Design and Implementation of Automatic Rumor Detection System Based on Opposite Meaning Searching

Haori Lu[1], Jingrong Wang[1], Jiazhen Song[2], Yutong Li[1], and Peng Nie[3](✉)

[1] College of Cyber Science, Nankai University, Tianjin 300350, China
{1913106,1911569,1911551}@mail.nankai.edu.cn

[2] College of Computer Science, Nankai University, Tianjin 300350, China
2013904@mail.nankai.edu.cn

[3] College of Cyber Science & Tianjin Key Laboratory of Network and Data Security Technology, Nankai University, Tianjin 300350, China
niepeng@nankai.edu.cn

Abstract. With the development of social networks, information on social platforms is exploding. However, Plenty of rumors have also appeared on social platforms, which significantly affect social stability. To reduce the harm of rumors, we design and implement an automatic rumor detection system. In this system, we propose a new rumor detection method based on opposite meaning searching. This method constantly searches pairs of messages with opposite meaning in same event, and conducts rumor detection on them. The core of our system consists of two modules: content understanding module and opposition analysis module. In the content understanding module, the system extracts summaries and keywords of messages and stores them in the database to enhance the subsequent detection effectiveness of the system. In the opposition analysis module, the system judges whether the meanings of the summaries are opposing and finds the message that is a rumor among the pair of messages with opposite meanings. If a pair of messages from the same event have opposite meanings, one of them is very likely to be a rumor, so by finding a pair of messages with opposite meanings, the system can detect rumors more efficiently and accurately, and the detection results are more interpretable.

Keywords: Rumor Detection · Opposite Meaning Searching · Text Summarization · Keywords Extraction

1 Introduction

In recent years, with the rapid development of the Internet, social networks are also being widely used by more and more people. As of December 2021, the number of Chinese Internet users has reached 1.032 billion, with an Internet penetration rate of 73.0% [3]. The huge number of users means an explosion of information growth in social platforms. However, in an Internet environment where everyone can say whatever they want, the quality of information on social platforms cannot be guaranteed. Rumors can easily spread on social platforms. According to a report from China [21], more than

Z. Zhan et al. (Eds.): EIMSS 2022, AHCS 7, pp. 783–791, 2023.
https://doi.org/10.2991/978-94-6463-024-4_82

a third of the top stories on Weibo contain fake information. The spread of rumors can increase social instability and be a great danger to society [20]. Therefore, it is of great importance to research the method of rumor detection.

In previous work, the detection of rumors is generally implemented by a deep learning model [2, 8, 9, 19]. The detection process of this model is also often limited to focusing only on the content features of the message itself [4]. With this model, each message needs to be fed into the model independently for detection, which reduces the efficiency of rumor detection, while the features of the message content are limited, resulting in a limited correct rate of the detection [7]. In this paper, we design and implement an automatic rumor detection system based on opposite meaning searching. It does not perform direct rumor detection for all messages, but finds two messages with mutually opposite sentences meaning in advance, and then detects them. Its validity is based on the assumption that if there are two opposite messages about the same event in a social network, then one of the opposite messages is very likely to be a rumor. Based on this assumption, our system categorizes messages on social platforms by event and continuously tries to search for semantically opposite pairs of messages for the same event, and then conducts rumor detection on these two messages.

2 System Design

2.1 Overview of the System

In our design, the system is designed to be used by social platforms. The social platforms submit messages posted by users on their platforms over a period of time to the system through the front-end page, and the system uses its own database to detect the submitted messages completely and returns the detection results to the social platforms. As shown in Fig. 1, the system can be divided into two parts: the content understanding module and the opposition analysis module.

