Debunking Rumors on Twitter with Tree Transformer

Jing Ma
Dept. of Computer Science
Hong Kong Baptist University
Hong Kong
majing@comp.hkbu.edu.hk

Wei Gao
School of Information Systems
Singapore Management University
Singapore
weigao@smu.edu.sg

Abstract

Rumors are manufactured with no respect for accuracy, but can circulate quickly and widely by "word-of-post" through social media conversations. Conversation tree encodes important information indicative of the credibility of rumor. Existing conversation-based techniques for rumor detection either just strictly follow tree edges or treat all the posts fully-connected during feature learning. In this paper, we propose a novel detection model based on tree transformer to better utilize user interactions in the dialogue where post-level self-attention plays the key role for aggregating the intra-/inter-subtree stances. Experimental results on the TWITTER and PHEME datasets show that the proposed approach consistently improves rumor detection performance.

1 Introduction

Online rumor perhaps is one of the most prevalent social diseases in the era of social media. An immediate example we are witnessing is the unprecedented information disorder represented by various rumors, conspiracy theories, hoaxes, fake news, etc. in parallel with the worldwide pandemic of COVID19. In different places, a number of people were hospitalized or even died for drinking bootleg alcohol to prevent coronavirous infection, resulting from a false rumor attack on gullible public claiming that "smoking, methanol or cocaine can cure for the virus"[1]. Automatic rumor debunking is at the core of battle against such massive disorder of information especially in the midst of crisis.

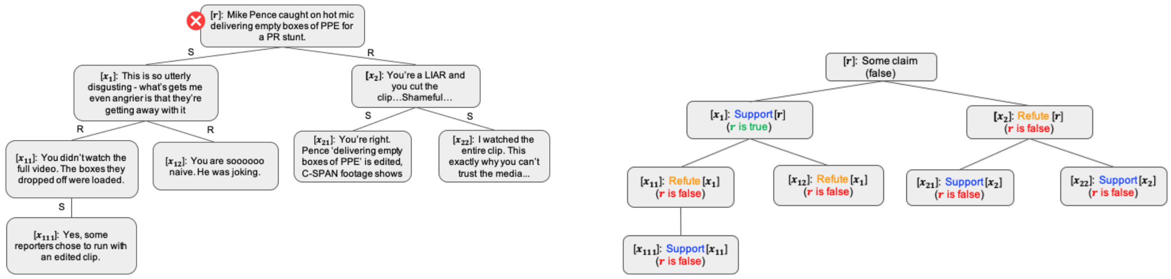
Rumor debunking aims to determine the veracity of a given topic or a claim. Fact-checking websites, such as snopes.com and politifact.com, employ manual verification and investigative journalism, which is prone to low efficiency and poor coverage. For automated approaches, prior studies focus on engineering or learning features from sequential microblog streams (Castillo et al., 2011; Yang et al., 2012; Kwon et al., 2013; Liu et al., 2015; Ma et al., 2015; Ma et al., 2016; Yu et al., 2017). More recently, structure-based learning based on structured neural networks are proposed to capture the interactive characteristics of rumor diffusion, such as tree kernel (Ma et al., 2017), recursive neural network (Ma et al., 2018) and tree LSTM model (Kumar and Carley, 2019). Khoo et al. (2020) proposed to model potential dependencies between any two microblog posts with the post-level self-attention networks (PLAN), which has achieved the state-of-the-art detection performance.

The PLAN model essentially treats the input tweets as a fully connected graph, by assuming that a user may not be directed solely at the tweet being replied considering the content created could also be applicable to other tweets in the thread (Khoo et al., 2020). Also, the representation of posts is enhanced by leveraging the strength of transformer's encoding architecture. Nevertheless, we argue that such full connection which ignores the specific targets of replies in the hierarchy could create salient issues on post representation learning, especially in the vein of relatively deep conversation or argument. Meanwhile, other existing tree-structured models based on propagation trees (Wu et al., 2015; Ma et al., 2017) or recursive trees (Ma et al., 2018; Kumar and Carley, 2019) tend to oversimplify user interaction by genuinely following the tree edges for post matching or encoding.

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

[1] https://time.com/5828047/methanol-poisoning-iran/

Proceedings of the 28th International Conference on Computational Linguistics, pages 5455–5466
Barcelona, Spain (Online), December 8-13, 2020


[r]: Mike Pence caught on hot mic delivering empty boxes of PPE for a PR stunt.
S
R
[x_1]: This is so utterly disgusting - what's gets me even angrier is that they're getting away with it
[x_2]: You're a LIAR and you cut the clip…Shameful…
R
R
S
S
[x_{11}]: You didn't watch the full video. The boxes they dropped off were loaded.
[x_{12}]: You are soooooo naive. He was joking.
[x_{21}]: You're right. Pence 'delivering empty boxes of PPE' is edited, C-SPAN footage shows
[x_{22}]: I watched the entire clip. This exactly why you can't trust the media...
S
[x_{111}]: Yes, some reporters chose to run with an edited clip.
[r]: Some claim (false)
[x_1]: Support [r] (r is true)
[x_2]: Refute [r] (r is false)
[x_{11}]: Refute [x_1] (r is false)
[x_{12}]: Refute [x_1] (r is false)
[x_{21}]: Support [x_2] (r is false)
[x_{22}]: Support [x_2] (r is false)
[x_{111}]: Support [x_{11}] (r is false)

(a) Propagation tree of a false rumor

(b) Relative stance and underlying veracity patterns

Figure 1: A motivating example: A false rumor about "Mike Pence delivering empty boxes of PPE for PR stunt" widely spread on Twitter and the stances relative to parent nodes implying the underlying credibility of the claim.

