

[4] Ajao O, Bhowmik D, Zargari S. Fake news identification on twitter with hybrid cnn and rnn models[C]//Proceedings of the 9th international conference on social media and society. 2018: 226-230.

[5] Chen W, Zhang Y, Yeo C K, et al. Unsupervised rumor detection based on users' behaviors using neural networks[J]. Pattern Recognition Letters, 2018, 105: 226-233.