

Exploiting Microblog Conversation Structures to Detect Rumors

Jiawen Li*, Yudianto Sujana*, Hung-Yu Kao

Intelligent Knowledge Management Lab

Department of Computer Science and Information Engineering

National Cheng Kung University

Tainan, Taiwan

p78073012@gs.ncku.edu.tw, yudianto.sujana@staff.uns.ac.id,

hykao@mail.ncku.edu.tw

Abstract

As one of the most popular social media platforms, Twitter has become a primary source of information for many people. Unfortunately, both valid information and rumors are propagated on Twitter due to the lack of an automatic information verification system. Twitter users communicate by replying to other users' messages, forming a conversation structure. Using this structure, users can decide whether the information in the source tweet is a rumor by reading the tweet's replies, which voice other users' stances on the tweet. The majority of rumor detection researchers process such tweets based on time, ignoring the conversation structure. To reap the benefits of the Twitter conversation structure, we developed a model to detect rumors by modeling conversation structure as a graph. Thus, our model's improved representation of the conversation structure enhances its rumor detection accuracy. The experimental results on two rumor datasets show that our model outperforms several baseline models, including a state-of-the-art model.

1 Introduction

Social media platforms have become a primary source of information due to the ease of sharing information they provide. The latest survey from the Pew Research Center states that 68% of American adults occasionally read news on social media platforms (Holcomb et al., 2013). However, the credibility of the massive amount of news propagated through social media is questionable due to the absence of editors who can validate it (Zubiaga et al., 2018). As a result, social media platforms have become perfect avenues for spreading unverified information and rumors.

Users on social media platforms communicate by replying to other users' posts, repeatedly responding to one another, forming a conversation structure. The conversation structure established by social media consists of a tree representation of the information distributed by users posting at a specific time in response to a source post. (Belkaroui et al., 2014; Cogan et al., 2012; Magnani et al., 2011). Furthermore, Pace et al. (2016) distinguish two types of conversation: dialogic (horizontal conversation among users) and dialectic (vertical conversation with a source post) conversations.

Conversations on social media can influence users' perception of the information within them (Pace et al., 2016). For example, Figure 1 shows a rumor conversation on Twitter in which users give opinions, make conjectures, or supply evidence in reply to the source post or other users' replies. Users can obtain clues about the truth of the source post by reading the replies in the conversation structure. Ma et al. (2018) refer to this self-correcting mechanism in their research. However, the majority of users disregard such comments, preferring to immediately share tweets; thus, this self-correcting mechanism cannot

* Equal contributions.

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: <http://creativecommons.org/licenses/by/4.0/>.

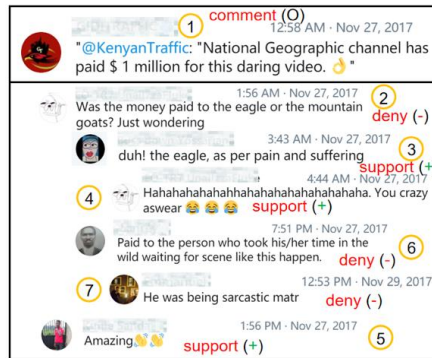


Figure 1. Conversation structure

prevent the spread of rumors in social media. Therefore, an automatic rumor detection mechanism is necessary.

Several researchers have attempted to automatically detect rumors by analyzing the text features (Ajao et al., 2018; T. Chen et al., 2018; Guo et al., 2018; Ma et al., 2016; Yu et al., 2019; Zubiaga et al., 2017), user and message features (Liu et al., 2015; Yang et al., 2012; Zhao et al., 2015), or both (Lukasik et al., 2016; Nguyen et al., 2017). However, most of the existing approaches rely on analyzing a single message and ignore the topological information of the social media conversation structure.