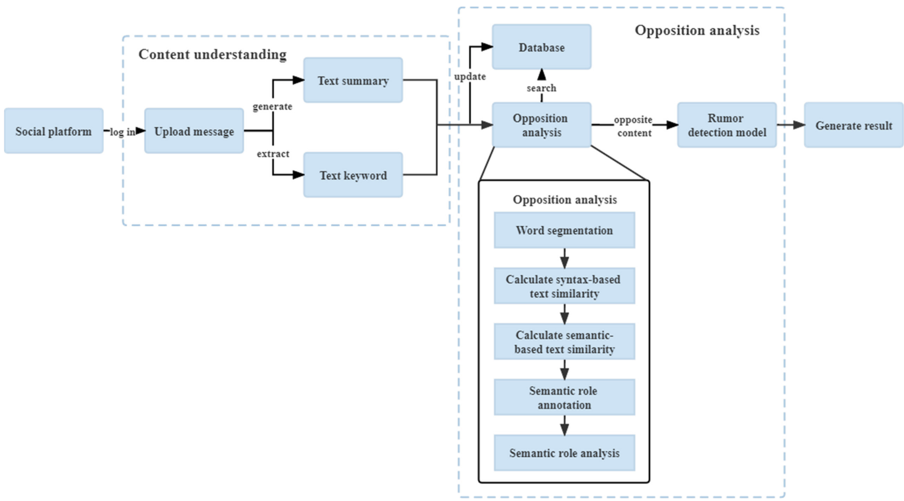

Fig. 1. System framework

2.2 Content Understanding Module

In the content understanding module, the main function to be implemented in the system is to understand and summarize the content of the input original message so that the subsequent opposition analysis module can work properly and efficiently. From the complete architecture of the system, we can see that the purpose of content comprehension is to improve the accuracy of judging whether a pair of messages has opposite syntactic meaning. In our system, judging whether a pair of messages has opposite meanings is mainly implemented by calculating text similarity and analyzing semantic roles. In social networks, message contents are usually rich and the syntactic structure is complex, so if the text similarity is calculated by the original two messages directly, the result will be disturbed by some insignificant details. Therefore, it is necessary to extract the summary of the original text, and consider the meaning that the summary wants to express as the meaning of the original text. In addition, the number of messages on social networks is very large, and if every two texts have to be compared in terms of sentence meaning, it will bring unacceptable time consumption and reduce the accuracy of the model. In the system, we restrict the scope of message pairwise comparisons to the same event and use keywords as a method to determine that the message content belongs to a certain event. Only if two messages have almost the same keywords, these two messages can be considered to describe the same event.

Therefore, we designed the content understanding module with two functions, namely, generating text summary and extracting text keywords. As shown in Fig. 2, these two functions do not depend on each other and can be run in parallel, thus improving the efficiency of the model. In the text summary generation part, we use Transformer [1] to generate a generative summary, which takes the Original message as input, understands and analyzes the text meaning, and generates a summary. In the keyword extraction part, we use the Textrank algorithm [10] to efficiently extract the keywords from the text. After obtaining the text summary and keywords, we store both together with the original message in a database table specifically for storing the original message, text summary, and keywords for future use in the opposition analysis module.

2.3 Opposition Analysis Module

In this module, we focus on the implementation of determining whether the respective meanings of a pair of messages are opposite to each other. Specifically, the logical process of determining whether a pair of messages has opposite meanings in the system is as follows.

1) Judging whether two messages describe the same event details. Limiting the scope of comparison only by the extracted keywords in the content understanding module can only ensure that the messages belong to the same event, but the single event also has a wide range of details, and it is meaningless to compare the meanings of two messages oriented to different details.
2) Determine whether the respective meanings of two messages are similar. If the meanings of the two messages are similar, it can be concluded that the meanings of the two messages are not opposite. However, if the semantics are not similar, because

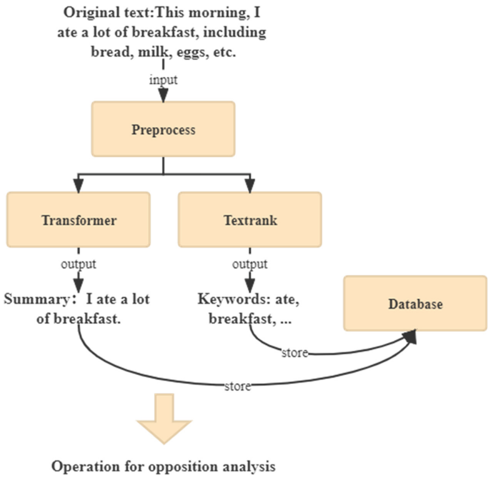

Fig. 2. Content understanding module structure

there is an inaccuracy in the preceding model, it cannot be directly determined that the meanings are opposite, and the semantic role analysis is needed to further determine.

