A Graph-Theoretic Embedding-Based Approach for Rumor Detection in Twitter

Muhammad Abulaish
Department of Computer Science
South Asian University, New Delhi, India
abulaish@ieee.org

Nikita Kumari
Department of Computer Science
South Asian University, New Delhi, India
nikita.kashyap007@gmail.com

Mohd. Fazil
Department of Computer Science
South Asian University, New Delhi, India
mohdfazil.jmi@gmail.com

Basanta Kumar Singh
Department of Computer Science
South Asian University, New Delhi, India
basantasingh@students.sau.ac.in

ABSTRACT

In this paper, we present a graph-theoretic embedding-based approach to model user-generated contents in online social media for rumor detection. Starting with a small set of seed rumor words of four different lexical categories, we generate a words co-occurrence graph and apply centrality-based analysis to identify prominent rumor characterizing words. Thereafter, word embedding is applied to represent each category of seed words as numeric vectors and to train three different classification models for rumor detection. The performance of the proposed approach is empirically evaluated over two versions of a benchmark dataset. The proposed approach is also compared with one of the state-of-the-art methods for rumor detection and performs significantly better.

CCS CONCEPTS

• **Information systems** → **Data analytics**; • **Human-centered computing** → *Social network analysis*; • **Computing methodologies** → Supervised learning by classification.

KEYWORDS

Rumor detection, Social network analysis, Word embedding

1 INTRODUCTION

Nowadays, Online social networks (OSNs) are replacing standard news sources as a primary source of information. In most of the cases, the breaking news are often appearing first on social media before being broadcasted by the traditional media outlets. Due to the fast and mass-level dissemination capabilities of the OSNs, adversaries are exploiting them for various illicit activities [1, 6]. In many cases, OSN users share news and information without any authentication of their veracity, making the OSNs ideal platform to diffuse polluted contents and misinformation[1]. OSN platforms are facing a threat in the form of rumors, where a user either knowingly or unknowingly diffuses false information about an individual, historical facts, etc. The political parties, antisocial elements, and adversaries are exploiting the huge user-base, real-time information diffusion, and lack of effective control mechanism to diffuse rumors against opposition parties and their policies.

Rumor is a piece of information, which is under wider circulation and whose veracity is yet to be confirmed [2]. Nowadays, rumor is one of the major challenges for all the stakeholders of the OSNs. There are several real-life incidents where rumor has resulted in large-scale chaos in the real-world. In some cases, it even caused the loss of human lives. Therefore, identification and removal of such life-threatening contents are vital for the growth and credibility of OSN platforms. Although researchers are devising methods for characterizing and identifying rumors in various social media platforms, most of them are based on hand-crafted features from user-generated contents, user profiles, and diffusion networks to train classification models for labeling the veracity of new instances [3, 9]. In addition, researchers have proposed approaches based on regular expressions for tracking and detecting the rumorous signals [18].

Recently, researchers have started using deep learning techniques towards the debunking of fake news and rumors [10]. Rumors contain more *scepticism*- and *doubts-related* words, such as negation and speculation, in comparison to general discourse on other topics. Rumorous documents generally contain sentimental aspects, such as *anxiety*, *sad*, or *sorrow* representing the uncomfortable and unsatisfactory conditions [4]. In this paper, we target the anxious rumor-spreaders who use anxiety and doubt-related terms in their documents. The frequent presence of such terms in a post or tweet indicates its suspicious nature. We propose a graph-theoretic approach to model user-generated contents as a word co-occurrence graph and use the degree and closeness centrality to identify prominent words of four different lexical categories – *noun*, *verb*, *adverb*, and *adjective*, representing rumor signals. Thereafter, word embedding is applied to represent each category of seed words as numeric vectors and to train three different classification models for rumor detection.

The rest of the paper is organized as follows. Section 2 presents a brief review of the existing literatures on rumor and fake news

[1] https://firstdraftnews.org/coe-report/

WI '19, October 14–17, 2019, Thessaloniki, Greece

ACM ISBN 978-1-4503-6934-3/19/10…$15.00
https://doi.org/10.1145/3350546.3352569

detection. Section 3 presents the functional details of the proposed approach. Section 4 presents the experimental details, performance evaluation, and comparative analysis results. Finally, section 5 concludes the paper and presents the future directions of research.