Ignoring the conversation structure can lead to information misinterpretation and affect rumor detection accuracy. For example, in Figure 1, the numbers indicate the sequence of tweet publishing times, and the words depicted in red font represent the stance of the posts, namely, support, deny, or comment, symbolized by (+), (-), and (O), respectively. Because each specific reply is directly related to the source tweet or other responses, a conversation structure is formed. Deep learning methods such as RNNs (including LSTM and GRU) and CNNs ignore this relationship and oversimplify it into a time-based chain structure for encoding tweets (see Figure 2). For instance, based on the tweet publication time, the 5th tweet will be fed into the model after the 4th tweet and before the 6th tweet. According to this time-based chain structure and the tweets' stances, the 5th tweet supports the 4th tweet, and the 6th tweet denies the 5th tweet. However, the 5th tweet is actually a reply to the source tweet, not the 4th tweet. In other words, there is no actual relationship between the 4th, 5th, and 6th tweets, but the time-based chain structure assigns them a false relationship.

A social media conversation can be illustrated by a graph where each message is a node, and the relationships between posts are edges. Figure 3 shows a graph representation of the conversation structure described in Figure 1 that maintains the original conversation structure.

To utilize the conversation structure between a source tweet and its replies, we propose a graph-based deep learning model for rumor detection. The graph structure overcomes the shortcomings of RNNs and CNNs by processing tweets based on conversation structure rather than based on time. Our model consists of the following three hierarchically structured modules: the Tweet Representation Module (TRM), the Conversation Propagation Module (CPM) and the Classifier Module (CM). The TRM captures the high-level information of a tweet and creates its representation, the CPM propagates the tweet's representation through a graph structure, and the CM is a deep neural network for rumor classification.

The main contributions of our study are as follows:

- To the best of our knowledge, our study is the first to empirically integrate both the conversation structure and graph neural networks to detect rumors.
- Experiments based on two Twitter datasets show that our model achieves the highest accuracy and outperforms state-of-the-art baselines.
- Our model can successfully detect rumor in early stages.

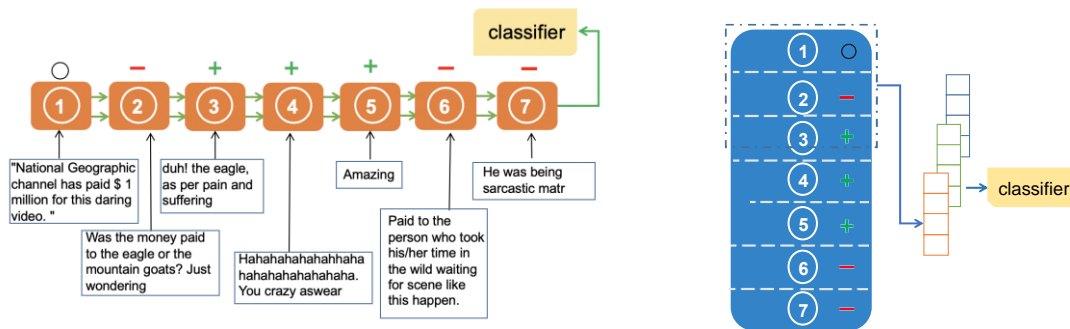


Figure 2. RNN and CNN models

2 Related Work

Rumor detection is generally approached as a classification task employing two major methodologies: traditional machine learning and deep learning. Traditional machine learning relies on structured data and hand-crafted features to identify rumors. Castillo et al. (2011) proposed a large number of features for rumor detection by analyzing users' behavior, messaging, and linking to external sources. Yang et al. (2012) introduced two new features to enrich the feature set of previous researchers: client-based features and location-based features. Wu et al. (2015) employed 23 features divided into three categories (message-based, user-based, and repost-based features) in a hybrid SVM.

On the other hand, deep learning does not depend on hand-crafted features and automatically extracts features from data. RNNs are the most widely used deep learning models for text classification tasks, including rumor detection. RNNs can extract features and learn contextual information from text over time (Ma et al., 2016; Rath et al., 2017). To shorten the training time and enhance the prediction accuracy, Ajao et al. (2018) proposed a hybrid approach that integrates CNNs and RNNs.