3) Semantic role analysis. The system divides the semantic roles of the two messages and obtains the semantic role composition of each of the two messages, such as subject, object, predicate, etc. The system compares the semantic roles of the two messages, and only when they fit the rules set in the system can the two messages be considered as having opposite meanings.

After finding a pair of messages with opposite meanings, the system inputs the pair of messages into a rumor detection model, obtains the probability that each of the two messages is a rumor, and considers the message with the higher probability as a rumor, clears the data of the rumor message from the database and generates the detection result.

2.4 Design for Multiple Social Platforms

In our design, the system is not limited in connecting to a single social platform, but accepts messages from multiple social platforms. Therefore, the system is not designed for the citizens of a single social platform, but for all social platforms. Social platform administrators log in on the front-end page of the system and upload the message data of users on their platforms for the system to detect. The reason for the multi-platform design of the system is to expand the scope of data acceptance and enrich the content of the system's internal database. The portraits of users on different social platforms are different, thus the content posted on different platforms also has differences. Such differences will be more conducive to the search of opposite meaning message pairs and have a great improvement on the ability of rumor detection.

3 Implementation of the System

3.1 Content Understanding Module

According to the design of the content understanding module, in the concrete implementation, we need to implement the two functions of text summarization and keywords extraction by two models respectively. These two models process the messages in parallel in the framework we designed.

When implementing the text summarization function, we choose a generative summarization model in order to ensure the generalization ability and the relative uniformity of the syntactic structure of the summary [18]. Specifically, we use a Transformer model to implement this seq2seq transformation from the original message to the message summary. We set the model parameters of the Transformer: the number of encoder and decoder layers is 12, the output depth of encoder and decoder is 768, the number of heads of multi-headed attention is 12, and the depth of the hidden layer of the feedforward network is 3072. To avoid overfitting, the dropout method is also added to our model, and the dropout rate is set to set to 0.1. The Adam optimizer was used to optimize the parameters in the model training. Finally, we set the maximum length of the output summary to 36 to ensure that the summary content is short enough.

In the implementation of the keywords extraction function, we use the Textrank algorithm, which is inspired by the PageRank algorithm [11]. It constructs the text as a directed graph and calculates the importance of each node in the graph. Specifically for the keywords extraction task, the system divides the original message into words through the Jieba library [14] in Python. Each word after word segmentation is then acted as a node of the directed graph. The algorithm exploits the idea of n-gram to define edges, which means that the node of a word has only two opposite directional edges with the n-words in its vicinity. After constructing the directed graph, the system iteratively calculates the importance of nodes using Eq. (1), where $S(V_i)$ denotes the importance of node V_i, which can be initially set to 1. d is the damping factor and is set to 0.85. $In(V_i)$ denotes the node corresponding to the in-degree of node V_i, and $Out(V_j)$ denotes the point corresponding to the out-degree of node V_j in the directed graph. After the algorithm finishes running, the system selects the $n(n = 5)$ node words with the highest importance as the keywords of the message.

$$S(V_i) = (1 - d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \quad (1)$$

3.2 Opposition Analysis Module

As shown in Fig. 1, in the opposition analysis module, the system needs to implement syntax-based text similarity calculation, semantic-based text similarity calculation, semantic role annotation and set the rules that semantic roles should satisfy.

In the syntax-based text similarity calculation, we call the function about calculating text similarity in the xmnlp library [17] in Python. It can represent the original text as a sentence vector using the model in the library. The system then calculates the cosine similarity for each of the two text's sentence vector as the syntax-based similarity of the two texts [16].