To illustrate our intuition, Figure 1 exemplifies the propagation structure of a (rumor) claim "Mike Pence caught on hot mic delivering empty boxes of PPE for a PR stunt". The PLAN model basically assumes each user directed at all the tweets in the thread, which may be true for a shallow tree where most of nodes respond to the root node. However, this is not the case when it comes to a tree hierarchy as Figure 1 shows. It can be seen that accurate viewpoints is generally associated with the context of parent posts, e.g., x_{21} support x_2, but x_2 refute the source claim r, therefore x_{21} believe that the claim is false even though it contains a non-rumor-indicative patten "be right". On the other hand, x_{21} even has no context correlation with the nodes from another branch such as x_{12}. But the PLAN model might brought unexpected errors in this case when linking x_{21} with r (or x_{12}) when making fully pairwise comparison.

To this end, we propose to enhance the representation by exploring the stances towards the same target utilizing the associated contextual information. The starting point of our approach is an observation: each post in the propagation tree may trigger a set of responsive tweets (such as $x_1 \rightarrow [x_{11}, x_{12}]$ in Figure 1), we define such unit as a *subtree*, which eventually compose the whole tree hierarchy. Accordingly, we extend the conventional transformer's encoder into three variants, i.e., a bottom-up transformer, a top-down transformer, and a hybrid transformer model. More specifically, our models selectively attend over tweets in the same subtree. As a result, it can be expected that user's viewpoint can be fully captured based on the context of propagation path. Meanwhile, inaccurate information in a subtree can be cross-checked as users share opinions towards the same target (i.e., the subtree root). We construct two shallow tree datasets and two deep tree datasets referring from two publicly benchmarks TWITTER and PHEME. Extensive experimental results demonstrate that our approach consistently improve over the state-of-the-art rumor detection and early detection baselines, particularly performing well on the deep trees.

2 Related Work

This section firstly reviews the recent progress about rumor detection. Most previous automatic approaches for rumor detection (Castillo et al., 2011; Yang et al., 2012; Liu et al., 2015) intended to learn a supervised classifier by utilizing a wide range of features crafted from post contents, user profiles and propagation patterns. Subsequent studies were then conducted to engineer new features such as those representing rumor diffusion and cascades (Kwon et al., 2013; Friggeri et al., 2014; Hannak et al., 2014). Ma et al. (2015) extended their model with a large set of chronological social context features. These approaches typically require heavy preprocessing and feature engineering.

Zhao et al. (2015) alleviated the engineering effort by using a set of regular expressions (such as "really?", "not true", etc) to find questing and denying tweets, but the approach was oversimplified and suffered from very low recall. Ma et al. (2016) and Yu et al. (2017) respectively used recurrent neural networks (RNNs) and convolutional neural networks (CNNs) to learn automatically the representations from tweets content based on time series. Guo et al. (2018) proposed a hierarchical attention model which captures important clues from social context of a rumorous event at the post and sub-event levels. Jin et al. (2016) exploited the conflicting viewpoints in a credibility propagation network for verifying news stories propagated among the tweets. However, these approaches cannot embed features reflecting how the posts are propagated and requires careful data segmentation to prepare for the time sequence.

Some kernel-based methods were exploited to model the propagation structure. Wu et al. (2015) proposed a hybrid SVM classifier which combines a RBF kernel and a random-walk-based graph kernel to capture both flat and propagation patterns for detecting rumors on Sina Weibo. Ma et al. (2017) used tree kernel to capture the similarity of propagation trees by counting their similar substructures in order to identify different types of rumors on Twitter. Ma et al. (2018) presented tree-structured *recursive neural networks* (RvNN) to jointly generate the representation of a propagation tree based on the posts content and their propagation structure.

In recent years, transformer (Vaswani et al., 2017) have demonstrates state-of-the-art performance in a variety of NLP tasks such as machine translation (Vaswani et al., 2017), sentence representation (Devlin et al., 2019), generative dialog (Tao et al., 2018), machine reading (Cheng et al., 2016), semantic labeling (Strubell et al., 2018), and rumor detection (Khoo et al., 2020). Transformer produce strong power of representations by applying attention to each pair of elements from an input sequence, regardless of their distance. Khoo et al. (2020) propose a rumor verification model that allows direct modeling of dependencies between any two posts without regarding to their responsive relation, thus it essentially treats the propagation as a fully connected graph instead of a tree. Our work is inspired by the idea of improving the representation power of transformer to model structured objects such as syntactic parse tree. In these works, a straightforward strategy is to augment the conventional transformer with structural positional embeddings (Wang et al., 2019a; Shiv and Quirk, 2019). On the other hand, Tree Transformer is proposed to attend over nearer neighbor nodes (Ahmed et al., 2019; Wang et al., 2019b). Our proposed method is a substantial extension of Tree Transformer for modeling propagation tree structures for detecting rumors on microblogging websites.