Regardless of their traditional machine learning or deep learning approaches, all of the aforementioned researchers process data based on time, ignoring structure information. To take advantage of the event structure in social media, Guo et al. (2018) proposed a hierarchical structure combined with social attention to process the data based on an event. These authors divided the process into three levels: word, post, and subevent levels. However, this event structure does not reflect the actual conversation structure. Using another approach, Ma et al. (2018) constructed a recursive neural network to handle conversational structure. This model generates a tree structure by bottom-up or top-down propagation. However, the nodes' influences are unbalanced, as the last nodes have a greater impact on the representation results. Moreover, because this model uses an RNN as a processing unit, it also encounters the long-term dependencies problem. In addition, an acyclic graph is required as input; thus, graph generalization is unreliable.

Graph neural networks (GNNs) have recently become a popular model in deep learning research. GNNs process information by modeling the dependencies between nodes through message passing. Moreover, GNNs achieve a state that contains information from their neighborhood with varied depth (Xu et al., 2019; J. Zhou et al., 2018). GNN variants have demonstrated solid performance on a variety of NLP tasks, e.g., text classification, sentiment analysis, neural machine translation, and multihop reading comprehension. The GNN variant GraphSAGE uses a general inductive model to learn embedded

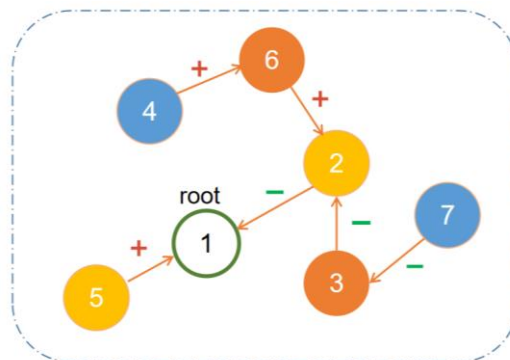


Figure 3. Graph representation

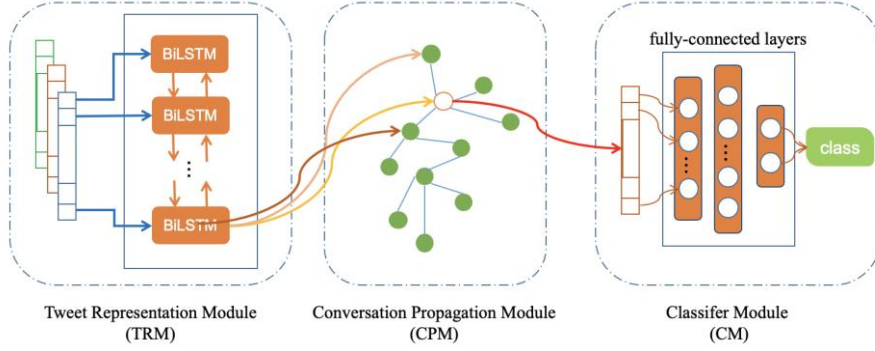


Figure 4. Rumor detection model

nodes where each node is represented by the aggregation of its neighbors (Hamilton et al., 2017). GraphSAGE can learn from dynamic graphs such as those found in social media conversations with variable numbers of graph nodes.

3 Model Architecture

We formulate our task as a supervised classification problem, designating the detection unit as a conversation involving a single source post and its replies. Let C denote a conversation $C_e = \{P_s, P_1, P_2, \dots, P_{|c|}\}$ where P_s is the source post, and $P_{|c|}$ is the last relevant post. Note that the numbers assigned to each post do not indicate that the conversation has a sequential structure; rather, the links between each post are based on reply or repost relationships. The objective of our model is to classify C_e as ‘rumor’ or ‘nonrumor’. The classifier performs learning through labeled information, i.e., $f_e: C_e \rightarrow y_e$.