For the implementation of the semantic-based text similarity computation and semantic role annotation, the system directly calls the relevant functions in HandLP library [6] in Python, which is a multilingual natural language processing toolkit for production environments that pre-trains dozens of models on more than a dozen tasks. With the models in this library, the system can accurately and efficiently implement the tasks of semantic-based text similarity computation [13] and semantic role annotation [12].

After finishing semantic role annotation, the words in the original text are divided into different roles, such as subject, object, predicate, gerund, etc. In order to improve the accuracy of the opposition analysis module, the system sets a series of restriction rules for the semantic roles of messages with opposite meanings, such as the need to ensure that the subject of the two messages is similar, or the object is similar, etc. Only when the semantic role rules set by the system are satisfied, the pair of messages can be considered to be of opposite meanings.

In the opposing analysis module, the above mentioned functions are executed sequentially to judge whether a pair of messages have opposite meanings. If a pair of messages has opposite meanings, they are fed into the rumor detection model [15] to obtain the probabilities that each is a rumor, and the message with the higher probability is considered to be a rumor. Following is the pseudo code of the whole module.

```
Begin
Sim_on_Structure = xmnlp.CalSimilarity(text1, text2)
 if Sim_on_Structure > threshold:
  Sim_on_Mean = hanlp.CalSimilarity(text1, text2)
  if Sim_on_Mean > threshold2:
   same_meaning_ops()
  else:
   s1, s2 = Semantic_roles_annotation(text1, text2)
   if meet_rules(s1, s2):
    rumor_detect(text1, text2)
   else:
    No_processing()
 else:
  No_processing()
End.
```

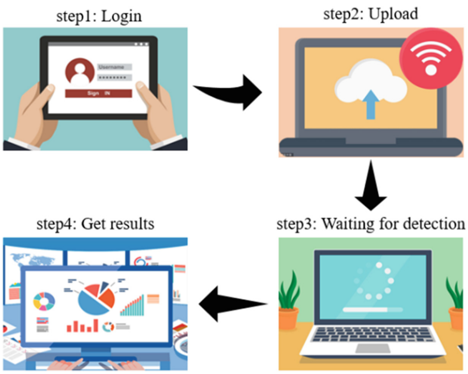

Fig. 3. Process of user operation

3.3 System Integration

After finishing the implementation of two important modules, we need to integrate the parts of the implementation into a complete system. When the modules are combined, we add control logic to the system to ensure a stable sequence of calls between modules. At the same time, we connect the system to MySQL database to implement the database storage of text summary and keywords. Finally, we built the front-end page for users to access through Vue.js framework. Users can submit the message data to be detected and get the detection results in the front-end page we built. Figure 3 shows the operation process of the user.

4 Conclusions

Currently, the widespread spread of rumors on social media has brought many negative effects to society [5]. In order to reduce the spread of rumors from the root, we designed and implemented an automatic rumor detection system based on opposite meaning searching by using various machine learning and deep learning models. The system contains two modules: content understanding module and opposition analysis module. In the content understanding module, our system extracts summaries and keywords from the original messages by two models in parallel in order to enhance the effectiveness of the subsequent module. In the contrastive analysis module, our system serially conducts similarity calculation and semantic role annotation on a pair of text summaries, and finally makes a judgment whether the meanings are opposite or not, and feeds the summary pair with opposite meanings into the rumor detection model. Through experiments, our system can implement the function of automatic rumor detection more accurately and has the practical significance.

In this paper, the effective operation of the system is based on the assumption that in social networks, if there are two opposite messages for the same event, then one of the opposite message pair is likely to be a rumor. Based on this assumption, our system constantly tries to find such a pair of messages with opposite meanings, and then conducts rumor detection on this pair of messages. While past rumor detection models tend to focus only on the content features of a message itself, our system introduces the influence of other messages with opposite meanings that exist in the social network. This is one of the biggest innovations of our system, which improves the accuracy and efficiency of rumor detection and makes rumor detection more automated, intelligent and interpretable.