3 Problem Statement and Notations

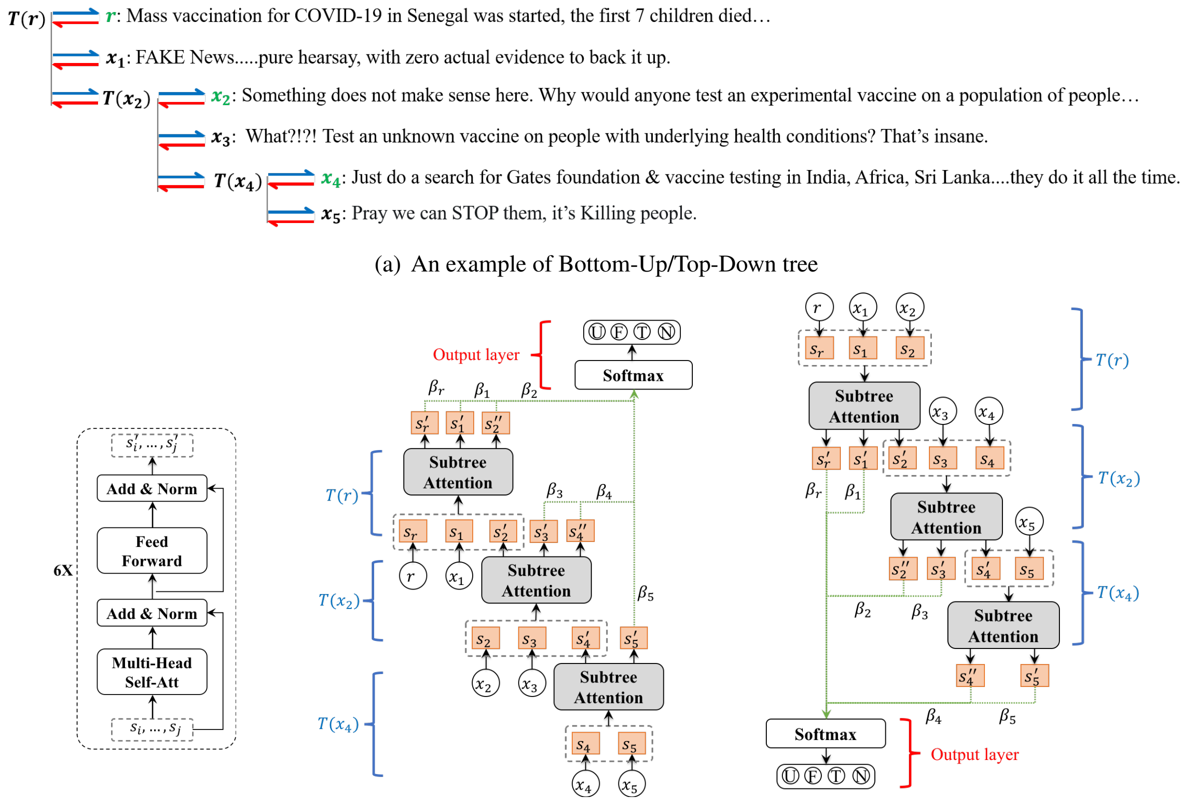
On microblogging platforms such as Twitter, the follower/friend relationship embeds shared interests among the users. Once a user has posted a tweet, all his followers will receive it. Twitter allows a user to retweet or comment on another user‘s post, so that the information could reach beyond the followers of the original creator. Therefore, we model the propagation of each claim as a tree structure $\mathcal{T}(r) = \langle V, E \rangle$, where r is tree root representing the source tweet that states the claim, V refers to a set of nodes each representing a responding post of r in the thread of the circulation, and E is a set of directed edges corresponding to the response relation among the nodes in V. Inspired by (Ma et al., 2018), here we consider two different propagation trees with distinct edge directions: (1) *Bottom-Up tree* where the responsive nodes point to their responded nodes, similar to a citation network; and (2) *Top-Down tree* where the edge follows the direction of information diffusion by reversing the Bottom-up tree.

We formulate this task as a supervised classification problem, which learns a classifier f from the labeled claims, that is, $f : C_i \rightarrow Y_i$, where Y_i takes one of the four categories: *N̲on-rumor, T̲rue rumor, F̲alse rumor* and *U̲nverified rumor* (NTFU), that are introduced in previous literature (Zubiaga et al., 2016b; Ma et al., 2017).

4 Tree Transformer Model for Rumor Detection

Rumor indicative features can be captured from propagation structures, e.g., the stances expressed in responsive tweets can further reinforce the stances of that tweet is replying to (Ma et al., 2018; Kumar and Carley, 2019), the posts with strong stance based on the tree branch is more important when determining the rumor veracity (Li et al., 2019), and inaccurate information might be "self-checked" by making comparison with correlative tweets (Zubiaga et al., 2018). However, such relation is not fully exploited by previous work. Our core idea is to enhance representation learning of rumor indicative features by selectively attending over the corresponding tweets, that deeply explore user opinions and refine inaccurate information following the propagation tree structure.

Unlike the PLAN model that rawly handcraft 5 types of responsive relation as an additional consideration when attending over all the other tweets, our idea and the adopted mechanisms are significantly different.


T(r)
r: Mass vaccination for COVID-19 in Senegal was started, the first 7 children died…
x1: FAKE News.....pure hearsay, with zero actual evidence to back it up.
T(x2)
x2: Something does not make sense here. Why would anyone test an experimental vaccine on a population of people…
x3: What?!?! Test an unknown vaccine on people with underlying health conditions? That's insane.
T(x4)
x4: Just do a search for Gates foundation & vaccine testing in India, Africa, Sri Lanka....they do it all the time.
x5: Pray we can STOP them, it's Killing people.
6X
Add & Norm
Feed Forward
Add & Norm
Multi-Head Self-Att
Output layer
Softmax
U F T N
Subtree Attention
T(r)
T(x2)
T(x4)
β_r β_1 β_2 β_3 β_4 β_5

(a) An example of Bottom-Up/Top-Down tree

(b) Subtree Attention

(c) Bottom-up Transformer

(d) Top-down Transformer

Figure 2: Illustration of a propagation tree and the corresponding tree transformer models. In Figure 2(a), $T(\cdot)$ denote a subtree rooted by the node in green, that we put at the first line of the subtree. The edges in red and blue apply the Bottom-Up and Top-Down tree respectively.

Figure 2 gives an overview of our transformer-based framework respectively based on Bottom-Up tree and Top-Down tree, which will be depicted in detail in the subsections.

