Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 11, No. 1, 2023, 69-76.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Efficient Stance Ordering to Improve Rumor Veracity Detection

Zeinab Mohammadhosseini and Amir Jalaly Bidgoly*

Computer Department Qom University, Qom, Iran.

Article Info

Article History:
Received 07 June 2022
Revised 04 August 2022
Accepted 04 January 2023

DOI:10.22044/jadm.2023.11978.2345

Keywords:
Stance, Rumor Detection, Rumor Veracity, Ordering, Deep Learning.

**Corresponding author: jalaly@qom.ac.ir (A. Jalaly Bidgoly).*

Abstract

Social media is an inseparable part of human life, although the published information through social media is not always true. Rumors may spread easily and quickly in the social media, and hence, it is vital to have a tool for rumor veracity detection. Papers have already proved that the users' stance is an important tool for this goal. To the best knowledge of the authors, so far, no work has been proposed to study the ordering of the users' stances to achieve the best possible accuracy. In this work, we investigate the importance of the stances ordering in the efficiency of rumor veracity detection. This paper introduces a concept called trust for stance sequence ordering, and shows that proper definition of this function can significantly help to improve veracity detection. This work examines and compares different modes of definition of trust. Then by choosing the best possible definition, it is able to outperform state-of-the-art results on a well-known dataset in this field, namely SemEval 2019.

1. Introduction

Nowadays, social media and networks have been widely used around the world. People can communicate directly through media, and participate in the preparation and collection of information, and also help to spread the news or information. Social media have become very popular for various reasons such as the ease of collecting and sharing news, and the ability to spread information quickly in a split minute. Social posts or tweets share the news by thousands of users in a matter of hours. The users can also chat on social media. Now social media has the power to change the public opinion on any subject. However, such widespread use of social media without any supervision will definitely lead to negative consequences. One of these negative effects could be rumors of fake news. Discovering these phenomena is vital because in many cases professionals like journalists and institutions use social information and put people at serious risk. If we want to introduce rumor from the social science perspective, the following definitions are normally addressed:

• Petterson considers" rumor to be a long story that revolves around different people, and is related to events and topics that are important to the public" [1].

• Stern and Allport believe that "rumor is a chain of subjects that conveys a story and a message so that most of the details of this message disappear in the early stages of the chain" [2] With the rapid advancement of technology, rumors spread easier and faster in social media. Besides, the users who are dealing with rumors are still on increase with each passing day. For this reason, a new form of definition of rumor has been proposed with the approach of information science and technology, two examples of which are as follows:

• A rumor is defined as a sentence that true value is true, unverified or false. When the value of a rumor is false, some studies call it fake news [3].

• Rumor means a word, post or tweet that has spread, and is fundamentally incorrect. The target of the rumor is usually unknown [4]. The serious damages that incorrect information can cause to the society has made the scientific community pay much attention to the development of tools for detecting incorrect information and verifying correct information on the social media. The features used in the research area for rumor detection are as follows:

• **Textual features:** Textual features can include word statistics, word patterns or emotional words. The features can be the total number of letters and words, the number of distinct words and the average word length in a rumor text, the presence of a first-person pronoun or whether the message contains a link to an external source or not. Words that have special meanings and emotions are also important clues to describe the text. Question marks and exclamation marks are considered as the textual features in many works [5].

• **User Features:** The user features are derived from the user's social network. Rumors are created by multiple users, and spread by many users. Personal characteristics of each user can be used for rumor detection such as registration time, age, gender, occupation or number of followers, number of posts, location of posts, and user credentials [6].

• **Propagation of network features:** This class of features uses the characteristics of the propagation network to identify the unique features of gossip. Research has shown that these features can have very high power in detecting rumors. Number of nodes and links, average network density, number of comments, number of message republishing, and the relationship between message and the replied message, etc. are some instances of these features.

Depending on the function of the rumor detection model, a number of these features are used in rumor detection. The user features are useful in the early stages of publishing for detecting rumors. Most papers in the literature have used textual features in their research works. However, the user features have recently gained more interest to detect rumors. There are also other methods such as dynamic propagation [7], time series [8], recursive trees [9], and anomaly [10] for determining rumors. Lately, NLI[1] has been widely taken into account in the literature to determine the veracity of the rumors [11].