The core concept of our approach is to strengthen the representation of the information by propagating it through conversation structure. To achieve this goal, we designed three modules: the Tweet Representation Module (TRM), Conversation Propagation Module (CPM), and Classifier Module (CM) (Figure 4).

3.1 TRM

The TRM module contains two components: a word embedding component that maps input words into fixed-sized vectors and a deep BiLSTM that processes sequential word vectors and extracts high-level information from each tweet.

In the word embedding layer, we map the words in post x_t into vectors, yielding fixed-length vectors for each word:

$$x_t = E\theta x_t \quad (1)$$

where x_t is the t_{th} word in a post, and E is a special word embedding matrix.

A deep BiLSTM is used to capture the relationships between words and generate the tweet representation:

$$h_{LSTM_i} = BiLSTM(x_t, h_{LSTM_{i-1}}) \quad (2)$$

3.2 CPM

To propagate through the conversation structure and generate improved representations of the conversation, we use GraphSAGE to create low-dimensional vector representations from both training and unseen nodes (Hamilton et al., 2017). The output of the TRM module is a set of tweet-embedding vectors lacking conversation structure information. Thus, before the vectors are fed into the CPM module, a mapping process needs to be performed. This mapping process aims to map the conversation structure to a graph object where one post becomes one node, and the edges reflect the reply relationship it has with each of the other posts:

$$\begin{aligned} G &= (V, E, U) \\ V &= [h_{LSTM_{p_s}}, h_{LSTM_{p_1}}, h_{LSTM_{p_2}}, \dots, h_{LSTM_{p_{|c|}}}] \\ E &= [v_{p_s} \rightarrow v_{p_1}, \dots, v_{p_n} \rightarrow v_{p_m}] \\ U &= y_{p_s} \end{aligned} \quad (3)$$

Statistic	PHEME 2017	PHEME 2018
Users	49,345	50,593
Posts	103,212	105,354
Events	5,802	6,425
Avg words/post	13.6	13.6
Avg posts/event	17.8	16.3
Max posts/event	346	246
Rumor	1972	2402
Non-rumor	3830	4023
Balance degree	34.00%	37.40%

Table 1. Datasets statistic

G is a graph object that consists of vertices V , edges E , and properties U . V contains the output of the TRM module represented as $h_{LSTM_{pi}}$. E is the relationship of the vertices, and U is the global property of the graph. Since the model is trained in a supervised manner, the label of the conversation is saved as the global property of the graph object.

At the beginning of the forward propagation step, the feature of each node is assigned to the nodes in the hidden state as follows:

$$h_{graph_v}^0 \leftarrow h_{LSTM_v}, \forall v \in V \quad (4)$$

where $h_{graph_v}^0$ is the initial hidden state of the nodes in GraphSAGE

GraphSAGE works by aggregating information from local neighbor nodes at each iteration until all the nodes are accessed. This process makes the nodes gain incrementally richer information (Hamilton et al., 2017):

$$\begin{aligned} h_{graph_{N(v)}}^k &\leftarrow AGGREGATE(\{h_{graph_u}^{k-1}, \forall u \in N(v)\}) \\ h_{graph_v}^k &\leftarrow \sigma(W^k \cdot CONCAT(h_{graph_v}^{k-1}, h_{graph_{N(v)}}^k)) \end{aligned} \quad (5)$$

where $h_{graph_{N(v)}}^k$ is the aggregated neighborhood vector, k is the depth of the information transmission updates (the number of times the graph information is updated), N is the neighborhood function, $N(v)$ is the set of the node's immediate neighborhood, and AGGREGATE is the aggregation function.

