Knowledge-Based Systems 207 (2020) 106391

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Near real-time topic-driven rumor detection in source microblogs

Fan Xu [a], Victor S. Sheng [b,*], Mingwen Wang [a]

[a] School of Computer Information Engineering, Jiangxi Normal University, Nanchang, 330022, China
[b] Department of Computer Science, Texas Tech University, Lubbock, TX 79409, USA

ARTICLE INFO

Article history:
Received 13 December 2019
Received in revised form 29 July 2020
Accepted 8 August 2020
Available online 21 August 2020

Keywords:
Rumor detection
Topic vector
Latent dirichlet allocation (LDA)
Source microblogs

ABSTRACT

Rumors can be propagated across online microblogs at a relatively low cost, but result in a series of major problems in our society. Traditional rumor detection approaches focus on exploring various propagation patterns or data interactions between a source microblog and its subsequent reactions. It is obvious that this causes missing interaction on rumor detection, especially in the absence of retweets or reactions. According to the communication theory of Allport and Postman (1947), Chorus (1953) and Rosnow (1988), the topic of a post can help determine its potential of being a rumor or not. Therefore, we develop a novel topic-driven rumor detection (TDRD) framework to determine whether a post is a rumor only according to its source microblog. Specifically, we first automatically perform topic classification on source microblogs, and then we successfully incorporate the predicted topic vector of the source microblogs into rumor detection. Our extensive experimental results demonstrate that our TDRD significantly outperforms state-of-the-art methods on both two English and two Chinese benchmark datasets.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

According to statistics, two-thirds of Americans reported that they got at least some of their news on social media.[1] However, microblogging platforms such as twitter or weibo are an ideal hotbed for spreading rumors, which are harmful to people's lives. For instance, the rumor in 2013 about the White House having been bombed resulted in stock markets spooked.[2] Similarly, the Hurricane Sandy rumor[3] has caused the US Federal Emergency Management Agency finally has to control this rumor.

Humans, however, are susceptible to false information and spread it [1]. Furthermore, studies in social psychology and communications have demonstrated that the human ability of detecting deception is only slightly better than chance. Typical accuracy rates are in the 55%–58% range [2]. Although some websites (snopes,[4] politifact,[5] factcheck,[6] etc.) can debunk some specific types of rumors, they heavily depend on domain experts, who conduct manual fact-checking. Manual fact-checking has serious issues, such as low coverage and long delay. Therefore, automatically debunking rumors is a crucial problem.

Existing research typically resorts to supervised classifiers trained on the well-designed hand-crafted features including user profiles [3–5], diffusion patterns [6–9], message content or its sentiment score [6,10,11]. In contrast, data-driven deep learning models were exploited to avoid manually designed feature engineering, such as RNN-based models [12–15], a multi-task model [16] and a GAN-based model [17].

Nevertheless, existing manually designed feature engineering and data-driven deep learning approaches typically rely on exploring various propagation patterns or data interactions between source microblogs and their subsequent reactions. These may result in missing interaction or a performance descent of rumor detection, especially in the absence of retweets or reactions. For example, the performance of the state-of-the-art GAN-based model [17] drops from accuracy 78.10% to 73.63% on the Pheme dataset without retweets or reactions. In addition, previous approaches generally resort to user profiles, which sometimes cannot be obtained. Therefore, we are motivated to design an innovative solution that can debunk rumors only based on source microblogs, so that we can catch rumors at the early beginning. Furthermore, according to the communication theory of [18,19] and [20], a rumor can be generally determined by its topics or importance. They summarized rumor was technically defined as a proposition of the belief of a topic reference disseminated without official verification. Besides, the amount of

* Corresponding author.
E-mail addresses: xufan@jxnu.edu.cn (F. Xu), Victor.Sheng@ttu.edu (V.S. Sheng), mwwang@jxnu.edu.cn (M. Wang).

1 https://www.journalism.org/2017/09/07/news-use-across-social-media-platforms-2017.
2 https://www.bbc.com/news/world-us-canada-21508660.
3 https://twitter.com/fema/status/264800761119113216.
4 https://www.snopes.com.
5 http://www.politifact.com.
6 https://www.factcheck.org.

https://doi.org/10.1016/j.knosys.2020.106391
0950-7051/© 2020 Elsevier B.V. All rights reserved.

Table 1
Samples from the Weibo benchmark dataset; R indicates rumor; N donates Non-rumor.

Source text	Topic	Type
To make dried fried meat using disposable chopsticks	Food health	R
Xinyu Mao sings his new song "Songs for grandpa and grandma"	Recreational sports	N

rumor in circulation will vary with the importance of the subject to the individuals involved times the ambiguity of the evidence pertaining to the topic at issue. The topic itself was inexplicably rated higher in terms of the importance by the nontransmitters than by the transmitters. Generally speaking, the more important the topic of a post, the higher potential it becomes a rumor. In fact, according to samples from a current widely used rumor dataset Weibo (as shown in Table 1), the post with the topic of food health has a high probability to be a rumor, while the post with the topic of recreational sports generally has a high probability to be a non-rumor.

Therefore, we will develop a novel topic-driven rumor detection (TDRD) framework to determine whether a post is a rumor or not only according to its source microblog. Briefly, we first perform topic classification for source microblogs, and then incorporate the predicted topic vector of the source microblogs into rumor detection. In fact, our novel real number representation of each topic is totally different with the traditional latent dirichlet allocation (LDA) topic model. The traditional LDA model adopts the word embeddings representation of the extracted topic words. In addition, we will investigate the performance of TDRD on four publicly available benchmark datasets (i.e., Pheme, Liar, Weibo and DataFountain), comparing with ten state-of-the-art approaches. The major contributions of this paper are three-folds, as follows:

(1) To the best of our knowledge, this is the systematic study using a distribution representation of the topic-driven model for rumor detection on only source microblogs. Different from the current one-hot LDA-driven topic model, our topic distribution has rich information. Furthermore, the predicted topic distribution can be successfully integrated into current deep learning frameworks according to the characteristic of different languages (i.e., CNN in English and FastText in Chinese) for rumor detection.

(2) We successfully annotate two publicly available benchmark microblog datasets with high quality topic labels, and demonstrate that our topic-driven rumor detection model TDRD significantly outperforms ten state-of-the-art approaches on the two datasets. Similar results can be obtained on other two benchmark fake news datasets (i.e., Liar and DataFountain) with golden topic labels.

The rest of this paper is organized as follows. We review related work in Section 2. In Section 3, we present our topic-driven rumor detection model TDRD in details. In Section 4, we conduct experiments to investigate the performance of our proposed TDRD, comparing with ten state-of-the-art approaches, including dataset preparations and experimental result analysis. Finally, we conclude the paper and discuss future research in Section 5.