4.1 Token-Level Tweet Representation

Given a tweet represented as a word sequence $x_i = (w_1 \cdots w_t \cdots w_{|x_i|})$, each $w_t \in \mathbb{R}^d$ is a d-dimensional vector which can be initialized with pre-trained word embeddings. We map each w_t into a fixed-sized hidden vector using Multi-Head Self-Attention networks(MH-SAN), which are the defaults setting in Transformer encoder (Vaswani et al., 2017). The core idea of MH-SAN is to jointly attend to words from different representation subspaces at different positions. More specifically, MH-SAN firstly transform the input word sequence x_i into multiple subspaces with different linear projections:

$$Q_i^h, K_i^h, V_i^h = x_i \cdot W_Q^h, \quad x_i \cdot W_K^h, \quad x_i \cdot W_V^h \tag{1}$$

where $\{Q_i^h, K_i^h, V_i^h\}$ are respectively the query, key and value representations, and $\{W_Q^h, W_K^h, W_V^h\}$ denote parameter matrices associated with the h^{th} head. Then, attention functions are applied to generate the output states.

$$O_i^h = \text{softmax}(\frac{Q_i^h \cdot {K_i^h}^\top}{\sqrt{d_h}}) \cdot V_i^h \tag{2}$$

Here $\sqrt{d_h}$ is the scaling factor and d_h represent the dimensionality of the h^{th} head subspace. Finally, the output of representation could be regard as a concatenation of all the heads $O_i = [O_i^1, O_i^2, \cdots, O_i^n] \in \mathbb{R}^{|x_i| \times d}$ with n as the number of heads, which followed by a normalization layer (layerNorm) and a feed-forward network (FFN) consistant with the usage of Transformer.

$$\begin{aligned} B_i &= \mathrm{layerNorm}(O_i \cdot W_B + O_i) \\ H_i &= \mathrm{FFN}(B_i \cdot W_S + B_i) \end{aligned} \tag{3}$$

where $H_i = [h_1; \dots; h_{|x_i|}] \in \mathbb{R}^{|x_i| \times d}$ is the matrix representing all words in tweet x_i, and W_B and W_h contain the weights of the transformation. Finally, we obtain the representation of x_i by maxpooling the vectors of all involved words:

$$s_i = \text{max-pooling}(h_1, \dots, h_{|x_i|}) \tag{4}$$

where $s_i \in \mathbb{R}^{1 \times d}$ is a d-dimensional vector, and $|\cdot|$ denotes the number of words.

4.2 Post-Level Tweet Representation

Previous literature has generally found that each node in the tree can trigger a set of responsive posts, i.e., a *subtree*, which contain distinct rumor-indicative pattens (Ma et al., 2017). Our goal is to cross-check all the posts in the same subtree to enhance the representation learning, because: (1) posts are generally short in nature thus the stance expressed in each node is closely correlated with the responsive context; and (2) posts in the same *subtree* direct at the individual opinion expressed in the root of the *subtree*. Thus coherent opinions can be captured by comparing ALL responsive posts in the same *subtree*, that lower weight the incorrect information (e.g., the supportive posts towards a false claim).

To this end, we propose to utilize transformer-based network to make pairwise comparison within a subtree, that capture users‘ opinions and enhance the representation for each node. In this paper, we develop two structures respectively based on Bottom-Up tree and Top-Down tree:

Bottom-Up Transformer. In Bottom-Up tree, we visit the root of each *subtree* from the leaf node hierarchically until reaching the source tweet. We propose a Bottom-Up transformer to capture coherent attitudes towards each tree node, by making pairwise comparison among its responsive tweets.

Figure 2(c) illustrated the structure of our tree transformer that cross-check all the posts from the bottom subtree to the upper subtrees. Specifically, given a subtree rooted at x_j, Let $\mathcal{V}(j) = \{x_j, \dots, x_k\}$ denote the set of node in the subtree, i.e., x_j and its direct response nodes. Then we apply a post-level subtree attention (i.e., a transformer-based block as shown in Figure 2(b)) on $\mathcal{V}(j)$ to get the refined representation for each node in $\mathcal{V}(j)$:

$$[s'_j; \dots; s'_k] = \textsc{Trans}([s_j; \dots; s_k], \Theta_T) \tag{5}$$

where $\textsc{Trans}(\cdot)$ is the transform function that has similar forms as shown in Eq. 2-4, and Θ_T contains the transformer parameters. Thus s'_* is the refined representation of s_* obtained based on the context of subtree. Note that each node can be treated as either parent or child in different subtrees, e.g., in Figure 2(a), x_2 can either be the parent node of $T(x_2)$, or a child node of $T(r)$. As a result, a part of nodes in our model are refined twice hierarchically from bottom subtree to upper subtree, that: (1) capture stances by comparing with parent node, and (2) lower-weight inaccurate information by attending over neighbor nodes, e.g., a parent that support a false claim might be refined if the majority responses refute the parent node.

Top-Down Transformer. This model is designed to leverage the structure of Top-Down tree, which is shown in Figure 2(d). Since Top-Down tree models how the information flows from source post to the current node, our model visits each subtree hierarchically from the source node until the leaf nodes. The transformer mechanism shares the similar intuition as the Bottom-Up transformer, thus node representation is enhanced by capturing stances and self-corrected context information.

4.3 The overall Model

To jointly capture the opinions expressed in the whole tree, we utilize an attention layer to select important posts with accurate information, which is obtained based on the refined node representation. This yields:

$$\begin{aligned} \alpha_i &= \frac{\exp(s'_i \cdot \mu^\top)}{\sum_j \exp(s'_j \cdot \mu^\top)} \\ \tilde{s} &= \sum_i \alpha_i \cdot s'_i \end{aligned} \tag{6}$$

where s'_i is obtained from either Bottom-Up Transformer or Top-Down Transformer[2], and $\mu \in \mathbb{R}^{1 \times d}$ contains the weights of the transformation. Here α_i is the attention weight of node x_i which is used to produce the representation $\tilde{s}$ of an entire tree. Lastly, we use a fully connected output layer to predict the probability distribution over the rumor classes.

$$\hat{y} = \mathrm{softmax}(V_o \cdot \tilde{s} + b_o) \tag{7}$$

where V_o and b_o are the weights and bias in the output layer.

