Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 11, No. 2, 2023

Fusion of Emotional Information for Rumor Detection Model

Bai Li, Yujun Zhang *

School of Computer and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China

* **Corresponding author**: Yujun Zhang

Abstract: With the development of technology, the platforms through which people acquire information have shifted from traditional sources such as television and newspapers to today's social media platforms. However, due to the openness of social media platforms, it is challenging to ensure the quality of information. If false information is not addressed promptly, it can adversely affect people's daily lives and lead to social panic. Previous research has largely focused on textual semantic information, which has raised concerns about its limited generalization ability. To address this issue, this study utilized a microblog sentiment analysis dataset to train a sentiment feature extraction model. This trained model was then used for extracting emotional features related to microblog rumors through transfer learning. These emotional features were subsequently integrated with the extracted semantic information features. Experimental results demonstrate that the model achieved an accuracy of 96% on a publicly available rumor detection dataset.

Keywords: Weibo Rumors; Sentiment Analysis; Deep Learning.

1. Introduction

In recent years, there have been many excellent social platforms both domestically and internationally. Internationally, there are social platforms like Twitter and Facebook, while domestically, there are platforms like Weibo and Douyin. Every day, a large number of users on each social platform share and exchange information. According to the fourth-quarter report released by Weibo in 2022, the platform had 586 million monthly active users and 252 million daily active users. However, due to the openness of social media platforms, the quality of users varies, and some users may post information that does not align with facts, creating rumors. These rumors can spread rapidly among users on social platforms. If these rumors are not promptly addressed, they can cause social panic and lead to financial losses for users.

2. Domestic and international research status

Early research on rumor detection often relied on manual feature extraction, including user characteristics, text statistics, and so on. Subsequently, machine learning algorithms were employed to classify rumor events based on these features. In 2011, Castillo et al[1] proposed four types of features: content features, user features, topic features, and propagation features. These features included attributes such as content length, the number of symbols, user followers, and the proportion of tagged tweets. They utilized a decision tree algorithm for conducting rumor detection work. In 2011, Qazvinian et al[2] utilized Twitter content features, network features, and symbol features to construct a Bayesian classifier for rumor detection.

With the advancement of computational power, deep learning methods have started to become popular. Most existing rumor detection work now predominantly utilizes deep learning methods, mainly focusing on two directions. First, it involves constructing deep models to extract more semantic information. Second, it leverages auxiliary tasks to assist in rumor detection. In 2019, Li et al[3] proposed the C-GRU model, which utilizes convolutional neural networks and gated recurrent units. This model utilizes convolutional neural networks to capture window feature representations and feeds different window feature representations into GRU to learn sequential contextual features of Weibo text. Finally, it performs event discrimination for Weibo content. In 2018, Ma et al[4] used stance detection tasks to assist in rumor detection tasks, employing parameter sharing to improve the accuracy of rumor detection.

3. Introduction to the rumor detection model

Research indicates that in the comments and retweets of microblog rumor events, the majority of content consists of negative information such as skepticism and fear. However, in non-rumor events, the content mostly consists of positive information that includes support and approval. Based on these differences, this paper first utilizes a microblog sentiment analysis dataset to train a sentiment analysis detection model. Then, it applies a transfer learning approach, utilizing a pretrained model to extract emotional features from the comments in the rumor dataset. Additionally, this paper will also utilize the pretrained BERT model to extract textual semantic information from Weibo events. These two different types of information will be combined and subsequently fed into the classification layer for rumor detection.

In the sentiment analysis model. Fasttext is a word-vector-based model developed by Facebook, renowned for its rapid training speed. Unlike traditional bag-of-words models, FastText takes into account subword information in vocabulary. It decomposes each word into character-level subwords and then averages or concatenates the vectors of these subwords to represent the entire sentence. This makes it perform better when dealing with rare or out-of-vocabulary words, making it suitable for informal Weibo text. Therefore, in this study, FastText will be used to obtain vector

representations of Weibo text. Additionally, to extract deeper semantic features, LSTM and GRU will also be employed in this paper. The overall model diagram for rumor detection is depicted in Fig. 1

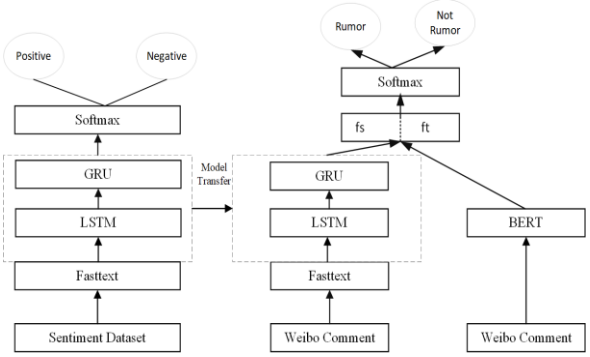

Fig. 1 Rumor Detection Model

(1) Sentiment analysis model

The sentiment analysis model utilizes FastText to vectorize the text, followed by the extraction of hidden features using LSTM and GRU. Finally, it is passed through Softmax for classification. The internal unit calculation process of LSTM is shown below:

$$f_t = sigmoid(W_f * [h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = sigmoid(W_i * [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = tanh(W_c * [h_{t-1}, x_t] + b_c) \quad (3)$$

$$o_t = sigmoid(W_o * [h_{t-1}, x_t] + b_o) \quad (4)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

$$h_t = o_t * tanh(C_t) \quad (6)$$

Where ft,it,ot are the forgotten gate, input gate and output gate at the current moment respectively. W is the weight matrix and b is the bias term. ht-1 is the hidden state of the previous moment, ht is the hidden state of the current moment, Ct is the memory state of the current moment, Ct-1 is the memory state of the previous moment, $\tilde{C}_t$ is the candidate memory state of the current moment, and xt is the input of the current moment.