2. Related works

In this section, we describe representative computational models for rumor detection and its related research areas, such as fake news detection and truth discovery. Different from rumors, fake news is intentionally and verifiably false but published by a news outlet. Generally, the authenticity of fake news is false, while the authenticity of rumors is unknown. Furthermore, truth discovery can be used to infer the credibility of both structural and unstructured data like twitter [21,22].

According to our survey and analysis, we can classify existing related representative computational models into three categories, i.e., manual feature engineering-based methods, kernel-based models and deep learning-based approaches.

Early research on rumor detection centers on manually designing feature engineering-based methods. Castillo et al. [3] adopted the statistics of user profiles and aggregated sentiment features to train a decision tree classifier to detect the credibility on twitter. In contrast, Kwon et al. [6] used temporal, structural, and linguistic features to investigate rumor propagation patterns in online social media. Similarly, Liu et al. [4] explored user characteristics, such as source credibility and source identity, to debunk rumors. What is more, Ma et al. [7] adopted Dynamic Series-Time Structure (DSTS) to capture the variation of content-based, user-based and diffusion-based features.

Wu et al. [8] found two interesting patterns. False rumors were first posted by normal users, and then reposted and supported by some opinion leaders, and finally reposted by a large number of normal users. By contrast, normal messages were posted by opinion leaders and reposted directly by many normal users. Based on these observations, they first generated propagation trees for rumors. Then, they integrated a random walk graph kernel and a feature vector kernel as a hybrid kernel to conduct rumor detection. In their work, they presented a topic vector feature. The difference between our topic vectors and theirs are two-fold. The first one is that they only adopt the top-k one-hot topic vectors, while our topic vectors are the topic distributions. In fact, the topic distributions keep enrich topic type information compared with the top-k one-hot vectors instead. The second one is that their topic vectors are generated by using the unsupervised LDA model, while our topic distributions are obtained by using the supervised deep learning-based text classification models. In nature, the topic label is not very precise generated by LDA. The topic in LDA are embodied via word distributions. By contrast, the topic types are annotated manually with high consistence in our extended datasets which can reflect the real topic type within source microblog. Similarly, Ma et al. [23] proposed a propagation structure kernel to debunk rumor, and investigated the clues outside the subtrees to incorporate more context information. These are existing representative kernel-based models.

Because of the success of deep learning, deep learning approaches are adopted in rumor detection. Ma et al. [12], Chen et al. [14] and Torshizi and Ghazikhani [15] adopted RNN to conduct rumor detection. In addition, Ma et al. [16] adopted a multi-task learning framework to conduct rumor detection and stance detection simultaneously. Recently, Ma et al. [17] presented a GAN-style approach to conduct rumor detection, where a generator is designed to produce uncertain or conflicting voices, and a discriminator is used to distinguish whether an instance is from the real world. What is more, the models for fake news detection include RNN-based approach [24–27] and CNN-based method [28–31].

Different from manually designing feature engineering methods, existing kernel-based models and deep learning-based approaches, we will automatically generate the representation of source microblogs (i.e., topic distribution of the source post) and make rumor detection only based on source microblogs.

3. Topic-driven rumor detection

In this section, we first provide the problem statement, and then explain our topic-driven rumor detection framework TDRD in detail.

3.1. Problem statement

In short, rumor detection can be addressed as a binary classification problem, which aims to learn a classifier to predict the label of a claim as rumor (R) or non-rumor (N). Generally, a claim is a factual assertion or statement that something is true. In the microblog rumor detection task, a claim is represented by a set of posts (i.e, tweets or weibo). More specifically, in this paper, we represent a rumor dataset as $\{X\}$, where each $X = (y$, source post), and y is a ground-truth label from $\{R, N\}$.

3.2. The framework of TDRD

Fig. 1 shows the framework of our topic-driven rumor detection model TDRD. As it shows, the whole model consists of two phases. The first phase is the topic vector generation, and the second phase is the rumor detection. Since CNN has obtained the state-of-the-art results on many text classification datasets, we build our topic classification based on a recently proposed CNN framework [32]. If source posts are in English, a CNN [32] framework is employed to conduct rumor detection. In contrast, if source posts are in Chinese, a FastText [33] framework will be adopted to debunk rumors. Compared with CNN, FastText can obtain more local information from Chinese word sequences through uni-grams, bi-grams and tri-grams. Besides, FastText can be taken as a specific type of CNN. Similarly, CNN can also obtain local information from word sequences, but it cannot keep the word order information. For Chinese, the word order information plays a crucial role in a sentence structure. The reason is that a same concept can be expressed with only one or two words in English, but needs more words in Chinese instead. After the first phase, the topic vector will be integrated into the CNN or the FastText model for English and Chinese rumor detection respectively.

In the following paragraphs, we will explain major components inside TDRD, i.e., word embeddings, topic vector generation, vector concatenation, and two deep learning models (i.e., CNN and FastText).

Word embeddings: As an effective type of word representation that allows words with similar meaning to have a similar representation, word embeddings are a distributed representation for a text. Each word is represented by a real-valued vector, which is contrasted to the sparse word representations, such as a one-hot encoding. In this work, we adopt the Glove [34] and word2vec [35] embeddings to warm-start the text embeddings for the English and the Chinese dataset respectively.

Topic vector generation: In TDRD, we simply represent the topic vector of source microblogs as the output score for each topic type (details will be shown in Section 4.1) of a popular deep learning model (i.e., CNN or FastText), and normalize it to [0,1] with softmax. Of course, we can represent the topic as a one-hot vector, but the one-hot representation has two potential issues. One issue is that the one-hot representation definitely loses the semantic information among different topics. The other issue is the predicted one-hot vector will be dominated by the output of the topic classification algorithm, especially in the case that some types of topics cannot be successfully detected. Since the output score for each topic is a real vector, it has more semantic information compared with the one-hot vector representation. Detail explanation will be illustrated through our experiments.

Vector concatenation: Suppose the word embeddings of a sentence is $\mathrm{E} = (\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_n)$, where $\mathrm{e}_i \in \mathrm{R}^d$, $\mathrm{E} \in \mathrm{R}^{n*d}$, n is the length of the sentence, and d is the dimension of the embeddings. Similarly, we define the topic vector H as $\mathrm{t} \in \mathrm{R}^m$, where m is the number of the topic types. Then, we copy and extend the topic vector to $\mathrm{T} \in \mathrm{R}^{m*d}$ by concatenating the topic vector H with E, and generate $\mathrm{G} = \mathrm{H} \oplus \mathrm{E}$, where $\mathrm{G} \in \mathrm{R}^{(n+m)d}$.

CNN and FastText: As Fig. 1 illustrates, we use existing deep learning models CNN [32] and FastText [33] to make rumor detection for English and Chinese source microblogs respectively.

Specifically, for the CNN model, we use a convolutional layer to capture the dependency between the word embeddings and topic vectors. Then, a standard max-pooling operation is performed on the latent space, and then feed previous results into a fully connected layer with a softmax activation function to generate the final prediction.