Stances of other users against a statement is one of the approaches that is recently gained much interest for detecting the veracity of rumors. Stance detection is the task of automatically determining the attitude towards a statement whether the is in favor, opposed or neutral to a target statement [12]. In rumor detection, a stance against the target may be one of the following:

• Support: The user is confirming the statement.

• Deny: The user denies the statement.

• Query: The user needs additional evidence.

• Comment: The user's response to the statement is not useful in determining the veracity of this statement [13].

Stance detection involves determining how responsive messages tend toward target messages. For example, if a tweet reply considered as definitely incorrect, the reply tweet stance compare to tweet target is undeniable. On the contrary, if you answer correctly, your stance is considered support. In social media, the users pin the opinions regarding the veracity of a tweet message that could lead to the collection of veracity stances against that message. For each tweet, a sequence of replies can be published, each one depicting a stance against the previous one. These tweets and replies can be modeled using a tree in which target tweet is the root, and reply tweets are the children of their own previous tweets. Each branch of this tree represents a path of stance against the target tweet. It is important to examine the whole branches to determine the veracity of a message. The researchers usually train an RNN[2] based model on the whole branches of trees [14]. On the other hand, research has already shown that the order of input to an RNN model affects its accuracy [15]. Detecting the rumor veracity is no exception. As far as we know, no work has been done to examine the appropriate ordering for the sequence of the mentioned branches. In this work, for the first time, an attempt has been made to identify the best way to increase the accuracy of news accuracy by examining different ordering. The obtained results outperformed the state-of-the-art methods, and the results also show that proper ordering can increase the accuracy of the model even more than the methods that use contextual information such as the propagation network.

This paper continues as what follows. In Section 2, the previous works are reviewed. Section 3 and Section 4 formally define the problem and the proposed method. In Section 5, several evaluations are performed in order to verify the performance of the proposed model in comparison with the relevant works. Lastly, Section 6 concludes the paper with the remarks and future works.

2. Related Works

Lately, stance analysis has been used to determine the veracity of rumors, and stance plays a

[1] Natural Language Inference

[2] Recurrent Neural Network

significant feature in predicting the veracity of the rumor. Dungs *et al*. used the hidden Markov model to show that the use of stances would improve the results of determining the veracity of rumors. Considering the user's stances along with other features as discussed in the introduction can lead to better determination of rumors [16, 33].

Shallow learning: In the recent years, many researchers have widely used shallow learning algorithms to classify rumors stances. One of the common methods that scholarly researchers such as Pamungkas *et al*. [17] have used to classify rumor stances is called the SVM[3] method. Another method is known as a decision tree as well as a random forest, which is normally expressed as a strong algorithm in the classification. Aker *et al*. employed this method to perform stance rumor classification, and finally reached the accuracy of 79% [18]. Additionally, logistic regression is another method that the Zubiaga's group has used along with textual features in order to classify rumor stances into two categories of agree and disagree [19]. Meanwhile, another shallow learning method for classifying rumor stances is the gradient boosting method, in which Bahuleyan *et al*. the mentioned algorithm along with textual features and obtained the F1 of 45% [20]. Bali and colleagues also used this method and classified the stances into four classes called agree, disagree, irrelevant, and discussion. They lastly reported the accuracy of 56% [21].

In an article published by Amiri *et al*. in 2022, they categorized the news related to Covid-19 by K-means method. The results of this article showed that people have less trust in health-related posts and the publication of these news leads to negative feelings [22].