2 RELATED WORKS

Rumor propagation is not a new phenomenon but exists since the early 20^{th} century. In 1902, German psychologist and philosopher, William Stern performed an experiment to observe the phenomenon of loss of information and found that the information given to an individual is either changed or manipulated when it passes through a chain of people. Robert Knapp studied the rumors propagated during world war II [7]. The scientists and researchers from different discipline have performed the multifaceted study of the problem of rumor and information credibility [5, 12, 14]. With the advent of online social media platforms, the problem of rumors has taken a rise, and computer scientists have started modeling it as a computational problem. Castillo et al. [3] used four categories of features based on *message*, *topic*, *user*, and *propagation tree* to train machine learning classifiers to segregating the credible and non-credible topics. In addition to *content*, *user*, and *propagation* features, Yang et al. [17] used *client* and *location*-based features to train classification models for detecting rumors in Sina Weibo network. In another approach, Qazvinian et al. [13] used the *content*, *network*, and *microblog* features to characterize and differentiate between the rumorous and non-rumorous tweets. Similarly, Sun et al. [15] presented a machine learning approach for characterizing and detecting rumorous events on Sina Weibo. In another approach, Wu et al. [16] modeled the message diffusion process as a propagation tree to extract 23 features and trained a graph kernel-based hybrid SVM classifier to classify the rumor and non-rumor messages in Sina Weibo.

In addition to the feature engineering-based approaches, a number of methods have exploited the scepticism and doubts raised by replying and sharing users to detect the rumor and fake posts in OSNs. In an approach based on inquiry posts, Zhao et al. [18] utilized the questions raised by users about the authenticity of OSN posts to extract *signal tweets*. Similarly, in [19], Zubiaga et al. proposed a context-based rumor detection approach using CRF classifier to learn from the sequential dynamics of social media posts.

With the advancement in deep learning technology, researchers have started exploiting it towards the detection of rumor and fake news. In this line of research, Ma et al. [10] presented a recurrent neural network-based approach to learn the representation of microblogging events for rumor detection. In another approach, Ma et al. [11] presented a deep neural network architecture to jointly model the task of rumor and stance detection. Similarly, a number of approaches have exploited the deep neural network-based techniques to debunk the rumor and fake news in OSNs [8].

3 PROPOSED APPROACH

Figure 1 presents the work-flow of the proposed approach for rumor detection. A detailed description of each functional module is presented in the following sub-sections.

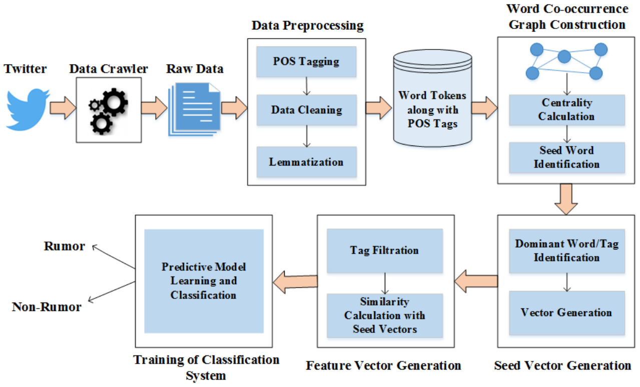

Twitter
Data Crawler
Raw Data
Data Preprocessing
POS Tagging
Data Cleaning
Lemmatization
Word Tokens along with POS Tags
Word Co-occurrence Graph Construction
Centrality Calculation
Seed Word Identification
Dominant Word/Tag Identification
Vector Generation
Seed Vector Generation
Tag Filtration
Similarity Calculation with Seed Vectors
Feature Vector Generation
Predictive Model Learning and Classification
Training of Classification System
Rumor
Non-Rumor

Figure 1: Work-flow of the proposed approach for rumor detection

3.1 Data Preprocessing

In this step, we perform a number of data pre-processing tasks, such as Parts-Of-Speech (POS) tagging, cleaning, tokenization, and lemmatization over the text documents to generate ready-to-analyze data for rumor detection. Initially, we apply POS tagging using spaCy to annotate each words of the documents with their respective POS tags. In data cleaning task, we remove all *URLs*, *mentions*, *hashtags*, and convert uppercase letters into lowercase. We also remove alphanumeric characters, emoticons, numbers, and punctuations. Further, all documents are tokenized and lemmatization is performed on the tokens to find their base forms.