For the FastText, we randomly initialize a matrix of embedding vectors with uni-gram, bi-gram and tri-gram respectively (again, details will be explained in Section 4.2). In addition, we map bi-gram and tri-gram to different tables by using a hash function. Then, we concatenate the uni-gram/bi-gram/tri-gram embeddings with the topic vector, and feed them to a fully connected layer with a softmax activation function to generate the final prediction.

4. Experiments

In this section, we will investigate the performance of our proposed framework TDRD, by comparing with ten state-of-the-art approaches. We will first describe how we extend the two publicly available benchmark datasets (i.e. Pheme and Weibo) in the next subsection. Furthermore, we will introduce other two benchmark datasets (i.e., an English dataset named Liar and a Chinese dataset named DataFountain) with golden topic types. Finally, we will explain our experimental settings, and provide our experimental results in terms of rumor detection and topic detection on the four datasets later.

4.1. Datasets

In this section, we introduce the two extended microblog datasets (i.e., an English microblog dataset Pheme and a Chinese microblog dataset Weibo), including the topic schema design, annotator training description, corpus statistics, and quality assurance. We also introduce other two datasets (i.e., Liar and DataFountain) with golden topic types.

Topic schema: Currently, Pheme [36][7] and Sina Weibo [12][8] are the two publicly available rumor microblog datasets. The former is in English, and the latter is in Chinese. Both of them are binary datasets (i.e., rumor and non-rumor). Further, we introduce other two fake news datasets (i.e., Liar and DataFountain). Wang [30] released an English dataset Liar which has six fine-grained labels for the truthfulness ratings: pants on fire, false, barely true, half true, mostly true, and true. In this work, we conduct a binary classification task on dataset Liar. For the binary runs we grouped pants on fire, false and barely true as FALSE and half true, mostly true and true as TRUE. In addition, we adopt another binary Chinese fake news dataset named DataFountain[9] in this work. The Liar and DataFountain datasets have released the golden topic types. Table 2 illustrates the statistics of four datasets.

As mentioned in the introduction section, the topic is a good indicator for rumor detection. Therefore, we extend Pheme and Weibo datasets with topics on the event level. That is, we annotate corresponding topics for 5,802 and 4,664 events in Pheme and Weibo respectively, totally 10,466 events. Since Liar and DataFountain datasets have released golden topic type, we directly use them in our evaluation. There are 8 topics (i.e., social politics, scientific research, recreational sports, food health,

[7] https://figshare.com/articles/PHEME_dataset_of_rumours_and_non-rumours/4010619.

[8] http://alt.qcri.org/~wgao/data/rumdect.zip.

[9] https://www.datafountain.cn/competitions/422/datasets.

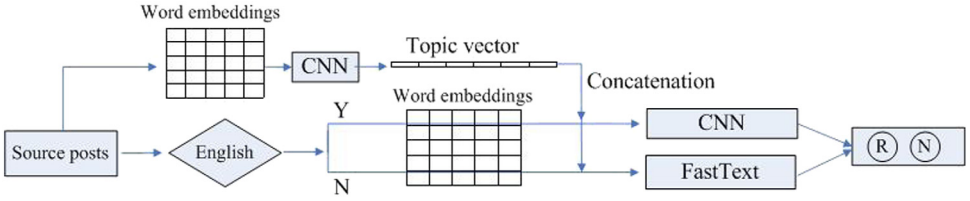

Fig. 1. The framework of our TDRD.

Table 2
Statistics of the four benchmark datasets.

Statistic	Pheme	Liar	Weibo	DataFountain
Source posts #	5,802	7,005	4,664	33,806
Rumor source posts #	1,972	3,060	2,313	16,841
Non-rumor source posts #	3,830	3,945	2,351	16,965

military field, education and employment, economic news, and social life) and 17 topics[10] (i.e., immigration, jobs, campaign-finance, federal-budget, economy, foreign-policy, elections, education, health-care, crime, abortion, children, candidates-biography, energy, job-accomplishments, state-budget, and taxes) in the datasets DataFountain and Liar respectively.

Since some microblogs could be related to multiple topics, it is very difficult to annotate the microblogs in the Pheme and Weibo datasets without an effective topic schema. Therefore, how to design an effective topic schema is critical before annotating. According to the latest survey released by Jingshi Chinese Media Think Tank (JCMTT),[11] they carefully investigated more than 6000 rumors and data from the Tencent mobile products and the platform of School of Journalism & Communication at Beijing normal university and Renmin University of China. They found most rumors are related to scientific knowledge, social politics, celebrity gossip, military, international field, historic culture, and business news. Furthermore, the scientific knowledge can be further divided into food health, ghost superstition, life in space and supernatural phenomenon. Based on this survey and the characteristics of the two datasets (i.e., Pheme and Weibo), we design a two level topic schema as shown in Tables 3 and 4 to extend the Pheme and Weibo dataset respectively. Generally speaking, the fine-grained 16 labels of topics can cover nearly all main popular topics in people's daily life, which guarantees the robustness of the topic labels when facing a new source microblog. Since the number of words in microblogs are limited (i.e., at most 140 characters or words for Weibo and Twitter respectively), people will express the most important topic through the contents in the microblogs. Therefore, we only annotate one main topic to a source post.

More specifically, we define 13 topics for the English Pheme dataset, and 16 topics for the Chinese Weibo dataset. Due to the diversity of Weibo corpus, the number of topics of the Weibo dataset is larger than that of the Pheme dataset, since the Pheme dataset only focuses on breaking news. As shown in Tables 3 and 4, the topic distributions of both Pheme and Weibo are skewed to one class (either rumor or non-rumor) in real situations without human intervention. This indicates that the topics of source microblogs are useful for rumor detection. Therefore, we can infer that our TDRD should perform very well. Since the source posts of the two datasets are not big enough (totally 10,466 posts), we merge some minority topics into a general type "others" in the level-1.

10 We set the minimum number of the topic to 20.

11 https://sjc.bnu.edu.cn/docs/2018-04/20180424112854671948.pdf.

Table 3
Statistics of topic distributions on the Pheme dataset; R indicates Rumor; N donates Non-rumor.

Level-1 topic	Level-2 topic	Events(%)	R(%)	N(%)
Revenge	Revenge	34.63	22.50	77.50
Hostage-taking	Hostage-taking	20.63	43.65	56.35
Police supervision	Police supervision	19.56	24.76	75.24
Police attack	Police attack	15.33	52.28	47.72
Air crash	Air crash	8.01	51.18	48.82
Recreational sports	Recreational sports	0.95	3.64	96.36
Others	Null	0.40	27.27	72.73
	Social politics	0.21	0.00	100.00
	Diffusion forwarding	0.10	0.00	100.00
	Traffic travel	0.09	0.00	100.00
	Education and employment	0.07	0.00	100.00
	Life tips	0.01	100.00	0.00
	Scientific research	0.01	0.00	100.00

Table 4
Statistics of topic distributions on the Liar dataset; R indicates Rumor; N donates Non-rumor.