Deep-learning: Deep learning is machine learning that solves complex tasks by identifying different patterns through experience, and is used to make predictions or decisions without being explicitly programmed to do so. According to the relevant papers in the open literature in the veracity of the rumors, the researchers have used methods based on LSTM[4], RNN, and a combination of them with textual features, propagation network feature, user, and timeframe. Some studies used LSTM neural networks to determine the veracity of the rumors. For instance, Kokina *et al*. achieved an f-score of 40.5% using this technique [1]. In another study, Kokina *et al*. using the same procedure along with textual features to enhance the results so that they obtained an accuracy of 48.5% [23]. Conforti *et al*. [24] and Ghanem *et al*. [4] used this approach to determine the veracity of the rumors. Gorrell *et al*. studied the LSTM neural networks, and classified rumors into three conventional groups: True, False, and Unverified, and they reached accuracy as 57.7% [12]. Along with this method, Li *et al*. investigated text, user, and propagation network features, and reached an F1-score of approximately 58% [13]. In another work proposed by this research group, they considered the user credibility to their previous method, and improved their previous f-score by 3% [25]. Here, some studies also used RNN neural networks in addition to the mentioned features to determine the veracity of the rumors. Ma and colleagues reached an f-score of 46.6% [26]. Interestingly, Islam *et al*. employed this method and obtained an f-score of 66% along with textual features, user and propagation network features [27]. Besides the methods discussed above, there are a few papers that used a combination of previous approaches with textual features, user, and propagation network features. As an example, Pouran et al. determined the rumors with an accuracy of 77% [2]. In another work, Annet Kendall reached an f-score of 58.8% in determining the accuracy of the rumors [28]. A number of studies have used other methods, Lee *et al*. investigated GCN[5] with time-based features to determine the veracity of the rumors and obtained an f-score of 59.9% [29]. In an article in 2019, Vosoughi *et al* .divided 126,000 stories from Twitter into two categories, true and false, and analyzed the speed and process of spreading this news. They found that false and true information are spread equally by robots but humans spread true information more than false information [30]. Giachanou *et al*. used emotional signals to detect fake news. Using an LSTM model, they used emotional signals extracted from the text of claims to distinguish between True and False claims and reached an f-score of 60% [31]. Ghanem *et al*. classified the stance of the rumor, and determined its truth. Their approach was based on stylistic, lexical, emotional, sentiment, meta-structural features and using Twitter data. Using this approach, they achieved significant accuracy in stance classification [32].

3. Problem Definition

Suppose we have t, and a set of reply tweets $r_1, r_2, ..., r_n$. Each reply tweet r_i is published in

[3] Support Vector Machine
[4] Long Short-Term Memory
[5] Graph Convolution Network

reply of another tweet. Let's define function $P(r_i)$ as the reply indicator function, where $r_j = P(r_i)$ indicates that r_i is published in reply to r_j; here we call r_j as the parent of r_i. Note that parent can also be the target tweet (i.e., $t = P(r_i)$), which means the tweet is in reply to the target tweet. The reply tweets form a conversation regarding the target tweet, and each one depicts a stance against either the target tweet or another reply tweet. The whole conversation can be modeled as a tree, in which the target tweet is the root, and r_i is the child of r_j if $r_j = P(r_i)$.

Figure 1 shows the tree of a sample conversation. The goal is predicting the veracity of the target tweet, using the reply tweets. It is usually done by training an RNN based model, on the different branches of the tree. Each branch is a path from to root to a leaf. For example, in Figure 1, there are four branches $b_1 = \{t_1, t_2, t_5\}$, $b_2 = \{t_1, t_3, t_6, t_9\}$, $b_3 = \{t_1, t_4, t_7\}$, $b_4 = \{t_1, t_4, t_8, t_{10}, t_{11}\}$.

Each branch creates a sequence of stances. Now the goal can be defined as how to train the RNN based model, on the sequences of branches, to achieve the maximum accuracy. As mentioned earlier, in the RNNs class of deep model, the ordering of the sequences strongly affects the performance of the model. The problem of this work is to examine different ordering methods and find the best one to achieve the highest accuracy in veracity prediction of the target tweet. Let us define $B(T)$ as the set of whole branches on T (i.e., the conversation tree); the problem can be formally defined as finding the best ordering function O on B to maximize the veracity prediction accuracy.

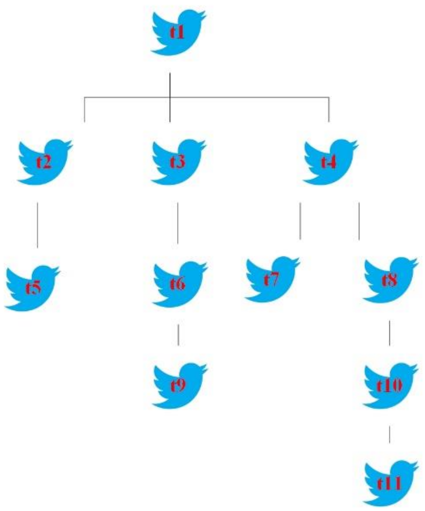

Figure 1. A schematic representation of the tree conversation of a tweet.