In this paper, we use the max pooling aggregator to aggregate a node with neighborhood information, where max is the element-wise max operator, and σ is a nonlinear activation function:

$$AGGREGATE_k^{pool} = \max(\{\sigma(W_{pool} \cdot h_{graph_{N(v)}}^k + b_{pool}), \forall u_i \in N(v)\}) \quad (6)$$

After k iterations, we obtain the output representation C_v , the conversation embedding results:

$$C_v \leftarrow h_{graph_v}^k, \forall v \in V \quad (7)$$

3.3 CM

The CM module is an MLP module. Based on the output of the CPM, we use a softmax function in the output layer to predict the label of the conversation:

$$\hat{y} = Softmax(V \cdot C_v + b_y) \quad (8)$$

where V and b_y are parameters in the output layer.

For each training process, the goal is to minimize the standard deviation between the predicted and output values using the following loss function:

$$L = \sum(y_e - \hat{y})^2 + \sum\|\theta_i\|^2 \quad (9)$$

where y_e is the target value, and θ_i is the model parameters to be estimated. The L2-regularization penalty is used for trading off the error and scale of the problem.

Dataset	Method	Acc	Pre	Rec	F1
PHEME 2017	SVM-BOW	0.669	0.535	0.524	0.529
	CNN	0.787	0.737	0.702	0.719
	BiLSTM	0.795	0.763	0.691	0.725
	BERT	0.865	0.859	0.851	0.855
	RDM*	0.873	0.817	0.823	0.820
	Our Model	0.900	0.893	0.869	0.881
PHEME 2018	SVM-BOW	0.688	0.518	0.512	0.514
	CNN	0.795	0.731	0.673	0.701
	BiLSTM	0.794	0.727	0.677	0.701
	BERT	0.844	0.834	0.835	0.835
	RDM*	0.858	0.847	0.859	0.853
	Our Model	0.919	0.892	0.923	0.907

* state-of-the-art

Table 2. Experiment result

4 Experiments and Results

4.1 Dataset

Two Twitter datasets are used to extensively evaluate our proposed model, PHEME 2017 and PHEME 2018, which were created by (Zubiaga, Liakata, and Procter, 2016) and (Zubiaga, Liakata, Procter, et al., 2016), respectively. Table 1 describes the statistics of these dataset.

4.2 Experimental Setup

In our preprocessing phase, we empirically clean the text by deleting hyperlinks, emojis, and stop-words. We use Twitter 27B pretrained GloVe data with 200 dimensions for word embedding and set the maximum vocabulary to 80,000. We use Adam with a 0.001 learning rate to optimize the model during training.

The hyperparameters on TRM are: batch size = 64; dropout rate = 0.5; hidden size = 70; and number of layers = 2. For CPM, they are: batch size = 128; aggregation function is the maximum; and activation function is ReLU. For CM, they are: number of layers = 2; dropout rate = 0.5; activation function between layers is ReLU and Sigmoid in the last layer to predict rumors.

4.3 Experiments

We compared our model with the following baseline models, among which RDM is considered the state-of-the-art model:

- SVM-BOW: an SVM classifier using bag-of-words and N-gram (e.g., 1-gram, bigram, and trigram) features (Ma et al., 2018).
- CNN: a convolutional neural model for obtaining the representation of each tweet and classifying tweets with a softmax layer (Y.-C. Chen et al., 2017).
- BiLSTM: a bidirectional RNN-based tweet model that considers the bidirectional contexts between targets and tweets (Augenstein et al., 2016).
- BERT: a fine-tuned BERT to detect rumors (Devlin et al., 2019)
- RDM: a method integrating GRU and reinforcement learning to detect rumors at an early stage (K. Zhou et al., 2019).