Topic	#Events (%)	#R (%)	#N (%)
Immigration	5.00	49.43	50.57
Jobs	3.07	38.61	61.39
Campaign-finance	2.84	47.74	52.26
Federal-budget	6.22	40.37	59.63
Economy	14.26	35.44	64.56
Foreign-policy	4.91	50.87	49.13
Elections	7.02	44.92	55.08
Education	8.84	35.54	64.46
Health-care	11.85	52.89	47.11
Crime	5.32	39.14	60.86
Abortion	4.11	46.18	53.82
Children	2.71	39.47	60.53
Candidates-biography	9.61	49.18	50.82
Energy	4.13	48.44	51.56
Job-accomplishments	1.74	46.72	53.28
State-budget	3.20	41.07	58.93
Taxes	5.17	41.44	58.56

Annotator Training: Our annotator team consists of a Ph.D. in Chinese linguistics as the supervisor (a senior annotator) and two graduate students as annotators. The annotation was done in four phases. In the first phase, the annotators spend one week learning the principles of scheme. In the second phase, the annotators spend one week independently annotating the same 500 events, and another one week crosschecking to resolve the difference and to revise the guidelines. In the third phase, the annotators spend

Table 5
Statistics of topic distributions on the Weibo dataset; R indicates Rumor; N donates Non-rumor.

Level-1 topic	Level-2 topic	Events (%)	R (%)	N (%)
Recreational sports	Recreational sports	33.17	27.93	72.07
Social politics	Social politics	16.38	57.46	42.54
International field	International field	14.04	53.89	46.11
Diffusion forwarding	Diffusion forwarding	12.03	68.27	31.73
Anti-corruption	Anti-corruption	9.20	90.21	9.79
Education and employment	Education and employment	4.40	23.90	76.10
Food health	Food health	3.69	70.35	29.65
Others	Life tips	1.52	59.20	40.80
	Emergencies	1.29	33.33	66.67
	Traffic travel	1.16	12.96	87.04
	Natural disaster	1.03	85.42	14.58
	Scientific research	0.49	34.78	65.22
	Health and epidemic prevention	0.49	65.22	34.78
	Economic news	0.43	30.00	70.00
	Null	0.41	31.58	68.42
	Military field	0.27	30.77	69.23

Table 6
Statistics of topic distributions on the DataFountain dataset; R indicates Rumor; N donates Non-rumor.

Topic	#Events (%)	#R (%)	#N (%)
Social politics	3.67	46.01	53.99
Scientific research	0.83	44.84	55.16
Recreational sports	7.75	48.36	51.64
Food health	18.70	52.55	47.45
Military field	1.10	40.59	59.41
Education and employment	2.67	56.49	43.51
Economic news	4.30	47.63	52.37
Social life	60.98	49.49	50.51

three weeks annotating the remaining 9,966 events. In the final phase, the supervisor spends one week carefully proofreading all 10,466 events. (See Tables 5 and 6).

Corpus Statistics: As shown in Table 5, the topics are diverse in the Weibo dataset. By contrast, the Pheme dataset has fewer topics as shown in Table 3, because the Pheme corpus contains a collection of twitter rumors and non-rumors posted during breaking news. The topic distributions of both Pheme and Weibo are skewed to one class (either rumor or non-rumor). By contrast, the topic distributions on both Liar and DataFountain datasets are slight imbalanced as shown in Table 4 and Table 6 respectively.

Quality Assurance: In order to ensure the quality of the extended Pheme and Weibo corpus, we calculate the annotation consistency value. The inter-annotator consistency of the two annotators are 96.63% and 98.17% for level-2 topics on the Weibo and Pheme dataset respectively. This high inter-annotator consistence of both datasets guarantees the corpus's quality and the design of the topic schema.

4.2. Experimental settings

In this section, we first briefly introduce the ten state-of-the-art methods, and then briefly describe evaluation metrics and parameter settings used in our experiments.

Existing systems for rumor detection. To investigate the rumor detection performance of our proposed framework, we will perform comparison studies on following approaches.

DT-Rank: Zhao et al. [9] proposed a decision-tree-based ranking model to detect trending rumors. In their method, they adopted regular expression to search for inquiry phrases and cluster claims, and ranked the clustered results based on well-designed features. We implement their inquiry phrases and features on both datasets.

BOW: We implemented a naive baseline representing the source post using bag-of-words, and trained an SVM classifier for rumor detection. Specifically, for the Chinese Weibo dataset, we use uni-gram, bi-gram, tri-gram, and word segmentation of the source posts respectively. We reported the best results of BOW in Tables 7–10 later, because of the space limitation. For the English Pheme dataset, we only use uni-gram representation.

RNN_Attention: Chen et al. [14] proposed a deep attention based RNN model to rebunk rumor. We implement their algorithm in this work.

LSTM: Torshizi and Ghazikhani [15] proposed a LSTM based model to detect rumor. We also implement their algorithm in this work.

CSI: Ruchansky et al. [25] proposed a CSI (Capture, Score, and Integrate) model to detect fake news. We directly adopt their publicly available source codes[12] on the four datasets.

GAN_RNN: Ma et al. [17] proposed a Generative Adversarial Networks (GAN) style approach to detect rumors. We directly adopt their publicly available source codes[13] on the four datasets.

LDA-Topic: Wu et al. [8] proposed a LDA-based one-hot 18-topic vectors to detect rumor. We implement their topic features, and integrated these topic type features into our CNN and FastText framework for two English and two Chinese datasets respectively.

Other text-classification based models (i.e., CNN, FastText and Transformer): Since we detect rumor from the content of the source microblogs and take the rumor detection as a classification problem, the traditional text-classification or relation-classification based models can be taken as baseline models. Kim [32] proposed a CNN-based model to conduct sentence classification. Joulin et al. [33] proposed a simple and efficient method, FastText, for text classification. In contrast, Vaswani et al. [37] proposed a new simple network architecture (i.e., the transformer) based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. We also reimplement their algorithms in our experiment.

Existing systems for topic detection. We take the topic detection of the source posts as a text classification problem, and investigate the performance of following text classification approaches.

BOW: We implemented a naive baseline representing the source posts using bag-of-words and trained an SVM classifier for topic detection.

Bi-LSTM: Liu et al. [38] proposed a recurrent neural network to capture the semantic relationship with a long distance for text classification. We reimplement their algorithm in this work.

Bi-LSTM_Attention: Zhou et al. [39] proposed an attention-based RNN framework for relation classification. In fact, the attention is the weighted average for the output of each hidden layer. We implement their algorithm in this work.

12 https://github.com/sungyongs/CSI-Code.
13 https://github.com/majingCUHK/Rumor_GAN.

CNN: Kim [32] proposed a CNN-based framework for sentence classification. We implement their algorithm in this work.