3.2 Rumor Lexicon Generation

In this step, we create a lexicon of seed words that are generally used in rumorous tweets. In a study, Dalziel et al. [4] analyzed the association between anxiety and rumor diffusion by examining tweets related to the Mumbai terrorist attack happened in November 2008 in India. Towards the analysis, they collected 932 tweets related to the attack from November 26 to November 28, 2008, and also on November 30, 2008. They created two lexicons – one for *rumor* and another for *anxiety*, and categorized the words of the both lexicons into four lexical groups based on their POS tags – *noun*, *adjective*, *adverb*, and *verb*. Although lexicons of rumor and anxiety also contain phrases consisting of multiple words, we tokenized them and used each token as a seed word of the lexicon. Rumor and anxiety lexicons have 42 and 30 seed words respectively as given in table 1. Finally, we merge both lexicons into a single one, which is used in the remaining paper as a lexicon of rumor words, also called *seed rumor words*.

3.3 Word Co-occurrence Graph Construction

In this step, we model the user-generated contents as a word co-occurrence graph. The proposed approach is novel in the sense that it utilizes a graph-based algorithm for rumor detection in online social media. Among all the existing graph representations that map texts to graphs, we use the word co-occurrence graph to model the user-generated contents. Formally, the word co-occurrence graph is denoted by a tuple, i.e., $G = (V, E)$ where V is a finite set of vertices representing the words and $E \subseteq V \times V$ is a finite set of edges between the vertices representing the co-occurrence of two words within a fixed window size of n words. In the proposed work, we have taken the value of n as 2. In order to identify the prominent words from the graph, we first calculate two centrality measures – *degree centrality* and *closeness centrality* as prominence indicators for the respective nodes (words).

The *degree centrality* of a node represents its topological importance. It shows connection formation (in our case co-occurrence) probability of the nodes in the graph. Formally, the *degree centrality* of a node $u \in V$ refers to the fraction of all its adjacent neighbors to the number of possible edges with all the nodes of the graph, and it is defined in equation 1, where A represents the adjacency matrix of graph G, and N represents the total number of nodes in G, i.e., $N = |V|$.

$$C_D(u) = \frac{\sum\limits_{v \in adj(u)} A(u,v)}{N-1} \tag{1}$$

On the other hand, *closeness centrality* of a node represents its reachability in the graph. Formally, the closeness centrality of a node $u \in V$ is the average sum of the shortest paths of u with every other vertex of G, and it is defined in equation 2, where $d(u,v)$ represents the shortest distance between the nodes u and v.

$$C_C(u) = \frac{\sum\limits_{v \in V} \frac{1}{d(u,v)}}{N-1} \tag{2}$$

Finally, centrality measure $C(u)$ for each vertex $u \in V$ is calculated using equation 3 which represents the prominence of a node in the graph.

$$C(u) = C_D(u) \times C_C(u) \tag{3}$$

3.4 Seed Vector Generation

This step aims to identify prominent words that are adjacent to the seed rumor words in the graph and generates their vector representations. Initially, we locate all seed rumor words in graph G, and they have usually high centrality values due to the fact that rumor seed words appear more often in rumorous tweets and hence they have high values for the two centrality measures C_D and C_C. Thereafter, based on the lexical category of a seed word s in G, we select the most dominant adjacent words from the remaining three lexical categories. For example, if the seed rumor word *unclear* is found in G which is an *adjective*, then we find the most dominant *noun*, *verb*, and *adverb* words that are adjacent to *unclear* in G. The dominance of a word is calculated based on its centrality value.

Table 1: List of rumor and anxiety words compiled by [4]

Lexical category	Rumor lexicon	Anxiety lexicon
Verb	expect, might, consider, think, assume, deem, sound, appear, claim, hear/heard, seems, believe, doubt, say, guess, thought, confirm, alert, possible	hate, bitch
Adverb	quite, supposedly, perhaps, apparently, maybe, almost, likely, probably	
Adjective	wrong, unsure, unclear, unreal, surreal, possible	wild, crashing, pissed, erie, sad, haunting, heartbreaking, scared, painful, stressed, confused, maddening, heartened, terrible, disturbed, overwhelmed, moronic
Noun	belief, someone, story, gossip, anecdote, buzz, tale, rumor, possibility	fatigue, shock, anger, grief, moron, horror, fear, panic, outrage, hatred, retaliation

Once the dominant adjacent words for each seed rumor word are determined, the next task is to generate their vector representations. The vector representation $\mathcal{S}(s)$ of a seed rumor word s is the average of the word embeddings of s and its dominant adjacent words, one from each remaining three lexical categories, in G. In this work, we have used 200-dimensional `GloVe` word vectors trained over `Twitter` dataset.