Table 2 shows that our model outperforms the other models, including the state-of-the-art model, RDM. The SVM uses bag-of-words features for text encoding and statistical methods that miss essential text features, which leads to low rumor detection accuracy. CNN, BiLSTM, and RDM process tweets based on time, thus they lose valuable information from the conversation structure. BERT has a

Dataset	Method	Acc	Pre	Rec	F1
PHEME 2017	GNN-U	0.606	0.6005	0.602	0.602
	GNN-M	0.630	0.597	0.598	0.597
	GNN-MU	0.622	0.585	0.586	0.585
	GNN-S	0.737	0.734	0.735	0.734
	GNN-SB	0.816	0.837	0.823	0.830
	TRM	0.900	0.893	0.869	0.881
PHEME 2018	GNN-U	0.581	0.563	0.560	0.561
	GNN-M	0.588	0.565	0.559	0.562
	GNN-MU	0.646	0.625	0.628	0.626
	GNN-S	0.731	0.669	0.677	0.673
	GNN-SB	0.798	0.816	0.804	0.810
	TRM	0.919	0.892	0.923	0.907

Table 3. Conversation representation study

multilayer architecture that performs well in various NLP tasks; however, BERT also processes tweets based on time; therefore, it has the same disadvantage as the other models. On the other hand, our model benefits from the utilization of conversation structure, where every reply supports, denies or comments on the source message. The results show that exploiting conversation structure enables improved rumor detection performance.

4.4 Conversation representation impact study

To further investigate the impact of the tweet representation used in our model, we evaluated several conversation information extraction approaches by replacing the TRM module with hand-crafted features and sentence embedding, namely: (1) GNN-U: user features; (2) GNN-M: message features; (3) GNN-MU: both user and message features; (4) GNN-S: fastText sentence embedding with GloVe 840B embeddings; and (5) GNN-SB: BERT (feature-based) sentence embedding.

As shown in Table 3, the performance of the first three models based on hand-crafted features is low, as characterized by accuracies of less than 68%. This low performance indicates that hand-crafted features with small dimensions are unlikely to represent all the helpful information in a conversation.

On the other hand, the sentence embedding methods GNN-S and GNN-SB achieve higher performance than the first three models. GNN-S outperforms GNN-MU by 7.62% on the PHEME 2017 dataset and 5.61% on the PHEME 2018 dataset, while GNN-SB achieves 15.35% and 13.95% higher accuracies on the PHEME 2017 and PHEME 2018 datasets, respectively. These results show that information propagation through the conversation structure based on the text can generate a more comprehensive representation.

Overall, we found that TRM as a Twitter conversation representation achieves the best performance because it was built to perform well on the rumor detection task. Unlike the other representations, TRM can learn through its loss function that can change the way posts are represented.

4.5 Early detection performance

Detecting rumors at an early stage of propagation is vital so that interventions can be carried out as soon as possible.

To evaluate our model’s performance on the early detection of rumors, we created eight test sets reflecting real scenarios of rumors spreading on Twitter. Unlike other researchers who define early rumors based on time, we define them based on the number of replies. We claim that, on the one hand, a small number of replies to a source tweet means that the rumor has just begun to spread because only a few tweets refer to it. On the other hand, a large number of replies to a source tweet suggests that the rumor has been widely spread.

The detection capability of the SVM is deficient in all the cases, achieving accuracies of less than 70%. Improved results of approximately 80% accuracy are obtained by CNN and BiLSTM. Moreover, CNN and BiLSTM exhibit higher accuracy when they use only a few replies (less than ten replies). In

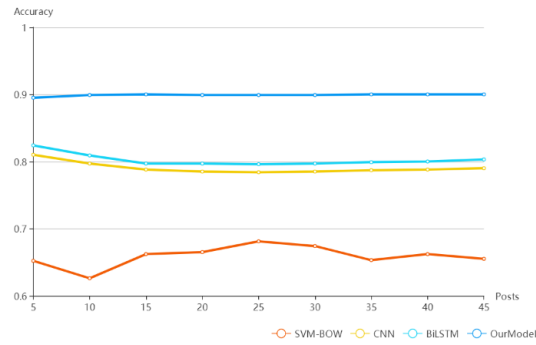


Figure 5. Early rumor detection result

Figure 5, we see that the accuracies of CNN and BiLSTM slightly decrease as the number of replies increases. This result occurs because CNN and BiLSTM ignore conversation structure information and use a time-based chain structure. As a result, CNN and BiLSTM are unable to encode correct representations when the number of replies is greater than ten.