FastText: Joulin et al. [33]'s efficient model for text classification, which is similar to the baseline in rumor detection. We also implement their algorithm in this work.

Evaluation Metrics. We will investigate the performance of our proposed framework and other comparing approaches in terms of four popular evaluation metrics: i.e., accuracy Eq. (1), precision Eq. (2), recall Eq. (3), and F1 Eq. (4).

$$Accuracy = \frac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|} \tag{1}$$

$$Precision = \frac{|TP|}{|TP| + |FP|} \tag{2}$$

$$Recall = \frac{|TP|}{|TP| + |FN|} \tag{3}$$

where TP indicates true positive, TN stands for true negative, FP donates false positive, and FN refers to false negative.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

Parameter settings. For the CNN model, we depict three filter region sizes: 2, 3 and 4, each of which has 2 filters. Every filter performs convolution on the sentence matrix and generates (variable-length) feature maps. Then 1-max pooling is performed over each map. In all cases, each size has 256 filters, the batch size is set to 128, the number of hidden layers is set to 2, the dropout probability is set to 0.5, the learning rate is set to 0.001, the large size of words of a sentence is set to 32, and the number of filters is set to 250. For the FastText model, the size of n-gram is set to 250499, the number of neurons in a hidden layer is set to 256, and the setting of the batch size, the dropout probability, the learning rate, the word length, and the number of hidden layers are the same to that of CNN.

What is more, we used pretrained 300-dimensional Glove embeddings[14] and 300-dimensional word2vec embeddings[15] to warm-start the text embeddings for the English and the Chinese datasets respectively. Furthermore, we adopted the utility jieba[16] to conduct Chinese word segmentation. For the transformer model, the number of heads is set to 5, the number of encoders is set to 2, and the batch size, the dropout probability, the large size of words of a sentence, the number of neurons of a hidden layer are set the same to CNN. Besides, we implement DT-Rank using Sklearn[17] with its default parameter settings, SVM using LibSVM[18] with default parameter settings, and deep learning models with pyTorch.[19] In addition, we adopt cross entropy as our loss function for all deep learning models, and use Adam [40] as the adaptive moment estimation approach to optimize proposed models.

4.3. Experimental results

Results on rumor detection: Table 7, Table 9, and Table 10 show the average performance of 5-fold cross validation of all systems on Pheme, Weibo and DataFountain respectively. Table 8 shows the performance of all systems on Liar under the default training and testing splits. From Tables 7–10, we can see that our model outperforms all the ten state-of-the-art approaches on both two English and two Chinese benchmark datasets in terms of accuracy metric.

Table 7
Rumor detection results on the Pheme dataset (R: rumor; N: Non-rumor); Performances that are significantly superior to baseline systems ($p < 0.01$, using paired t-test for significance) are indicated by * (Note here C. indicates class; Att donates attention).

Method	C.	Accu.	Prec.	Rec.	F1
DT-Rank	R	0.5921	0.3820	0.3194	0.3468
	N		0.6765	0.7329	0.7032
BOW	R	0.6601	0.0000	0.0000	0.0000
	N		0.6601	**1.0000**	0.7951
RNN_Attention	R	0.7741	0.6376	**0.8144**	**0.7152**
	N		**0.8835**	0.7526	0.8129
LSTM	R	0.6871	0.0000	0.0000	0.0000
	N		0.6871	**1.0000**	0.8145
CSI	R	0.3399	0.0000	0.0000	0.0000
	N		0.3399	**1.0000**	0.5070
FastText	R	0.6601	0.0000	0.0000	0.0000
	N		0.6601	**1.0000**	0.7951
Transformer	R	0.6040	0.1676	0.2355	0.1585
	N		0.6369	0.7811	0.6361
GAN_RNN	R	0.7363	0.6626	0.4782	0.5080
	N		0.7788	0.8607	0.8140
CNN_text	R	0.7941	0.7642	0.5644	0.6441
	N		0.8051	0.9108	0.8540
CNN_text_predicted LDA-Topic	R	0.7673	0.6862	0.6637	0.6659
	N		0.8244	0.8223	0.8188
TDRD(CNN_text_ golden topics)	R	0.8135*	0.7777	0.6338	0.6973
	N		0.8275	0.9067	0.8650
TDRD (CNN_text_ predicted topics)	R	**0.8266***	**0.8133**	0.6355	0.7130
	N		0.8314	0.9249	**0.8755**

Table 8
Rumor detection results on the Liar dataset (R: rumor; N: Non-rumor); Performances that are significantly superior to baseline systems ($p < 0.01$, using paired t-test for significance) are indicated by * (Note here C. indicates class; Att donates attention).

Method	C.	Accu.	Prec.	Rec.	F1
DT-Rank	R	0.5143	0.5753	**0.6180**	**0.5959**
	N		0.4138	0.3715	0.3915
BOW	R	0.5794	0.0000	0.0000	0.0000
	N		0.5794	**1.0000**	**0.7337**
RNN_Attention	R	0.5781	0.4986	0.5325	0.5150
	N		0.6430	0.6112	0.6267
LSTM	R	0.5807	0.5333	0.0248	0.0473
	N		0.5817	0.9843	0.7312
CSI	R	0.5794	0.0000	0.0000	0.0000
	N		0.5794	**1.0000**	**0.7337**
FastText	R	0.6068	0.5533	0.3375	0.4192
	N		0.6252	0.8022	0.7028
Transformer	R	0.5664	0.4695	0.2384	0.3162
	N		0.5927	0.8045	0.6826
GAN_RNN	R	0.5919	0.5023	0.4068	0.4495
	N		0.6364	0.7203	0.6758
CNN_text	R	0.6120	0.5899	0.2539	0.3550
	N		0.6169	0.8719	0.7225
CNN_text_ predicted LDA-Topic	R	0.6133	0.5401	0.5418	0.5410
	N		**0.6667**	0.6652	0.6659
TDRD (CNN_text_ golden topics)	R	**0.6224***	0.5563	0.5046	0.5292
	N		0.6632	0.7079	0.6848
TDRD (CNN_text_ predicted topics)	R	0.6146	**0.5957**	0.2601	0.3621
	N		0.6188	0.8719	0.7239

14 https://nlp.stanford.edu/projects/glove/.
15 https://github.com/Embedding/Chinese-Word-Vectors.
16 https://pypi.org/project/jieba/.
17 https://scikit-learn.org/stable/.
18 https://www.csie.ntu.edu.tw/~cjlin/libsvm/.
19 https://pytorch.org/.

Table 9
Rumor detection results on the Weibo dataset (R: rumor; N: Non-rumor); Performances that are significantly superior to baseline systems ($p < 0.01$, using paired t-test for significance) are indicated by * (Note here C. indicates class; Att donates attention).