3.5 Feature Vector Generation

This step aims to convert each document/tweet into a feature vector, as illustrated in figure 2. To this end, from each tweet, we filter out all the words having POS tags other than the *noun*, *adjective*, *adverb*, or *verb* tags. Thereafter, remaining words are grouped into four lexical categories depending on their respective POS tags.

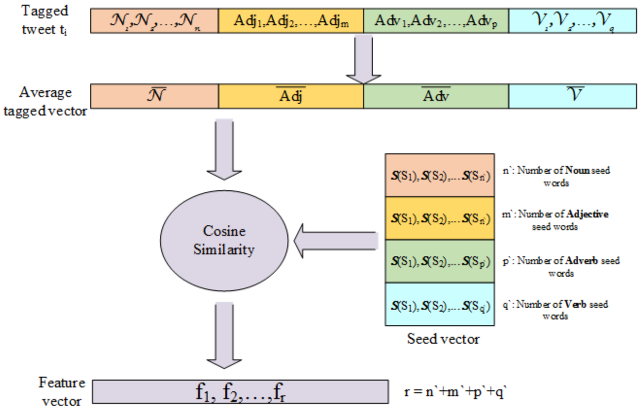

Tagged tweet t_i
$\mathcal{N}_1,\mathcal{N}_2,\ldots,\mathcal{N}_n$
$Adj_1, Adj_2, \ldots, Adj_m$
$Adv_1, Adv_2, \ldots, Adv_p$
$\mathcal{V}_1,\mathcal{V}_2,\ldots,\mathcal{V}_q$
Average tagged vector
$\overline{\mathcal{N}}$
$\overline{Adj}$
$\overline{Adv}$
$\overline{\mathcal{V}}$
Cosine Similarity
$\mathcal{S}(S_1), \mathcal{S}(S_2), \ldots \mathcal{S}(S_{n'})$
$\mathcal{S}(S_1), \mathcal{S}(S_2), \ldots \mathcal{S}(S_{n'})$
$\mathcal{S}(S_1), \mathcal{S}(S_2), \ldots \mathcal{S}(S_{p'})$
$\mathcal{S}(S_1), \mathcal{S}(S_2), \ldots \mathcal{S}(S_{q'})$
n`: Number of Noun seed words
m`: Number of Adjective seed words
p`: Number of Adverb seed words
q`: Number of Verb seed words
Seed vector
Feature vector
$f_1, f_2, \ldots, f_r$
r = n`+m`+p`+q`

Figure 2: Feature vector generation

Within a tweet, the words pertaining to a particular lexical category have a certain degree of similarity, i.e., they are contextually similar up to some extent. For example, all nouns in a tweet will be generally related to each other up to some extent, and hence their word vectors will be closer in the vector space. Therefore, following the categorization of words into four lexical groups, each tweet is converted into a feature vector, based on the word vector-based similarity between the average vector of the words of each lexical group and the word vectors of the seed rumor words of the respective category. For example, if a tweet has four nouns, then first we found the average of the word vectors of all four noun words to generate a single noun vector. Thereafter, we compute the cosine similarity between the noun vector with the word vectors of each seed rumor words that are noun. Similar procedure is repeated for the words of other lexical groups of a tweet. As a result, the length

Table 2: Statistics of the Pheme dataset

Event	Rumors	Non-Rumors	Total
Charlie Hebdo	6887	35104	41991
Ferguson	6195	16837	23032
Germanwings Crash	2256	1764	4020
Ottawa Shooting	5966	5428	11394
Sydney Siege	8154	14621	22775
Total	29458	73754	103212

Table 3: Performance evaluation results over D_b

Classifier	Precision	Recall	F-Score
DT	0.591	0.551	0.578
SVM	0.523	0.507	0.515
CRF	**0.819**	**0.683**	**0.745**