4. Proposed Method

A new model is presented to determine the veracity of the target tweet rumors by ordering the stance of tweet replies in the conversation tree. The inputs of this t target tweet model is the stance reply $B(T)$ which are sorted based on the O sorting function, and the user features of the target tweet. Here, *GloVe* and *BERT* embedding are used in order to vectorize the text of target tweet t. We have used LSTM as the most important model RNN-based and ordering stances along with the ordering function O, which is a trust function in this paper. More information will be described below about the word embedding layer.

The *GloVe* and *BERT* word embeddings are employed in this work. A two-way representation of target tweets is trained via this model. The BERT model produces the vector of each word, the previous and next words should, of course, be considered. This model has been used to create the text word vector of target tweets. In this work, the BERT model produces the vector of word of a sentence and if the input size is different, all inputs are padded to the largest sentence.

- LSTM Network for target tweet: As shown in Figure 2, after being embedded, input t is concatenated with the user's features, and will be given to the LSTM network. The LSTM layer is used for processing the sequence of embedding of target tweet's words and extracts textual based features. In this layer, the input to the LSTM network is the vector e_k, and h_k is the hidden state of the stage t. The output of the stage k is calculated as follows:

$$O_k = LSTM(e_k + h_k - 1) \quad (1)$$

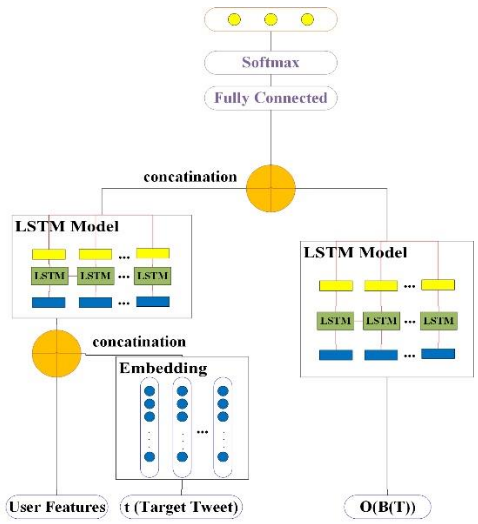

Figure 2. Proposed model structure.

- LSTM Network for branch of stance: In a T, different tweets will have different effects on the rumor's veracity. For instance, tweets that support or deny can be more effective in determining the veracity of a rumor. The input b_i that is a sequence from the $O(B(T))$ target tweet conversation tree is passed to these LSTM layers. Here, we have used two stacked LSTMs that are responsible for processing the sequence of stance tweets in a branch (i.e., t_i) and the branches themselves (i.e., b_i).
- Rumor verification layer: The output of the LSTM networks after concatenation is given to some dense layers to predict one of three states of *true*, *false* and *unverified* labels for the target tweet. The output of this layer is to determine the veracity of the target tweet rumor. The output of the model is acquired by softmax as follows:

$$Z = Concatination(OutputLstm_1, OutputLstm_2) \quad (2)$$

$$Label_{veracity} = Soft\max(Z) \quad (3)$$

Trust score: The ordering function in the proposed approach is so-called the trust function. This function defines the measure that is required for O to sort $B(T)$ based on. In these works, we have examined different features for making a trust function including: 1) retweet-count: the number of retweets of a source tweet, 2) follower-count: the number of followers, 3) friend-count: the number of friends, 4) favorite-count: the number of user favorites, 5) listed-count: the number of lists that the user is a member of, 6) user verified-count: user is valid or not in addition to the combinations of these features. For combination, all simple and weighted combination modes have been investigated. The results show that the best case of the trust function is the weighted averaging of all features based on the importance of each.

5. Experiments and Results

5.1. Data

The dataset, namely SemEval2019[6] is used throughout this work. This dataset includes 325 conversation trees. Table 1 reviews the dataset in numbers. Note that the dataset is collected based on 9 events. Moreover, each target tweet or each reply tweet has some textual features, user as well as network propagation features. Since the number of each class in the dataset is not homogenous, the F-score validation [22] is taken into account to verify the proposed idea, which is shown below:

$$\mathrm{Re}\,call = \frac{TP_i}{TP_i + FN_i} \quad (4)$$

$$\mathrm{Pr}\,ecision = \frac{TP_i}{TP_i + TF_i} \quad (5)$$

$$F - score_i = 2 \times \frac{\mathrm{Pr}\,ecision_i \times \mathrm{Re}\,call_i}{\mathrm{Pr}\,ecision_i + \mathrm{Re}\,call_i} \quad (6)$$

where TP_i, FP_i, and FN_i are, respectively, True Positive, False Positive, and False Negative samples in class *i*.