Method	C.	Accu.	Prec.	Rec.	F1
DT-Rank	R	0.6076	0.6038	0.6088	0.6059
	N		0.6120	0.6065	0.6088
BOW	R	0.7436	0.7153	0.8032	0.7562
	N		0.7808	0.6857	0.7294
RNN_Attention	R	0.5290	0.5219	**0.9497**	0.6736
	N		0.6250	0.0879	0.1541
LSTM	R	0.6013	0.5741	0.7413	0.6471
	N		0.6490	0.4651	0.5419
CSI	R	0.4959	0.0000	0.0000	0.0000
	N		0.4959	**1.0000**	0.6630
Transformer	R	0.5041	0.0000	0.0000	0.0000
	N		0.5041	**1.0000**	0.6372
GAN_RNN	R	0.7649	0.7457	0.7805	0.7622
	N		0.7859	0.7503	0.7671
CNN_text	R	0.7301	0.7313	0.7388	0.7314
	N		0.7389	0.7231	0.7260
FastText	R	0.8501	0.8464	0.8330	0.8396
	N		0.8383	0.8511	0.8446
Fast_text_ predicted LDA-Topic	R	0.8544	**0.8863**	0.8107	0.8467
	N		0.8280	0.8974	0.8612
TDRD (Fast_text_ golden topics)	R	**0.8671***	0.8703	0.8603	**0.8652**
	N		**0.8640**	0.8737	**0.8688**
TDRD (Fast_text_ predicted topics)	R	0.8645*	0.8672	0.8240	0.8450
	N		0.8349	0.8759	0.8549

Table 10
Rumor detection results on the DataFountain dataset (R: rumor; N: Non-rumor); Performances that are significantly superior to baseline systems ($p < 0.01$, using paired t-test for significance) are indicated by * (Note here C. indicates class; Att donates attention).

Method	C.	Accu.	Prec.	Rec.	F1
DT-Rank	R	0.7131	0.7054	0.7283	0.7166
	N		0.7215	0.6981	0.7096
BOW	R	0.8864	**0.9792**	0.7886	0.8732
	N		0.8248	0.9834	0.8970
RNN_Attention	R	0.9152	0.8987	0.9352	0.9165
	N		0.9333	0.8954	0.9138
LSTM	R	0.9079	0.9010	0.9156	0.9081
	N		0.9153	0.9001	0.9075
CSI	R	0.5018	0.0000	0.0000	0.0000
	N		0.5018	**1.0000**	0.6683
Transformer	R	0.5018	0.0000	0.0000	0.0000
	N		0.5018	**1.0000**	0.6683
GAN_RNN	R	0.8190	0.7947	0.7955	0.7948
	N		0.8387	0.8371	0.8377
CNN_text	R	0.5018	0.0000	0.0000	0.0000
	N		0.5018	**1.0000**	0.6683
FastText	R	0.9554	0.9565	0.9538	0.9550
	N		0.9547	0.9570	0.9558
Fast_text_ predicted LDA-Topic	R	0.9567	0.9663	0.9461	0.9559
	N		0.9481	0.9673	0.9575
TDRD (Fast_text_ golden topics)	R	**0.9651***	0.9697	0.9598	**0.9646**
	N		0.9609	0.9703	**0.9655**
TDRD (Fast_text_ predicted topics)	R	0.9622*	0.9561	**0.9685**	0.9622
	N		**0.9686**	0.9559	0.9622

Specifically, on the English Pheme and Liar datasets, we can conclude that our TDRD performs the best in terms of accuracy and precision metrics. Compared with the 10 state-of-the-art methods. The accuracy TDRD achieved is significant higher than all other 10 state-of-the-art methods. Its experimental results in terms of the other three measures are also much higher than other 10 state-of-the-art methods except for RNN_Attention model. However, the performance of RNN_Attention is not stable. The recall and F1 of RNN_Attention slightly outperforms our model for rumor type on Pheme dataset. However, its performance is inferior to our model on the Liar dataset.

Among the ten state-of-the-art methods, the CSI performs the worst in terms of accuracy. Since the CSI captures the temporal pattern of user activity on a given article, it is not effective when determining whether a post is a rumor only according to its source microblog. Although the simplest BOW model performs better than DT-Rank in terms of accuracy, it can only detect Non-rumor with F1 of 0.7951 and 0.7337 on Pheme and Liar datasets. This indicates that the basic BOW can learn discriminative features effectively for non-rumor detection.

Among the six representative deep learning-based models (i.e., CNN, LSTM, RNN_Attention, GAN_RNN, Transformer, and FastText), Kim [32]'s CNN model outperforms all others in terms of all four evaluation metrics. As expected, the performance of the GAN_RNN model drops dramatically (from 0.781 accuracy reported in Ma et al. [17] to 0.736 accuracy on the source posts) on only source posts without using retweets or reactions. However, the CNN model can detect Rumor with 0.7642 precision, and detect Non-rumor with 0.9108 recall on Pheme dataset. This shows the high-level discriminative features can be successfully extracted by CNN. Compared with CNN, LSTM can only detect Non-rumor on Pheme dataset. This is because the sequence information does not help much on the source rumor detection. Since the characteristics of rumor or non-rumor cannot be embodied by some specific words in a post, the Transformer with an attention mechanism does not perform very well on rumor detection. After incorporating topic vectors into the CNN framework, we obtain the best performance in terms of all most measures. This shows the importance of the topic for rumor detection. The reason of the performance of the predicted topics is higher than the golden topics is that the golden topic vector is represented by using one-hot style, while the predicted topic vector is the softmax of the output of the CNN, which is more meaningful and has much more semantic information to represent the difference among different topics. Compared with unstable performance of the top-k one-hot LDA-based topic types, our topic distribution representation obtains stable performance.

On the Chinese Weibo and DataFountain datasets, we can make similar conclusion: our TDRD performs the best in terms of two evaluation metrics (i.e., accuracy and F1). The accuracy TDRD achieved is significant higher than all other ten state-of-the-art methods. Its experimental results in terms of the F1 are also much higher than other 10 state-of-the-art methods. Compared with the top-k one-hot LDA-based topic types, our real number representation of topic labels can be successfully integrated into the FastText framework instead and obtain higher performance.

Among the ten state-of-the-art methods, the simplest BOW model performs better than DT-Rank in terms of all four measures. As we mentioned before, due to space limitation, we only report the performance of the character-level BOW, because it performs much better than the word-level one in terms of all four measures. Again, the CSI performs the worst in terms of accuracy. The reason is that the source microblog does not include much temporal pattern of user activity on a given article.

Among the six representative deep learning-based models (i.e., CNN, LSTM, RNN_Attention, GAN_RNN, Transformer, and FastText), FastText obtains the best performance on the Weibo and DataFountain datasets. This indicates that the basic unigram/bi-gram/tri-gram and word order information can be successfully extracted by FastText, while CNN only extracts the local

features of a source post. Again, since the number of source posts are not big enough, the state-of-the-art GAN–RNN model cannot play a significant role on the source microblogs. Similarly, after incorporating topic types into the FastText-based model, we get the best performance in terms of accuracy and F1 measures.