Table 4: Performance evaluation results over D_u

Classifier	Precision	Recall	F-Score
DT	0.438	0.449	0.443
SVM	0.410	0.450	0.429
CRF	**0.646**	**0.599**	**0.622**

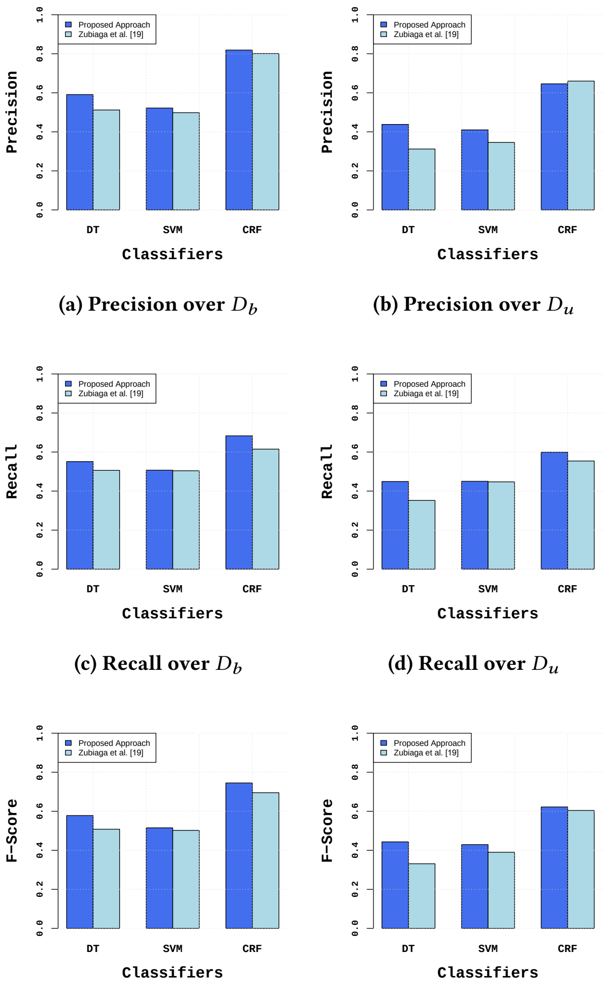
of the feature vector is same as the number of seed rumor words present in G.

4 EXPERIMENTAL SETUP AND RESULTS

The experimental evaluation of the proposed approach is performed using three machine learning algorithms – *decision tree (DT)*, *support vector machine (SVM)*, and *conditional random field (CRF)* over two variations of a benchmark dataset. We have used Pheme dataset provided by [19] which consists of 103212 labeled tweets, out of which 29458 are rumors and 73754 are non-rumors. We call it unbalanced dataset D_u because it contains a large number of non-rumor tweets in comparison to the rumor tweets. The dataset is related to five real-world events, as described in table 2. In order to remove class bias, we created a balanced dataset D_b from D_u consisting of equal number of 29458 rumor and non-rumor tweets.

We trained the classification models using `scikit-learn` python library over both datasets D_b and D_u, separately. We used 70% data for training and remaining 30% data for validation. Tables 3 and 4 present the evaluation results over D_b and D_u, respectively. It can be observed from these tables that CRF performs best in terms of all evaluation metrics. However, performance of all classifiers degrades in case of D_u due to class imbalance problem.

We also compared our approach with one of the state-of-the-art rumor detection methods proposed by Zubiaga et al. in [19]. To this end, we implemented and repeated the same set of experiment for [19] over both datasets D_b and D_u using the same set of classifiers. Figure 3 presents the comparative analysis results in terms of *precision*, *recall*, and *f-score* values. It can be observed from this figure that our approach performs significantly better, except the case of CRF over the unbalanced dataset D_u.



(a) Precision over D_b (b) Precision over D_u

(c) Recall over D_b (d) Recall over D_u

(e) F-Score over D_b (f) F-Score over D_u

Figure 3: Comparative analysis results over D_b and D_u

5 CONCLUSION AND FUTURE WORKS

In this paper, we have proposed a graph-theoretic embedding-based approach for rumor detection in OSN. Starting with the modeling of user-generated contents as a word co-occurrence graph, the proposed approach extracts prominent words from different lexical categories to represent rumorous signals. Word embeddings are used to model tweets as numeric vectors and train classification models. The empirical evaluation of the proposed approach is encouraging and it performs better in comparison to a state-of-the-art rumor detection method. The proposed approach can be extended as a bidirectional learning in which initial set of seed words are used to identify relevant prominent words from the user-generated contents and the process can be repeated until convergence to generate an exhaustive set of rumor characterizing words.