5.2. Tuning trust functions

Our goal is to find the best ordering function on set $B(T)$ In order to find the $O(B(T))$ function, a concept called trust function was defined here. Clearly, the trust function should be defined using a function of available fields. Thus, to find the best trust function, in this section, different definitions of trust functions are examined. Initially, the trust function is defined based on a single feature field in different modes and the achieved results are presented. Then we have examined the trust function defined based on the combination of all fields. In both cases, the results are presented and discussed to find out the best possible trust function to reach the highest possible accuracy.

Trust function using single feature: As mentioned earlier, the dataset includes a number of features such as retweet-count, friend-count, follower-count, favorite-count, listed-count, user verified-count. In the first try, we have examined all these features one by one to obtain the trust function. For each field, the function is tested in both ascending and descending orders. In order to obtain the proper trust function, different aggregation function such as average, sum and maximum are also examined. The results of these experiments are presented in Table 2. Whether in user verified account the user is verified or not, the maximum for each branch is equal to one. Due to this consideration, just sum and average of each branch are studied. Note that since both Glove and BERT are used as the embedding layers, each test was repeated twice for each of these embedding. As it can be seen in Table 2, in all features, the best outcomes are related to the descending order along with the use of BERT embedding. The best result is obtained for retweet-count with an F1 of 64.48%. Despite that, other features such as friend-count, favorite-count, follower-count, and

[6] https://figshare.com/articles/dataset/RumorEval_2019_data/8845580

user verified-count listed in other ranks, respectively.
Meanwhile, the worst score is found to be for listed-count feature. As discussed earlier, due to better results of BERT embedding in the descending order, just BERT embedding will be studied in the following experiment.

Table 1. Number of data using this work.

Data	Train	Validation	Test	All
Number of Tweets	63980	15995	26025	106000
Number of Trees	195	49	81	325
Number of Branches	2559	640	1041	4240

Trust function using multiple feature: After reviewing the results of different features, different combinations of the mentioned features are used to obtain the best possible trust function. In this section, two states of simple average and weighted average of features are considered. The results are presented in Table 3. Considering the results, it is found that the best $O(B(T))$ is the weighted average of the mentioned features, which is shown in following:

$$Trust_{function} = \frac{(6 \times ret_{count}) + (5 \times fri_{count}) + (4 \times fav_{count}) + (3 \times fall_{count}) + (2 \times user_{count}) + list_{count}}{21} \quad (7)$$

5.3. Performance comparison

In order to prove the reliability and efficiency of the proposed method, several comparisons with the following methods will be presented here:

1. A base model without any ordering is considered that receives the branch in no particular order (randomly) and predicts the veracity of the target tweet.
2. The Branch-LSTM model [1] that won the SemEval competition in task 8 in 2017 on determining rumor veracity and support for rumors and its proposed model have a similar structure to this work. It considers the relationship between target tweet and tweets response as a conversation tree but without any particular ordering.
3. MTL2 model [23] is a multi-tasking method that is able to perform two actions simultaneously and uses the LSTM branch as a shared task.
4. In user credibility model [13] the trust user is calculated based on user features, and propagation network. Subsequently, the veracity of tweet source is determined via the obtained trust user and user’s stances.
5. The GCN model [29], a time hierarchical model, to predict the veracity of a rumor using stances.

The results of the comparison of these works on SemEval2019 dataset are presented in Figure 3. As it can be seen, the proposed ordering method significantly improved the results and outperforms the previous results. Considering that our proposed method, just used textual features (i.e., text of target tweet and stance tweets), it is even better the models that uses contextual features such as user and propagation features. In order to provide a more complete comparison, the proposed method was evaluated once along with these features. The results of this evaluation that named “Ordering based on trust function + user’s features”, have even improved the previous results and reached 67% F-score in veracity prediction.