Furthermore, there is a straightforward way to concatenate the tree representation from the Bottom-Up transformer, with that from the Top-Down transformer to obtain a richer tree representation, which is then fed into the above $\mathrm{softmax}(\cdot)$ function to make rumor predictions.

Model Training. All our models are trained to minimize the squared error between the probability distribution of the prediction and that of the ground truth:

$$L(y, \hat{y}) = \sum_{n=1}^{N} \sum_{c=1}^{C} (y_c - \hat{y}_c)^2 + \lambda ||\Theta||_2^2 \tag{8}$$

where y_c is the ground-truth label and $\hat{y}_c$ is the predicted probability of class c, N is the number of trees for training, C is the number of classes, $||.||_2$ is the L_2 regularization term over all the model parameters Θ, and λ is the trade-off coefficient.

During training, parameters are updated through back-propagation (Collobert et al., 2011) with AdaGrad (Duchi et al., 2011) for speeding up convergence. The training process ends when the model converges or the maximum epoch number is met. We represent input words using pre-trained GloVe Wikipedia 6B word embeddings (Pennington et al., 2014). We set model dimension d to 300 and the dimension for feedforward network is 600. We used 1 and 6 layers of transformer encoder for token-level representation and post-level representation respectively, and set the head number n as 12. The learning rate is initialized as 0.01, and the dropout rate is 0.2.

5 Experiments and Results

5.1 Datasets

For experimental evaluation, we refer two publicly available tree datasets released by (Ma et al., 2017) and (Kochkina et al., 2018), namely TWITTER and PHEME. In each dataset, a group of source tweets, which form the tree roots, together with their replies are provided in the form of tree structure. Each tree is annotated with one of the four class labels, i.e., non-rumor, true rumor, unverified rumor and false rumor.

To evaluate the robustness of our tree structured detection methods, we consider two types of datasets: propagation trees with shallow depth and trees with deep depth (i.e., complex responsive relations). Therefore, we regroup the trees in each of the datasets into TWO according to the tree depth. Specifically, we split Twitter (PHEME) dataset into TWITTER-S (PHEME-S) and TWITTER-D (PHEME-D), comprised by shallow trees and deep trees respectively. Table 1 displays the basic statistics of the four datasets.

[2] It can be s''_i if the node is refined twice, e.g., s''_2 in Figure 2(c) and 2(d).

Table 1: Statistics of the datasets

Statistic	TWITTER-S	TWITTER-D	PHEME-S	PHEME-D
# of source tweets	1,293	813	1,946	1,842
# of tree nodes	29,142	72,365	28,550	60,943
# of non-rumors	334	244	1,231	1,248
# of false	326	231	167	134
# of true	389	183	310	277
# of unverified	244	155	238	183
Avg. time length / tree	212 Hours	628 Hours	15 Hours	25 Hours
Avg. depth / tree	1.82	8.02	2.39	7.71
Max depth / tree	3	84	3	47
Min depth / tree	1	4	2	4
Avg. # of posts / tree	22	89	14	33
Max # of posts / tree	203	788	51	346
Min # of posts / tree	2	7	3	5

5.2 Experimental Setup

For evaluation, we will make comprehensive comparisons between our proposed models and state-of-the-art baselines on rumor classification and early detection tasks.

- DT-Rank: (Zhao et al., 2015) proposed a Decision-Tree-based Ranking model to identify trending rumors by searching for inquiry phrases.

- DTC: An information credibility model using a Decision-Tree Classifier (Castillo et al., 2011) using hand-crafted features that are based on the overall statistics of the posts without temporal information.

- RFC: A Random Forest Classfier which used three fitting parameters as temporal properties and a set of hand-crafted features based on user, linguistic and structural properties (Kwon et al., 2013).

- SVM-TK: A SVM classifier that uses a Tree Kernel (Ma et al., 2017) which try to capture propagation structure via kernel learning.

- GRU-RNN: A rumor detection model based on recurrent neural networks (Ma et al., 2016) with GRU for learning rumor representations by modeling sequential structure of relevant posts.

- BU-RvNN and **TD-RvNN**: The rumor detection models respectively based on bottom-up and top-down RvNN models (Ma et al., 2018) for integrating tweet contents and structure clues.

- PLAN: A rumor detection model based on transformer networks (Khoo et al., 2020) to model long distance interactions between any pair of tweets that oversimplifies responsive relations.

- BU-Trans, **TD-Trans** and **HD-Trans** : Our proposed tree transformer models respectively with Bottom-Up, Top-Down and Hybrid manner (see Section. 4).

We implement **DT-Rank**, **DTC** and **RFC** using Weka[3], **SVM-TK** using LibSVM[4] and all neural-network-based models with pytorch[5]. We use micro-averaged and macro-averaged F1 score, and class-specific F-measure as evaluation metrics. We hold out 10% of the datasets for tuning the hyper parameters, and conduct 5-fold cross-validation on the rest of the datasets.

5.3 Rumor Classification Performance

Table 2 demonstrate the performance of all the compared methods respectively based on the shallow trees and deep trees from TWITTER and PHEME datasets. The results indicate that our proposed methods outperform all the baselines[6], confirming the advantages of Tree transformer for rumor detection task.