Compared with our topic schema, the 7 out of 8 topics (87.50%) in the DataFountation dataset also exist in our topic schema on the Weibo dataset. Similarly, the 12 out of 17 topics (70.60%) in the Liar dataset also exist in our topic schema on the Weibo dataset. For example, immigration vs. international field, jobs vs. Education and employment, campaign-finance vs. economy news, economy vs. economy news, foreign-policy vs. international field, elections vs. social politics, education vs. Education and employment, health-care vs. Health and epidemic prevention, crime vs. Emergencies, job-accomplishments vs. Education and employment, state-budget vs. economic news, taxes vs. economic news. The topic coverage shows the effectiveness of our topic schema design.

In addition, the experimental results as shown in Tables 7–10 above also show the effectiveness of our topic schema. The fixed 18-topic distribution of sina weibo (http://huati.weibo.com/) generated by using LDA cannot bring performance improvement on the four benchmark datasets, simultaneously. On the contrary, our topic distribution generated by using the deep learning-based text classification models can help to detect rumor effectively on the four benchmark datasets. Although, the topic distributions of both Pheme and Weibo are skewed to one class (either rumor or non-rumor), the slight imbalanced topic distribution on both Liar and DataFountain datasets can also help to debunk rumor.

Furthermore, on the English Pheme dataset, the four baselines (i.e., BOW, LSTM, CSI, and FastText) obtain the recall for "N" with 1.0 and cannot detect rumor "R". Similar results can be observed for BOW and CSI on the English Liar dataset. On the Chinese Weibo dataset, the two baselines (i.e., CSI and Transformer) reach the recall for "N" with 1.0. Similar results can be achieved for CSI, Transformer and CNN_text on the Chinese DataFountain dataset. Since the CSI only captures the temporal pattern of the user activity on a given article, it has limitation on the lack of interaction only according to its source microblog. Therefore, it cannot perform well on both two English and two Chinese benchmark datasets. Due to the rich semantic representation of Chinese compared with English, the basic BOW cannot perform well on the two English datasets. On the contrary, it can obtain quite promising performance on the two Chinese datasets (i.e., F1 0.8732 and 0.8970 for "R" and "N" respectively on the DataFountain dataset). Since the rumor cannot be embodied by using some representative words in the source post, the attention mechanism in the Transformer does not help rumor detection. The characteristic is different between Chinese and English. Compared with English, the word order of the sequence of word and the fine-grained words semantic representation are much important in Chinese. Since the FastText can obtain more local information (i.e., word order) in Chinese, it performs well on the Chinese dataset. By comparison, the CNN cannot keep the word order information, resulting the low performance in English. Compared with the Liar dataset, the Pheme, Weibo and DataFountation datasets are much imbalanced. Therefore, the LSTM cannot perform well on the Pheme. Considering the above different characteristic between Chinese and English, our models can obtain the best performance when using FastText on Chinese and CNN in English. Performance can be further improved by integrating our topic distribution information of a source post.

Results on topic detection: Tables 11–14 show the performance of all the systems on topic detection in terms of accuracy, precision, recall and F1-measure on the four dataset respectively. Since the topic distributions for the four datasets are imbalanced, we report the weighted average of precision, recall and F1-measure. Similarly, we also conduct 5-fold cross-validation for topic detection on Pheme, Weibo and DataFountain datasets. We adopt the default training and testing splits of Liar dataset to conduct topic detection task. What is more, due to the space limitation and the performance of BOW using the character-level on the Chinese Weibo dataset is much higher than that of BOW using the word-level representation, we only report the performance of character-level BOW on the Weibo and DataFountain corpus, as shown in Tables 13 and 14. From Tables 11–14, we can see that CNN has the best topic detection performance on both the two Chinese and the two English datasets, along with all other evaluation metrics (i.e., precision, recall and F1). Furthermore, we also conduct topic detection using CNN with attention and transformer models. However, their performance is much lower. Since the Liar dataset was extracted by using the API of POLITIFACT, their topics are much closer, resulting in lower topic detection performance on Liar dataset. Overall, the performance of the topic detection is not high. This indicates the challenging only using source microblogs. We will improve the performance of topic detection in the future.

Table 11
Topic detection results on the Pheme dataset.

Method	Accu.	Prec.	Rec.	F1
BOW	0.3462	0.1200	0.3462	0.1782
Bi-LSTM	0.3462	0.1200	0.3462	0.1782
Bi-LSTM_Attention	0.3032	0.1513	0.3032	0.1625
FastText	0.3462	0.1200	0.3462	0.1782
CNN	**0.7365**	**0.7295**	**0.7352**	**0.7309**

Table 12
Topic detection results on the Liar dataset.

Method	Accu.	Prec.	Rec.	F1
BOW	0.1367	0.0187	0.1367	0.0329
Bi-LSTM	0.2904	0.3135	0.2904	0.2913
Bi-LSTM_Attention	0.3073	0.3025	0.3073	0.2887
FastText	0.3346	**0.3546**	0.3346	0.2922
CNN	**0.3424**	0.3423	**0.3424**	**0.3314**

Table 13
Topic detection results on the Weibo dataset.

Method	Accu.	Prec.	Rec.	F1
BOW	0.3559	0.3454	0.3559	0.2111
Bi-LSTM	0.4736	0.4694	0.4736	0.4280
Bi-LSTM_Attention	0.5832	0.5708	0.5832	0.5618
FastText	0.6726	0.6737	0.6726	0.6656
CNN	**0.7091**	**0.7050**	**0.7091**	**0.7010**

Table 14
Topic detection results on the DataFountain dataset.

Method	Accu.	Prec.	Rec.	F1
BOW	0.7376	0.7156	0.7376	0.6784
Bi-LSTM	0.8822	0.8796	0.8822	0.8759
Bi-LSTM_Attention	0.8930	0.8924	0.8930	0.8901
FastText	0.9156	0.9148	0.9156	0.9134
CNN	**0.9176**	**0.9177**	**0.9176**	**0.9158**

5. Conclusions and future work

Most existing works on rumor detection from social media focus on extracting propagation or data interactions among source microblogs and their following retweet or reactions. This results in missing interaction on rumor detection, especially when the number of reactions is rare or absent. In order to capture the importance of a post, according to existing communication-driven theory, we developed a novel topic-driven rumor detection (TDRD) framework to determine whether a post is a rumor

according to its source microblog only. Specifically, we first automatically performed topic classification on source microblogs, and then we incorporated the predicted topic vector of the source microblogs with word embeddings of the source microblogs for rumor detection. Our novel real number representation of each topic label is totally different from the traditional word embeddings representation of the extracted topic words through the traditional LDA model. Our extensive experimental results demonstrate our TDRD significantly outperforms ten state-of-the-art methods on both two English and two Chinese benchmark datasets.