6. Conclusion and Future Works

Currently stance detection is one of the most important tools for rumor veracity prediction, and several methods have been already proposed to use these stances but no research has considered the importance of ordering these stances in increasing the accuracy of the model. In this work, for the first time, the importance of stance ordering in this field was shown and also the best available ordering method was presented. In the proposed method, a concept called trust for each user is defined, and then the stance sequence is sorted considering this function. The proposed method is able to outperform the previous results just taking the advantage of the textual features. Adding other types of features including user and propagation network features, even improved the results more, achieving the F-Score of 67.32% in” SemEval 2019” dataset. We can increase the results of our paper by adding emotional feature to our work in the future.

Table 2. Results obtained from ordering based on different features.

Models		Ascending Ave	Ascending Sum	Ascending Max	Descending Ave	Descending S	Descending Max
ret_{count}	Glove	51.8	53.08	53.21	53.43	55.62	56.56
	BERT	61.96	62.64	63.38	62.93	64.06	64.48
fri_{count}	Glove	51.85	53.43	53.83	52.78	54.67	55.6
	BERT	57.39	62.24	62.39	61.72	63.42	64.1
$foll_{count}$	Glove	50.8	51.25	53.23	52.89	53.44	54.5
	BERT	56.82	59.79	61.68	62.27	62.64	63.24
fav_{count}	Glove	48.94	50.67	52.02	49.11	52.1	52.44
	BERT	61.13	62.2	62.72	62.3	62.69	63.7
lis_{count}	Glove	50.4	51.67	53.43	51.23	52.7	54.67
	BERT	57.62	60.21	61.34	60.38	61.59	61.86
ver_{count}	Glove	49.3	50.42		50.31	51.08	
	BERT	60.27	61.47		61.45	62.5	

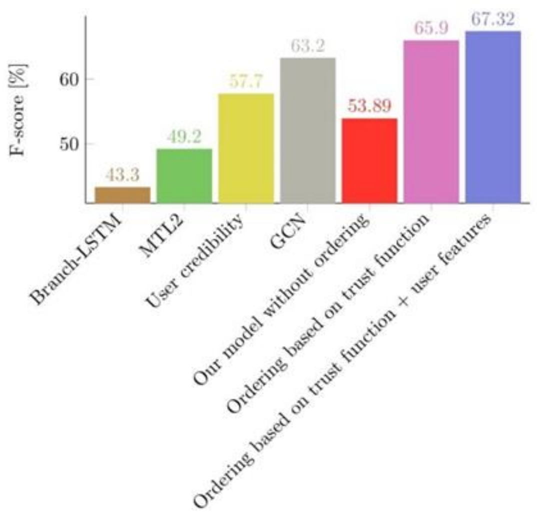

Figure 3. Comparison of the proposed work with state-of-the-art models.

Table 3. Results obtained from ordering based on different trust function.

Mean Type	F-Score %
Simple Average	64.57
Weighted Average	65.9

References

[1] E. Kochkina, M. Liakata, and I. Augenstein, "Turing at semeval-2017 task 8: Sequential approach to rumor stance classification with branch-lstm", *arXiv preprint arXiv:1704.07221*, 2017.

[2] A. P. B. Veyseh, J. Ebrahimi, D. Dou, and D. Lowd, "A temporal attentional model for rumor stance classification", *in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 2335–2338, 2017.

[3] A. Hanselowski, A. PVS, B. Schiller, F. Caspelherr, D. Chaudhuri, C. M. Meyer, and I. Gurevych, "A retrospective analysis of the fake news challenge stance detection task", *arXiv preprint arXiv:1806.05180*, 2018.

[4] B. Ghanem, P. Rosso, and F. Rangel, "Stance detection in fake news a combined feature representation", *in: Proceedings of the first workshop on fact extraction and VERification (FEVER)*, pp. 66–71, 2018.

[5] J. Cao, J. Guo, X. Li, Z. Jin, H. Guo, and J. Li, "Automatic rumor detection on microblogs: A survey", *arXiv preprint arXiv:1807.03505*, 2018.

[6] Q. Zhang, S. Zhang, J. Dong, J. Xiong, and X. Cheng, "Automatic detection of rumor on social network", *in: Natural Language Processing and Chinese Computing, Springer*, pp. 113–122, 2015.