It is observed that the performances of the three baselines in the first group based on handcrafted features are obviously poor. RFC perform relatively better because of the usage of additional temporal traits. Among the baselines without feature engineering in the second group, the sequential neural model GRU-RNN without considering structural information performs slightly worse than SVM-TK, because SVM-TK is an integrated kernel that utilize the propagation structure by comparing the trees based on

[3] `www.cs.waikato.ac.nz/ml/weka`

[4] `www.csie.ntu.edu.tw/~cjlin/libsvm`

[5] `pytorch.org`

[6] We use micF to evaluate TWITTER-S (D) datasets, but macF for PHEME-S (D) datasets owing to the imbalanced class prevalence (see Table 1), to capture competitive performance beyond the majority class (Zubiaga et al., 2016a)

Table 2: Results of comparison among different models. (NR: non-rumor; FR: false rumor; TR: true rumor; UR: unverified rumor)

(a) TWITTER-S (-D) dataset

Dataset	TWITTER-S						TWITTER-D					
Method			NR	FR	TR	UR			NR	FR	TR	UR
	micF	macF	F_1	F_1	F_1	F_1	micF	macF	F_1	F_1	F_1	F_1
DT-Rank	0.467	0.443	0.622	0.329	0.520	0.299	0.566	0.516	0.447	0.577	0.555	0.484
DTC	0.523	0.502	0.728	0.418	0.512	0.349	0.538	0.497	0.758	0.516	0.332	0.381
RFC	0.599	0.550	0.782	0.470	0.561	0.385	0.582	0.533	0.774	0.501	0.461	0.395
SVM-TK	0.719	0.714	0.705	0.683	0.785	0.682	0.669	0.663	0.698	0.649	0.689	0.615
GRU-RNN	0.715	0.701	0.700	0.697	0.780	0.620	0.646	0.645	0.645	0.624	0.714	0.598
BU-RvNN	0.738	0.728	0.734	0.672	0.825	0.681	0.698	0.699	0.674	0.693	0.741	0.687
TD-RvNN	0.749	0.738	0.724	0.729	0.830	0.684	0.705	0.704	0.725	0.677	0.759	0.656
PLAN	0.764	0.757	0.742	0.744	**0.840**	0.699	0.719	0.715	0.746	0.708	0.760	0.646
BU-TRANS	0.774	0.729	**0.750**	0.772	0.821	0.753	0.753	0.745	0.771	0.772	0.767	0.670
TD-TRANS	0.776	0.772	0.739	0.780	0.826	0.742	0.755	0.748	**0.778**	0.773	0.740	0.701
HD-TRANS	**0.789**	**0.787**	0.749	**0.784**	0.837	**0.776**	**0.768**	**0.764**	0.773	**0.781**	**0.783**	**0.721**

(b) PHEME dataset

Dataset	PHEME-S						PHEME-D					
Method			NR	FR	TR	UR			NR	FR	TR	UR
	micF	macF	F_1	F_1	F_1	F_1	micF	macF	F_1	F_1	F_1	F_1
DT-Rank	0.557	0.319	0.722	0.194	0.323	0.037	0.543	0.303	0.710	0.136	0.187	0.177
DTC	0.614	0.424	0.763	0.308	0.341	0.286	0.695	0.465	0.819	0.271	0.442	0.328
RFC	0.701	0.482	0.825	0.304	0.486	0.332	0.708	0.515	0.820	0.231	0.528	0.484
SVM-TK	0.771	0.656	0.864	0.610	0.629	0.523	0.776	0.639	0.862	0.578	0.572	0.578
GRU-RNN	0.765	0.632	0.872	0.698	0.574	0.384	0.781	0.610	0.868	0.629	0.510	0.393
BU-RvNN	0.775	0.649	0.862	0.622	0.592	0.523	0.789	0.651	0.877	0.606	0.583	0.546
TD-RvNN	0.783	0.668	0.874	0.607	**0.631**	0.561	0.786	0.667	0.881	0.634	0.648	0.508
PLAN	**0.800**	0.688	0.872	0.645	0.629	0.605	0.798	0.681	0.879	0.689	0.602	0.551
BU-TRANS	0.794	0.704	0.875	0.683	0.621	0.636	0.831	0.731	**0.908**	0.652	0.708	0.656
TD-TRANS	0.790	0.701	**0.881**	**0.730**	0.620	0.570	0.825	0.722	0.904	0.681	0.667	0.635
HD-TRANS	0.793	**0.710**	0.872	0.728	0.600	**0.644**	**0.839**	**0.745**	0.806	**0.715**	**0.755**	**0.702**

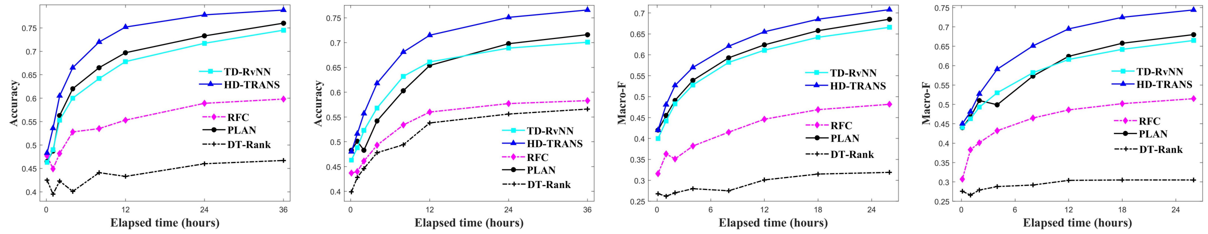
both textual and structural similarities. Tree-structured neural models, i.e., BU-RVNN and TD-RvNN, make further improvements since it deeply bridge the content semantics and propagation clues.

Among all the baselines, PLAN perform best since it leverage the representation power of transformer by modeling dependencies between any two tweets, but this may under-utilize the structural information. In contrast, our proposed TRANS-based models (in the third group), not only inherently leverage propagation structure but also take advantages of the representation power of transformer, thus beat PLAN on the four datasets. Among our three TRANS-based models, BU-TRANS and TD-TRANS perform comparable because both explore tree structure utilizing Transformer. And combing them makes further improvements as HD-TRANS did, suggesting that the learned pattens from the two models are complementary.