REFERENCES

[1] Faraz Ahmed and Muhammad Abulaish. 2013. A Generic Statistical Approach for Spam Detection in Online Social Networks. *Computer Communications* 36, 10-11 (2013), 1120–1129.

[2] Gordon W Allport and Leo J Postman. 1947. *The psychology of rumor* (first ed.). Henry Holt and Company.

[3] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information Credibility on Twitter. In *Proceedings of Int'l Conf. on WWW*. ACM, Hyderabad, India, 675–684.

[4] Greg Dalziel. 2013. Rumor and Communication in Asia in the Internet Age. , 209 pages.

[5] Nicholas DiFonzo and Prashant Bordia. 2007. Rumor Psychology: Social and Organizational Approaches. https://doi.org/10.1037/11503-000

[6] Mohd Fazil and Muhammad Abulaish. 2017. Why a Socialbot is Effective in Twitter? A Statistical Insight. In *Proceedings of the 9th Int'l Conf. on Communication Systems and Networks (COMSNETS), Social Networking Workshop, Bengaluru, India*. 562–567.

[7] Robert H Knapp. 1944. A Psychology of Rumor. *The Public Opinion Quarterly* 8, 1 (1944), 22–37.

[8] Lizhao Li, Guoyong Cai, and Nannan Chen. 2018. A Rumor Events Detection Method Based on Deep Bidirectional GRU Neural Network. In *Proceedings of the Int'l Conf. on Image, Vision and Computing*. IEEE Computer Society, Chongqing, China, 755–759.

[9] Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. 2015. Real-time Rumor Debunking on Twitter. In *Proceedings of Int'l Conf. on Information and Knowledge Management*. ACM, Melbourne, Australia, 1867–1870.

[10] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwonand Bernard J Jansen, Kam F Wong, and Meeyoung Cha. 2016. Detecting Rumors from Microblogs with Recurrent Neural Networks. In *Proceedings of Int'l Joint Conf. on AI*. AAAI Press, New York, USA, 3818–3824.

[11] Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Detect Rumor and Stance Jointly by Neural Multi-task Learning. In *Proceedings of the Int'l Conf. on WWW*. AAAI Press, Lyon, France, 1–9.

[12] Meredith R Morris, Scott Counts, Asta Roseway, Aaron Hoff, and Julia Schwarz. 2012. Tweeting is Believing? Understanding Microblog Credibility Perceptions. In *Proceedings of the Int'l Conf. on WWW*. ACM, Seattle, USA, 441–450.

[13] Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying Misinformation in Microblogs. In *Proceedings of the Int'l Conf. on Empirical Methods in NLP*. ACL, Edinburgh, Scotland, 1589–1599.

[14] Ralph L Rosnow. 1991. Inside rumor: A personal Journey. *American Psychologist* 46, 5 (1991), 484–496.

[15] Shengyun Sun, Hongyan Liu, Jun He, and Xiaoyong Du. 2013. Detecting Event Rumors on Sina Weibo Automatically. In *Proceedings of the Asia-Pacific Web Conf. on Web Technologies and Applications*. Springer-Verlag Berlin Heidelberg, Sydney, Australia, 120–131.

[16] Ke Wu, Song Yang, and Kenny Q Zhu. 2015. False Rumors Detection on Sina Weibo by Propagation Structures. In *Proceedings of Int'l Conf. on Data Engineering*. ACM, Seoul, South Korea, 651–662.

[17] Fan Yang, Xiaohui Yu, Yang Liu, and Min Yang. 2012. Automatic Detection of Rumor on Sina Weibo. In *Proceedings of the ACM-SIGKDD Workshop on Mining Data Semantics*. ACM, Beijing, China, 1–7.

[18] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Enquiring Minds: Early Detection of Rumors in Social Media from Enquiry Posts. In *Proceedings of the Int'l Conf. on WWW*. ACM, Florence, Italy, 1395–1405.

[19] Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2017. Exploiting Context for Rumor Detection in Social Media. In *Proceedings of the Int'l Conf. on Social Informatics*. Springer Cham, Oxford, UK, 109–123.