[7] S. Wang, Q. Kong, Y. Wang, and L. Wang, "enhancing rumor detection in social media using dynamic propagation structures", *in: 2019 IEEE International Conference on Intelligence and Security Informatics (ISI), IEEE*, pp. 41–46, 2019.

[8] Z. Wang, Y. Guo, Z. Li, M. Tang, T. Qi, and J. Wang, "Research on microblog rumor events detection via dynamic time series based gru model", *in: ICC 2019 IEEE International Conference on Communications (ICC), IEEE*, pp. 1–6, 2019.

[9] J. Ma, W. Gao, and K.-F. Wong, "Rumor detection on twitter with tree-structured recursive neural networks", *Association for Computational Linguistics*, 2018.

[10] A. E. Fard, M. Mohammadi, S. Cunningham, and B. Van de Walle, "Rumor as an anomaly: rumor

detection with one-class classification", *in: 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), IEEE*, pp. 1–9, 2019.

[11] A. Bidgoly, H. Amirkhani, and F. Sadeghi, "Fake news detection on social media using a natural language inference approach", *in: springer/ Multimedia Tools and Applications*, 2020.

[12] G. Gorrell, K. Bontcheva, L. Derczynski, E. Kochkina, M. Liakata, and A. Zubiaga, "Rumoreval 2019: Determining rumor veracity and support for rumors", *arXiv preprint arXiv:1809.06683*, 2018.

[13] Q. Li, Q. Zhang, and L. Si, "eventai at semeval-2019 task 7: Rumor detection on social media by exploiting content, user credibility and propagation information", *in: Proceedings of the 13th International Workshop on Semantic Evaluation*, pp. 855–859, 2019.

[14] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets", *arXiv preprint arXiv:1511.06391*, 2015.

[15] L. Logeswaran, H. Lee, and D. Radev, "Sentence ordering using recurrent neural networks", 2016.

[16] S. Dungs, A. Aker, N. Fuhr, and K. Bontcheva, "Can rumor stance alone predict veracity", *in: Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3360–3370, 2018.

[17] E. W. Pamungkas, V. Basile, and V. Patti, "Stance classification for rumor analysis in twitter: Exploiting affective information and conversation structure", *arXiv preprint arXiv:1901.01911*, 2019.

[18] A. Aker, L. Derczynski, and K. Bontcheva, "Simple open stance classification for rumor analysis", *arXiv preprint arXiv:1708.05286*, 2017.

[19] A. Zubiaga, E. Kochkina, M. Liakata, R. Procter, and M. Lukasik, "Stance classification in rumors as a sequential task exploiting the tree structure of social media conversations", *arXiv preprint arXiv:1609.09028*, 2016.

[20] H. Bahuleyan and O. Vechtomova, Uwaterloo, "semeval-2017 task 8: Detecting stance towards rumors with topic independent features", *in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 461–464, 2017.

[21] R. Baly, M. Mohtarami, J. Glass, L. Marquez, A. Moschitti, and P. Nakov, "Integrating stance detection and fact checking in a unified corpus", *arXiv preprint arXiv:1804.08012*, 2018.

[22] Amiri, F., S. Abbasi, and M. Babaie Mohamadeh. "Clustering Methods to Analyze Social Media Posts during Coronavirus Pandemic", *in: Iran. Journal of AI and Data Mining,* vol. 10, no. 2, pp. 159-169, 2022.

[23] E. Kochkina, M. Liakata, and A. Zubiaga, "All-in-one: Multi-task learning for rumor verification', *arXiv preprint arXiv:1806.03713*, 2018.

[24] C. Conforti, M. T. Pilehvar, and N. Collier, "Towards automatic fake news detection: cross-level stance detection in news articles", *in: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pp. 40–49, 2018.

[25] Q. Li, Q. Zhang, and L. Si, "Rumor detection by exploiting user credibility information, attention and multi-task learning", *in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1173–1179, 2019.

[26] J. Ma, W. Gao, and K.-F. Wong, "Detect rumor and stance jointly by neural multi-task learning", *in: Companion proceedings of the the web conference 2018*, pp. 585–593, 2018.

[27] M. R. Islam, S. Muthiah, and N. Ramakrishnan, "Rumorsleuth: Joint detection of rumor veracity and user stance", *in: 2019 IEEE/ACM International Conference on Advances, in: Social Networks Analysis and Mining (ASONAM), IEEE*, pp. 131–136, 2019.