Furthermore, when drilling down to the performance of our TRANS-based models on specific datasets, we find that there are distinct observations of model performance between the shallow tree and deep tree. Specifically, on TWITTER-D and PHEME-D datasets, we observe the tree-based baselines (e.g., BU-RvNN and TD-RvNN) perform comparable to PLAN, and the improvements of our models over PLAN range from 5.31%−6.82% (7.34%−9.40%) accuracy (macroF score) on Twitter-D (PHEME-D). The reason is that PLAN is originally proposed and experimented on shallow trees (Khoo et al., 2020), which may not be generalize well on trees with deep and/or complex responsive relationships.

In comparison, on TWITTER-S and PHEME-S dataset, PLAN perform better than TD-RvNN (i.e., the best tree-structured baseline) in a larger margin, and our TRANS-based models improve over PLAN by 1.31%−3.27% (2.33%−3.20%) in terms of accuracy (macroF score) on TWITTER-S (PHEME-S) dataset, which is relatively lower than the improvements made on TWITTER-D and PHEME-D datasets. This is because the homogeneous edges (e.g., majority responsive nodes straightforwardly direct at the source post) in shallow trees have limited identical structure clues for rumor detection. This also verifies the hypothesis we made in Section 1 that tree-structured methods is more effective for deep trees.

5.4 Early Rumor Detection Performance

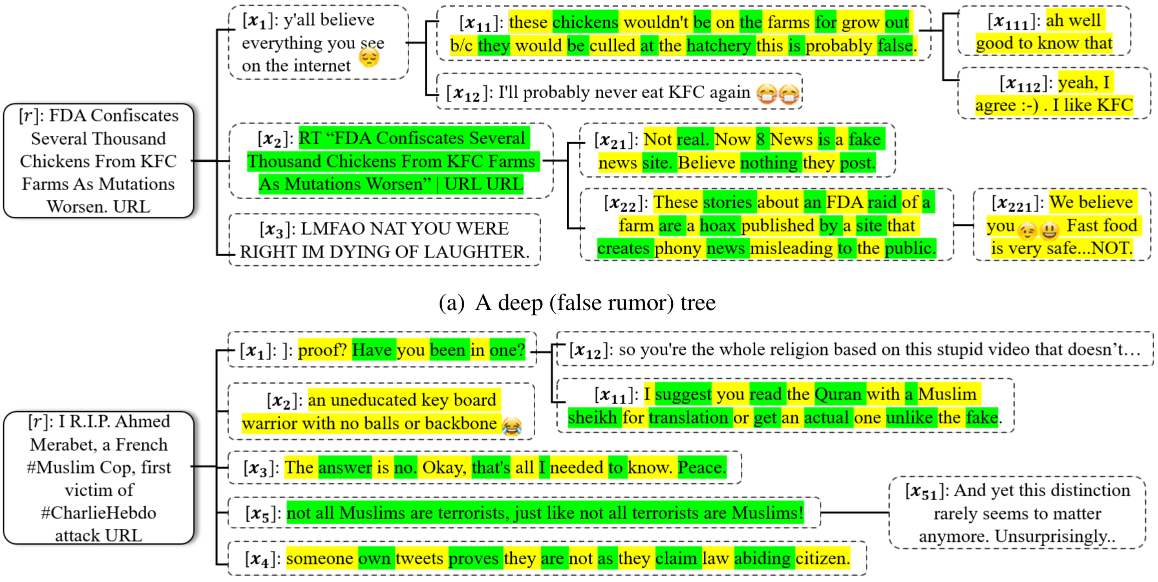


(a) TWITTER-S (b) TWITTER-D (c) PHEME-S (d) PHEME-D

Figure 3: Early rumor detection accuracy at different checkpoints in terms of elapsed time.

Debunking rumors at early stage of their propagation is very important so that preventive measures can be taken in a timely manner. In early rumor detection task, we compare different detection methods at a series of elapsed time checkpoints. Figure 3 shows the performance of our HD-TRANS model versus PLAN (the best performed baseline), TD-RvNN (the best tree-structured neural model), RFC (the best system based on feature engineering), and DT-Rank (an algorithm proposed for early rumor detection).

We observe that within the first few hours, the performance of our HD-TRANS model grows more quickly and starts to supersede the other models at the early stage of propagation. Particularly, HD-TRANS achieves 75.0% (72.3%) accuracy on TWITTER-S (-D) and 65.9% (69.5%) macF score on PHEME-S (-D) within 12 hours. Although all the methods are getting saturated as time goes by, HD-TRANS only need around 14 (12) hours on TWITTER-S (-D) and about 15 (10) hours on PHEME-S (-D), to achieve the comparable performance of the best baseline model (i.e., PLAN), indicating superior early detection performance of our method especially when comes to more complex or deeper propagation pattens.



[r]: FDA Confiscates Several Thousand Chickens From KFC Farms As Mutations Worsen. URL

[x_1]: y'all believe everything you see on the internet 😔

[x_{11}]: these chickens wouldn't be on the farms for grow out b/c they would be culled at the hatchery this is probably false.

[x_{111}]: ah well good to know that

[x_{112}]: yeah, I agree :-) . I like KFC

[x_{12}]: I'll probably never eat KFC again 😂😂

[x_2]: RT “FDA Confiscates Several Thousand Chickens From KFC Farms As Mutations Worsen” | URL URL

[x_{21}]: Not real. Now 8 News is a fake news site. Believe nothing they post.

[x_{22}]: These stories about an FDA raid of a farm are a hoax published by a site that creates phony news misleading to the public.