Unlike the other models, our model benefits from encoding information based on conversation structure, thus obtaining the correct representation. The results show that, in every case, our model outperforms the other methods, suggesting that our models can classify rumors at very early or late stages.

5 Conclusions

We introduce a novel model that shows how the conversation structure of social media can help detect rumors. By viewing conversation structure as a graph, we propagate a message through its structure and benefit from users' related stances. In experiments using the PHEME 2017 and PHEME 2018 datasets, which contain only small amounts of data, our model outperforms the baseline and state-of-the-art models. We expect that our model's performance will increase when it is trained on larger datasets.

Moreover, since the CPM module can aggregate information from local neighbors and create a representation of unseen nodes, our model can be used for unsupervised tasks because it enables nearby nodes to have a similar representation while enforcing the highly distinct representation of the disparate node. In the future, we will investigate unsupervised models on massive amounts of unlabeled rumor data from social media.

Acknowledgements

We thank the anonymous reviewers for their feedback. This work was partially supported by the project MOST 109-2221-E-006-173 funded by Ministry of Science and Technology, Taiwan. We thank to National Center for High-performance Computing (NCHC) for providing computational and storage resources.

References

- Ajao, O., Bhowmik, D., & Zargari, S. 2018. Fake News Identification on Twitter with Hybrid CNN and RNN Models. *Proceedings of the 9th International Conference on Social Media and Society - SMSociety '18*, 226–230.
- Augenstein, I., Rocktäschel, T., Vlachos, A., & Bontcheva, K. 2016. Stance Detection with Bidirectional Conditional Encoding. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 35(6), 876–885.
- Belkaroui, R., Faiz, R., & Elkhilfi, A. 2014. Conversation Analysis on Social Networking Sites. *2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems*, 172–178.
- Castillo, C., Mendoza, M., & Poblete, B. 2011. Information credibility on Twitter. *Proceedings of the 20th International Conference Companion on World Wide Web, WWW 2011*, 675–684.
- Chen, T., Li, X., Yin, H., & Zhang, J. 2018. Call Attention to Rumors: Deep Attention Based Recurrent Neural Networks for Early Rumor Detection. In *Lecture Notes in Computer Science: Vol. 11154 LNAI* (pp. 40–52).