[28] A. Khandelwal, "Fine-tune longformer for jointly predicting rumor stance and veracity", *in: 8th ACM IKDD CODS and 26th COMAD*, pp. 10–19, 2021.

[29] A. P. B. Veyseh, M. T. Thai, T. H. Nguyen, and D. Dou, "Rumor detection in social networks via deep contextual modeling", *in: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 113–120, 2019.

[30] Vosoughi S., Roy D. and Aral S, "The Spread of True and False News Online. Science", pp. 1146–1151, 2019.

[31] Giachanou A., Rosso P., and Crestani F, "Leveraging Emotional Signals for Credibility Detection"; *In: Proc. of the 42nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '19), July 21–25, Paris, France*, 2019.

[32] Ghanem B., Cignarella A., Bosco C., Rosso P., and Rangel F, "UPV-28-UNITO at SemEval-2019 Task 7 Exploiting Post's Nesting and Syntax Information for Rumor Stance Classification". *In: Proc. of the 13th Int. Workshop on Semantic Evaluation (SemEval-2019), co-located with the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), Minneapolis, Minnesota, USA, June 6-7*, pp. 1125–1131, 2019.

[33] Sadeghi F, Jalaly Bidgoly A. "A survey of rumor detection methods in social networks". *Biannual Journal Monadi for Cyberspace Security* (AFTA). 2019 Sep 10;8(1):3-14.

اولویت‌بندی موضع در شبکه‌های اجتماعی برای تعیین صحت شایعات

زینب محمدحسینی و امیر جلالی بیدگلی*

دانشکده مهندسی، گروه مهندسی کامپیوتر، دانشگاه قم، قم، ایران.

ارسال ۲۰۲۲/۰۶/۰۷؛ بازنگری ۲۰۲۲/۰۸/۰۴؛ پذیرش ۲۰۲۳/۰۱/۰۴

چکیده:

امروزه اگرچه از طریق شبکه‌های اجتماعی و اینترنت می‌توان اطلاعات مفیدی به‌دست آورد اما در دنیایی که جدا کردن حقیقت از باطل دشوار است، شایعه به آسانی گسترش می‌یابد. در فرآیند تشخیص شایعات، کاربران و واکنش‌ها و مواضعی که نسبت به پیام یا خبر شایعه نشان می‌دهند، بسیار می‌تواند مثمر ثمر واقع شود. در این پژوهش برای تعیین صحت شایعات از اولویت‌بندی مواضع کاربران استفاده شده است. مجموعه داده استفاده شده در این پژوهش SemEval2019 می‌باشد و در این پژوهش اولویت‌بندی مواضع بر اساس ویژگی‌های موجود در این مجموعه‌داده انجام شده است. دنباله‌ی مواضع در هر درخت از توئیت منبع و پاسخ‌های آن بر اساس ویژگی‌های مختلف تعداد بازتوئیت‌ها، تعداد دوستان، تعداد دنبال‌شوندگان، تعداد علاقه‌مندی‌ها، تعداد لیست‌های معتبری که کاربران عضو آن هستند و تعداد کاربران معتبر در هر شاخه در این مجموعه داده مرتب شده و مدل، آموزش داده می‌شود. بهترین نتایج با استفاده از مرتب‌سازی مواضع بر اساس ویژگی بازتوئیت‌هاست که به F-Score برابر با ۶۴٫۴۸٪ رسیده است. سپس نمره اعتمادی بر اساس ویژگی‌های استفاده شده، محاسبه شده و بهترین F-Score در این حالت به ۶۵٫۹٪ رسیده است. در گام آخر این پژوهش از ویژگی‌های کاربر نیز در جهت بهبود نتایج استفاده شده که F-Score آن در این حالت برابر است با ۶۷٫۳۲٪. نتایج این نشان می‌دهد که اولویت‌بندی مواضع عملکرد مناسبی در تعیین صحت شایعات با استفاده از تشخیص موضع دارد و می‌تواند نتایج را به میزان حدود ۴٪ بهبود ببخشد.

کلمات کلیدی: شایعه، صحت شایعه، تعیین صحت شایعه، موضع، تشخیص موضع، اولویت‌بندی، ویژگی‌های کاربر.