[x_{221}]: We believe you 😴😃 Fast food is very safe...NOT.

[x_3]: LMFAO NAT YOU WERE RIGHT IM DYING OF LAUGHTER.

(a) A deep (false rumor) tree

[r]: I R.I.P. Ahmed Merabet, a French #Muslim Cop, first victim of #CharlieHebdo attack URL

[x_1]:]: proof? Have you been in one?

[x_{12}]: so you're the whole religion based on this stupid video that doesn't…

[x_{11}]: I suggest you read the Quran with a Muslim sheikh for translation or get an actual one unlike the fake.

[x_2]: an uneducated key board warrior with no balls or backbone 😂

[x_3]: The answer is no. Okay, that's all I needed to know. Peace.

[x_5]: not all Muslims are terrorists, just like not all terrorists are Muslims!

[x_{51}]: And yet this distinction rarely seems to matter anymore. Unsurprisingly..

[x_4]: someone own tweets proves they are not as they claim law abiding citizen.

(b) A shallow (false rumor) tree

Figure 4: Examples of correctly detected false rumors at early stage of our model.

To get an intuitive understanding of what is happening when we use HD-TRANS model, we design an experiment to highlight the nodes with higher attention scores (i.e., “α_i” in Eq. 6) at the tree representation layer. Specifically, we sample two trees from TWITTER dataset, i.e., a shallow tree and a deep tree, at the early stage of propagation, that both have been correctly classified as false rumors by our HD-TRANS. In Figure 4, we observe that: 1) the highly ranked nodes with higher attention scores by HD-TRANS (in yellow), illustrated obvious structured rumor-indicative pattens, e.g., denial post spark affirmative replies as $x_{11} \rightarrow [x_{111}, x_{112}]$ shows in the deep tree; 2) the nodes attended by PLAN (in green) are generally independent of structure but taking coherent stances or semantics; and 3) the results of HD-TRANS and PLAN are significantly different on the deep tree, but similar results can be found on the shallow tree, implying that more complex propagation pattens can be better captured by our proposed model.

6 Conclusions and Future Work

In this paper, with the analysis that modeling propagation structure is an essential factor for detecting rumors, we propose three variants of transformer to further enhance the representation learning directed at tree-structured modeling: a Bottom-up transformer, a Top-down tranformer, and a Hybrid model. The results on four benchmark datasets confirm the advantages of our methods as compared to state-of-the-art baselines, especially well-generalized on trees with more complex responsive contexts. For future work, it is promising to include other types of edges/relationships besides the responsive relation to enhance rumor detection, such as friends/followers, quotation, mention, etc. We also plan to investigate the role of non-textual media such as images or videos on the effectiveness of detecting rumors.

References

Mahtab Ahmed, Muhammad Rifayat Samee, and Robert E Mercer. 2019. You only need attention to traverse trees. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 316–322.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of WWW*, pages 675–684.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 551–561.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Adrien Friggeri, Lada A Adamic, Dean Eckles, and Justin Cheng. 2014. Rumor cascades. In *Proceedings of ICWSM*.

Han Guo, Juan Cao, Yazi Zhang, Junbo Guo, and Jintao Li. 2018. Rumor detection with hierarchical social attention network. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 943–951. ACM.

Aniko Hannak, Drew Margolin, Brian Keegan, and Ingmar Weber. 2014. Get back! you don't know me like that: The social mediation of fact checking interventions in twitter conversations. In *Proceedings of ICWSM*.

Zhiwei Jin, Juan Cao, Yongdong Zhang, and Jiebo Luo. 2016. News verification by exploiting conflicting social viewpoints in microblogs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2972–2978. AAAI Press.

Ling Min Serena Khoo, Hai Leong Chieu, Zhong Qian, and Jing Jiang. 2020. Interpretable rumor detection in microblogs by attending to user interactions.

Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. PHEME dataset for Rumour Detection and Veracity Classification. 6.

Sumeet Kumar and Kathleen M Carley. 2019. Tree lstms with convolution units to predict stance and rumor veracity in social media conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5047–5058.

Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent features of rumor propagation in online social media. In *Proceedings of ICDM*, pages 1103–1108.

Quanzhi Li, Qiong Zhang, and Luo Si. 2019. Rumor detection by exploiting user credibility information, attention and multi-task learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1173–1179.

Xiaomo Liu, Armineh Nourbakhsh, Quanzhi Li, Rui Fang, and Sameena Shah. 2015. Real-time rumor debunking on twitter. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1867–1870.

Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. 2015. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1751–1754.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 3818–3824.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 708–717.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor detection on twitter with tree-structured recursive neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1980–1989.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Vighnesh Shiv and Chris Quirk. 2019. Novel positional encodings to enable tree-based transformers. In *Advances in Neural Information Processing Systems*, pages 12058–12068.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038.

Chongyang Tao, Shen Gao, Mingyue Shang, Wei Wu, Dongyan Zhao, and Rui Yan. 2018. Get the point of my utterance! learning towards effective responses with multi-head attention mechanism. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4418–4424.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2019a. Self-attention with structural position representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1403–1409.

Yaushian Wang, Hung-Yi Lee, and Yun-Nung Chen. 2019b. Tree transformer: Integrating tree structures into self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1060–1070.

Ke Wu, Song Yang, and Kenny Q Zhu. 2015. False rumors detection on sina weibo by propagation structures. In *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*, pages 651–662. IEEE.

Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. 2012. Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, MDS '12, pages 13:1–13:7.

Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2017. A convolutional approach for misinformation identification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3901–3907. AAAI Press.

Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1395–1405.

Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, and Michal Lukasik. 2016a. Stance classification in rumours as a sequential task exploiting the tree structure of social media conversations. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2438–2448.

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016b. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3):e0150989.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, 51(2):32.