Acknowledgements. The project is funded by National Key Project of China (No. 2020YFB1005700) and Undergraduate Education and Teaching Reform Project of Nankai University in 2020 (No. NKJG2020255). Besides, we would like to thank other bachelors for their help without reservation.

References

1. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Proceedings of the 31st international conference on neural information processing systems, NIPS'17. Curran Associates Inc., Red Hook
2. Cao J, Guo J, Li X, Jin Z, Guo H, Li J (2018) Automatic rumor detection on microblogs: a survey. arXiv preprint arXiv:1807.03505
3. CNNIC (2021) The 49th statistical report on the development of China's internet in 2021. R. China Internet Network Information Center
4. Gambhir M, Gupta V (2016) Recent automatic text summarization techniques: a survey. Artif Intell Rev 47(1):1–66. https://doi.org/10.1007/s10462-016-9475-9
5. Hashimoto T, Kuboyama T, Shirota Y (November 2011). Rumor analysis framework in social media. In: 2011 IEEE region 10 conference (TENCON 2011). IEEE, pp. 133–137
6. He H, Choi JD (2021) The stem cell hypothesis: dilemma behind multi-task learning with transformer encoders. In: Proceedings of the 2021 conference on empirical methods in natural language processing
7. Li Q, Zhang Q, Si L, Liu Y (2019) Rumor detection on social media: datasets, methods and opportunities. In: EMNLP-IJCNLP 2019, p 66
8. Ma J, Gao W, Wong KF (April 2018) Detect rumor and stance jointly by neural multi-task learning. In Companion proceedings of the web conference 2018, pp 585–593
9. Ma J, Gao W, Mitra P, Kwon S, Jansen BJ, Wong KF, Cha M (2016) Detecting rumors from microblogs with recurrent neural networks
10. Mihalcea R, Tarau P (2004) Textrank: bringing order into texts
11. Page L, Brin S, Motwani R, Winograd T (1998) The pagerank citation ranking: bringing order to the web. Stanford Digital Libraries Working Paper
12. Palmer M, Gildea D, Kingsbury P (2005) The proposition bank: an annotated corpus of semantic roles. Comput Linguist 31(1):71–106
13. Pu H, Fei G, Zhao H, Hu G, Jiao C, Xu Z (August 2017) Short text similarity calculation using semantic information. In: 2017 3rd international conference on big data computing and communications (BIGCOM). IEEE, pp 144–150
14. Sun J (2012) Jieba. Chinese word segmentation tool
15. Torshizi AS, Ghazikhani A (2019) Automatic Twitter rumor detection based on LSTM classifier. In: Grandinetti L, Mirtaheri SL, Shahbazian R (eds) TopHPC 2019, vol 891. CCIS. Springer, Cham, pp 291–300. https://doi.org/10.1007/978-3-030-33495-6_22
16. Wang J, Dong Y (2020) Measurement of text similarity: a survey. Information 11(9):421
17. Li X (2018) XMNLP: a lightweight Chinese natural language processing toolkit. https://github.com/SeanLee97/xmnlp

18. Zhang H, Cai J, Xu J, Wang J (November 2019) Pretraining-based natural language generation for text summarization. In: Proceedings of the 23rd conference on computational natural language learning (CoNLL), pp. 789–797
19. Zhang Q, Zhang S, Dong J, Xiong J, Cheng X (2015) Automatic detection of rumor on social network. In: Li J, Ji H, Zhao D, Feng Y (eds) NLPCC 2015, vol 9362. LNCS (LNAI). Springer, Cham, pp 113–122. https://doi.org/10.1007/978-3-319-25207-0_10
20. Zhang X, Ghorbani AA (2020) An overview of online fake news: characterization, detection, and discussion. Inf Process Manage 57(2):102025
21. Zhao Z, Resnick P, Mei Q (2015) Enquiring minds: early detection of rumors in social media from enquiry posts. In: Proceedings of the 24th international conference on world wide web. International World Wide Web Conferences Steering Committee, pp 1395–1405