The internal unit calculation process of GRU is shown below:

$$r_t = sigmoid(W_r * [h_{t-1}, x_t] + b_r) \quad (7)$$

$$z_t = sigmoid(W_z * [h_{t-1}, x_t] + b_z) \quad (8)$$

$$\tilde{h}_t = tanh(W_h * [r_t * h_{t-1}, x_t] + b_h) \quad (9)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (10)$$

Where rt,zt are the reset gate and update gate respectively, W is the weight matrix and b is the bias term., $\tilde{h}_t$ is the candidate hidden state, ht is the hidden state at the current moment, ht-1 is the hidden state at the previous moment, and xt is the input at the current moment.

The overall calculation process of the sentiment analysis model is as follows:

$$\text{fs} = \text{Fasttext}(\text{Sentiment Dataset}) \quad (11)$$

$$\text{output\_lstm} = \text{LSTM}(\text{fs}) \quad (12)$$

$$\text{output\_gru} = \text{GRU}(\text{output\_lstm}) \quad (13)$$

$$\text{output} = \text{Softmax}(\text{output\_gru}) \quad (14)$$

(2) Rumor detection model

First, the sentiment analysis model is used to extract the emotional inclination of the comments.

$$fs = SentiModel(Weibo\ Comment) \quad (15)$$

Secondly, semantic feature information from Weibo text is extracted through BERT.

$$ft = BERT(Weibo\ Comment) \quad (16)$$

Finally, the two types of vectors are merged and fed into the Softmax layer for classification.

$$output = Softmax(W_o * [fs, ft]) + b_o \quad (17)$$

4. Experiment

4.1. Dataset

The paper includes two datasets: one for sentiment analysis and another for rumor detection. To better support rumor detection tasks, this paper utilizes the "weibo_senti_100k" dataset for sentiment analysis. It consists of 119,988 data samples with a nearly 1:1 ratio of positive and negative sentiments. The rumor detection dataset is derived from a series of microblogging platform-based data collected and utilized in the research conducted by Ma et al. 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. Rumor events and non-rumor events are approximately in a 1:1 ratio.

4.2. 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 1 below:

Table 1. 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)

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (18)$$

$$Precision = \frac{TP}{TP + FP} \quad (19)$$

$$Recall = \frac{TP}{TP + FN} \quad (20)$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (21)$$

4.3. 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-1: Initialize BERT model parameters, conduct retraining, extract textual semantic features, and input them into a classifier for rumor detection.

BERT-2: Utilize a pre-trained BERT model, fine-tune it, extract textual semantic features, and subsequently feed them into a classifier for rumor detection.

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

GRU-1: 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.

BERT-1-SENTI: Employ a retrained BERT model to extract textual semantic information, employ a pre-trained sentiment analysis model to extract emotional features, merge these two types of features, and input them into a classification layer.

BERT-2-SENTI: Utilize a pre-trained BERT model to

extract textual semantic information, employ a pre-trained sentiment analysis model to extract emotional features, combine these two types of features, and input them into a classification layer.

Table 2. 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-1	R	0.915	0.903	0.915	0.909
	N		0.926	0.915	0.921
BERT-1-SENTI	R	0.917	0.935	0.894	0.914
	N		0.901	0.939	0.920
BERT-2	R	0.953	0.947	0.961	0.954
	N		0.961	0.945	0.953
BERT-2-SENTI	R	0.960	0.952	0.969	0.960
	N		0.968	0.951	0.959

The experimental results indicate that using LSTM and GRU can partially extract text features, but the performance is not very satisfactory. Whether retraining the BERT model from scratch or fine-tuning a pre-trained BERT model, both methods effectively extract deep-level textual features and result in improved model performance. Comparing the two sets of models, BERT-1, BERT-1-SENTI, and BERT-2, BERT-2-SENTI, after introducing sentiment information features, there was a respective increase in accuracy by 0.2% and 0.7%. This study has demonstrated the effectiveness of introducing sentiment features. Among them, fine-tuning a pre-trained BERT model yielded the best results, with an accuracy of 96%.

5. Conclusion

This study introduces a rumor detection model that incorporates sentiment features, addressing the issue of limited rumor detection data. Through experimental comparisons, it has been demonstrated that sentiment features are effective, resulting in improved model performance.

Acknowledgements

The authors sincerely thank all friends who provided assistance.

References

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

[2] 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.

[3] Lizhao Li,Guoyong Cai,Jiao Pan. Microblog Rumor Event Detection Method Based on C-GRU(in Chinese) [J]. Journal of Shandong University (Engineering Edition),2019,49(02):102-106+115.

[4] Ma J, Gao W, Wong K F. Detect rumor and stance jointly by neural multi-task learning[C]//Companion proceedings of the the web conference 2018. 2018: 585-593.

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