- Chen, Y.-C., Liu, Z.-Y., & Kao, H.-Y. 2017. IKM at SemEval-2017 Task 8: Convolutional Neural Networks for stance detection and rumor verification. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, (2011), 465–469.
- Cogan, P., Andrews, M., Bradonjic, M., Kennedy, W. S., Sala, A., & Tucci, G. 2012. Reconstruction and analysis of Twitter conversation graphs. *Proceedings of the First ACM International Workshop on Hot Topics on Interdisciplinary Social Networks Research - HotSocial '12*, 25–31.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.
- Guo, H., Cao, J., Zhang, Y., Guo, J., & Li, J. 2018. Rumor Detection with Hierarchical Social Attention Network. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management - CIKM '18*, 943–951.
- Hamilton, W. L., Ying, R., & Leskovec, J. 2017. Inductive representation learning on large graphs. *31st Conference on Neural Information Processing Systems (NIPS 2017)*.
- Holcomb, J., Gottfried, J., Mitchell, A., & Schillinger, J. 2013. News use across social media platforms. Retrieved September 3, 2019, from Pew Research Center website: <http://www.journalism.org/2013/11/14/news-use-across-social-media-platforms/>
- Liu, X., Nourbakhsh, A., Li, Q., Fang, R., & Shah, S. 2015. Real-time Rumor Debunking on Twitter. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management - CIKM '15*, 1867–1870.
- Lukasik, M., Srijith, P. K., Vu, D., Bontcheva, K., Zubiaga, A., & Cohn, T. 2016. Hawkes Processes for Continuous Time Sequence Classification: an Application to Rumour Stance Classification in Twitter. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 393–398.
- Ma, J., Gao, W., Mitra, P., Kwon, S., Jansen, B. J., Wong, K. F., & Cha, M. 2016. Detecting rumors from microblogs with recurrent neural networks. *IJCAI International Joint Conference on Artificial Intelligence, 2016-Janua*, 3818–3824.
- Ma, J., Gao, W., & Wong, K.-F. 2018. Rumor Detection on Twitter with Tree-structured Recursive Neural Networks. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 1*, 1980–1989.
- Magnani, M., Montesi, D., Nunziante, G., & Rossi, L. 2011. Conversation Retrieval from Twitter. In *Lecture Notes in Computer Science: Vol. 6611 LNCS* (pp. 780–783).
- Nguyen, T. N., Li, C., & Niederée, C. 2017. On Early-Stage Debunking Rumors on Twitter: Leveraging the Wisdom of Weak Learners. In *Lecture Notes in Computer Science: Vol. 10540 LNCS* (pp. 141–158).
- Pace, S., Buzzanca, S., & Fratocchi, L. 2016. The structure of conversations on social networks: Between dialogic and dialectic threads. *International Journal of Information Management*, 36(6), 1144–1151.
- Rath, B., Gao, W., Ma, J., & Srivastava, J. 2017. From retweet to believability: Utilizing trust to identify rumor spreaders on twitter. *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2017*, 179–186.
- Wu, K., Yang, S., & Zhu, K. Q. 2015. False rumors detection on Sina Weibo by propagation structures. *2015 IEEE 31st International Conference on Data Engineering, 2015-May*, 651–662.
- Xu, K., Jegelka, S., Hu, W., & Leskovec, J. 2019. How powerful are graph neural networks? *7th International Conference on Learning Representations, ICLR 2019*. arXiv: 1810.00826
- Yang, F., Liu, Y., Yu, X., & Yang, M. 2012. Automatic detection of rumor on Sina Weibo. *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics - MDS '12, 2*, 1–7.
- Yu, F., Liu, Q., Wu, S., Wang, L., & Tan, T. 2019. Attention-based convolutional approach for misinformation identification from massive and noisy microblog posts. *Computers & Security*, 83, 106–121.
- Zhao, Z., Resnick, P., & Mei, Q. 2015. Enquiring Minds. *Proceedings of the 24th International Conference on World Wide Web - WWW '15*, 1395–1405.
- Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., ... Sun, M. 2018. *Graph Neural Networks: A Review of Methods and Applications*. arXiv: 1812.08434.

- Zhou, K., Shu, C., Li, B., & Lau, J. H. 2019. Early Rumour Detection. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1614–1623.
- Zubiaga, A., Aker, A., Bontcheva, K., Liakata, M., & Procter, R. 2018. Detection and Resolution of Rumours in Social Media. *ACM Computing Surveys*, 51(2), 1–36.
- Zubiaga, A., Liakata, M., & Procter, R. 2016. *Learning Reporting Dynamics during Breaking News for Rumour Detection in Social Media*. arXiv: 1610.07363
- Zubiaga, A., Liakata, M., & Procter, R. 2017. Exploiting Context for Rumour Detection in Social Media. In A. Jatowt, E.-P. Lim, Y. Ding, A. Miura, T. Tezuka, G. Dias, ... B. T. Dai (Eds.), *Lecture Notes in Computer Science: Vol. 10539 LNCS* (pp. 109–123).
- Zubiaga, A., Liakata, M., Procter, R., Wong Sak Hoi, G., & Tolmie, P. 2016. Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads. *PLOS ONE*, 11(3), e0150989.