In the future, we would like to expand current corpus and integrate more communication theory driven features into our proposed framework. Besides, topic detection of source microblogs is a challenge task. We will develop novel solutions to improve the performance of topic detection of source microblogs in the future. In addition, we are leaving the time consumption of each model as one of our future works.

CRediT authorship contribution statement

Fan Xu: Conceptualization, Methodology, Software. **Victor S. Sheng:** Manuscript modification and suggestions. **Mingwen Wang:** Result analysis and editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank anonymous reviewers for their insightful comments on this paper. This research was supported by the National Natural Science Foundation of China under Grant 61772246 and 61876074, Joint Funding Project of Jiangxi Science and Technology Plan, China under Grant 20192ACBL21030, the Social Science Planning Project in Jiangxi, China under Grant 17YY05, and the Humanities and Social Sciences projects in Colleges and universities in Jiangxi, China under Grant YY17211.

References

[1] S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, Science 359 (6380) (2018) 1146–1151.
[2] V.L. Rubin, On deception and deception detection: Content analysis of computer-mediated stated beliefs, Proc. AIST (2010) 1–10.
[3] C. Castillo, M. Mendoza, B. Poblete, Information credibility on twitter, Proc. WWW (2011) 675–684.
[4] X.M. Liu, A. Nourbakhsh, L.Q. Z., S. Fang, Real-time rumor debunking on twitter, Proc. CIKM (2015) 1867–1870.
[5] V. Qazvinian, E. Rosengren, D.R. Radev, Q.Z. Mei, Rumor has it: identifying misinformation in microblogs, Proc. EMNLP (2011) 1589–1599.
[6] S. Kwon, M. Cha, K. Jung, W. Chen, Y. Wang, Prominent features of rumor propagation in online social media, Proc. ICDM (2013) 1103–1108.
[7] J. Ma, W. Gao, Z.Y. Wei, L.Y. M., W.K. F., Detect rumors using time series of social context information on microblogging websites, Proc. CIKM (2015) 1751–1754.
[8] K. Wu, S. Yang, K.Q. Zhu, False rumors detection on sina weibo by propagation structures, Proc. ICDE (2015) 651–662.
[9] Z. Zhao, R. P., M.Q. Z., Enquiring minds: early detection of rumors in social media from enquiry posts, Proc. WWW (2015) 1395–1405.
[10] F.T. Li, M. Huang, Y. Yang, X.Y. Zhu, Learning to identify review spam, Proc. IJCAI (2011) 2488–2493.
[11] J.W. Li, M. Ott, C. Cardie, E. Hovy, Towards a general rule for identifying deceptive opinion spam, Proc. ACL (2014) 1566–1576.
[12] J. Ma, W. Gao, P. Mitra, S. Kwon, B.J. Jansen, K.F. Wong, M.Y. Cha, Detecting rumors from microblogs with recurrent neural networks, Proc. IJCAI (2016) 3818–3824.
[13] J. Ma, W. Gao, K.F. Wong, Rumor detection on twitter with tree-structured recursive neural networks, Proc. ACL (2018) 1980–1989..
[14] T. Chen, L. Wu, X. Li, J. Zhang, H.Z. Yin, Y. Wang, Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, Cham, 2018, pp. 40–52.
[15] A.S. Torshizi, A. Ghazikhani, Automatic twitter rumor detection based on LSTM classifier, in: International Congress on High-Performance Computing and Big Data Analysis, Springer, Cham, 2019, pp. 291–300.
[16] J. Ma, W. Gao, K.F. Wong, Detect rumor and stance jointly by neural multi-task learning, Proc. WWW (2018) 585–593.
[17] J. Ma, W. Gao, K.F. Wong, Detect rumors on twitter by promoting information campaigns with generative adversarial learning, Proc. WWW (2019) 3049–3055.
[18] G.W. Allport, L.J. Postman, The psychology of rumor, J. Clin. Psychol. 3 (247) (1947).
[19] A. Chorus, The basic law of rumor, J. Abnorm. Soc. Psychol. 48 (2) (1953) 313–314.
[20] R.L. Rosnow, Rumor as communication: a contextualist approach, J. Commun. 38 (1988) 12–28.
[21] D. Wang, L. Kaplan, H. Le, T. Abdelzaher, On truth discovery in social sensing: a maximum likelihood estimation approach, Proc. IPSN (2012) 233–244.
[22] D. Wang, C. Huang, Confidence-aware truth estimation in social sensing applications, Proc. IEEE ICS (2015) 336–344.
[23] J. Ma, G. W., K.F. Wong, Detect rumors in microblog posts using propagation structure via kernel learning, Proc. ACL (2017) 708–717.
[24] K. Popat, S. Mukherjee, A. Yates, G. Weikum, DeClarE: debunking fake news and false claims using evidence-aware deep learning, Proc. EMNLP (2018) 22–32.
[25] N. Ruchansky, S.Y. Seo, Y. Liu, CSI: A hybrid deep model for fake news detection, Proc. CIKM (2017) 797–806.
[26] H. Rashkin, E. Choi, J.Y. Jang, Truth of varying shades: analyzing language in fake news and political fact-checking, Proc. EMNLP (2017) 2931–2937.
[27] L. Wu, H. Liu, Tracing fake-news footprints: characterizing social media messages by how they propagate, Proc. WSDM (2018) 1–9.
[28] F. Qian, C.Y. Gong, K. Sharma, Y. Liu, Neural user response generator: fake news detection with collective user intelligence, Proc. IJCAI (2018) 3834–3840.
[29] H. Karimi, P.C. Roy, S.S. Sadiya, J.L. Tang, Multi-source multi-class fake news detection, Proc. COLING (2018) 1546–1557.
[30] W.Y. Wang, "Liar, liar pants on fire":a new benchmark dataset for fake news detection, Proc. ACL (2017) 422–426.
[31] Y. Liu, Y.F.B. Wu, Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks, Proc. AAAI (2018) 354–361.
[32] Y. Kim, Convolutional neural networks for sentence classification, Proc. EMNLP (2014) 1746–1751.
[33] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, Proc. EACL (2017) 427–431.
[34] J. Pennington, R. Socher, C.D. Manning, GloVe: global vectors for word representation, Proc. EMNLP (2014) 1532–1543.
[35] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, Proc. ICLR (2013) 1–12.
[36] A. Zubiaga, M. Liakata, R. Procter, Learning reporting dynamics during breaking news for rumour detection in social media, 2016, arXiv preprint arXiv:1610.07363v1.
[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, Attention is all you need, Proc. NIPS (2017) 6000–6010.
[38] P.F. Liu, X.P. Qiu, X.J. Huang, Recurrent neural network for text classification with multi-task learning, Proc. IJCAI (2016) 2873–2879.
[39] P. Zhou, W. Shi, J. Tian, Z.Y. Qi, B.C. Li, H.W. Hao, B. Xu, Attention-based bidirectional long short-term memory networks for relation classification, Proc. ACL (2016) 207–212.
[40] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.