Rumor Detection on Twitter with Tree-structured Recursive Neural Networks

Jing Ma[1], **Wei Gao**[2], **Kam-Fai Wong**[1,3]
[1]The Chinese University of Hong Kong, Hong Kong SAR
[2]Victoria University of Wellington, New Zealand
[3]MoE Key Laboratory of High Confidence Software Technologies, China
[1]{majing,kfwong}@se.cuhk.edu.hk, [2]wei.gao@vuw.ac.nz

Abstract

Automatic rumor detection is technically very challenging. In this work, we try to learn discriminative features from tweets content by following their non-sequential propagation structure and generate more powerful representations for identifying different type of rumors. We propose two *recursive* neural models based on a *bottom-up* and a *top-down* tree-structured neural networks for rumor representation learning and classification, which naturally conform to the propagation layout of tweets. Results on two public Twitter datasets demonstrate that our recursive neural models 1) achieve much better performance than state-of-the-art approaches; 2) demonstrate superior capacity on detecting rumors at very early stage.

判别特征

1 Introduction

Rumors have always been a social disease. In recent years, it has become unprecedentedly convenient for the "evil-doers" to create and disseminate rumors in massive scale with low cost thanks to the popularity of social media outlets on Twitter, Facebook, etc. The worst effect of false rumors could be devastating to individual and/or society.

Research pertaining rumors spans multiple disciplines, such as philosophy and humanities (DiFonzo and Bordia, 2007; Donovan, 2007), social psychology (Allport and Postman, 1965; Jaeger et al., 1980; Rosnow and Foster, 2005), political studies (Allport and Postman, 1946; Berinsky, 2017), management science (DiFonzo et al., 1994; Kimmel, 2004) and recently computer science and artificial intelligence (Qazvinian et al., 2011; Ratkiewicz et al., 2011; Castillo et al., 2011; Hannak et al., 2014; Zhao et al., 2015; Ma et al., 2015). Rumor is commonly defined as information that emerge and spread among people whose truth value is unverified or intentionally false (DiFonzo and Bordia, 2007; Qazvinian et al., 2011). Analysis shows that people tend to stop spreading a rumor if it is known as false (Zubiaga et al., 2016b). However, identifying such misinformation is non-trivial and needs investigative journalism to fact check the suspected claim, which is labor-intensive and time-consuming. The proliferation of social media makes it worse due to the ever-increasing information load and dynamics. Therefore, it is necessary to develop automatic and assistant approaches to facilitate real-time rumor tracking and debunking.

For automating rumor detection, most of the previous studies focused on text mining from sequential microblog streams using supervised models based on feature engineering (Castillo et al., 2011; Kwon et al., 2013; Liu et al., 2015; Ma et al., 2015), and more recently deep neural models (Ma et al., 2016; Chen et al., 2017; Ruchansky et al., 2017). These methods largely ignore or oversimplify the structural information associated with message propagation which however has been shown conducive to provide useful clues for identifying rumors. Kernel-based method (Wu et al., 2015; Ma et al., 2017) was thus proposed to model the structure as propagation trees in order to differentiate rumorous and non-rumorous claims by comparing their tree-based similarities. But such kind of approach cannot directly classify a tree without pairwise comparison with all other trees imposing unnecessary overhead, and it also cannot automatically learn any high-level feature representations out of the noisy surface features.

In this paper, we present a neural rumor detection approach based on recursive neural networks (RvNN) to bridge the content semantics and propagation clues. RvNN and its variants

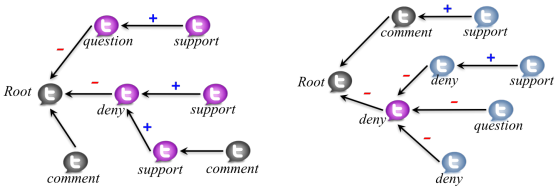
递归神经网络

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 1980–1989
Melbourne, Australia, July 15 - 20, 2018.

were originally used to compose phrase or sentence representation for syntactic and semantic parsing (Socher et al., 2011, 2012). Unlike parsing, the input into our model is a propagation tree rooted from a source post rather than the parse tree of an individual sentence, and each tree node is a responsive post instead of an individual words. The content semantics of posts and the responsive relationship among them can be jointly captured via the recursive feature learning process along the tree structure.

So, why can such neural model do better for the task? Analysis has generally found that Twitter could "self-correct" some inaccurate information as users share opinions, conjectures and evidences (Zubiaga et al., 2017). To illustrate our intuition, Figure 1 exemplifies the propagation trees of two rumors in our dataset, one being false and the other being true[1]. Structure-insensitive methods basically relying on the relative ratio of different stances in the text cannot do well when such clue is unclear like this example. However, it can be seen that when a post denies the false rumor, it tends to spark supportive or affirmative replies confirming the denial; in contrast, denial to a true rumor tends to trigger question or denial in its replies. This observation may suggest a more general hypothesis that the repliers tend to disagree with (or question) who support a false rumor or deny a true rumor, and also they tend to agree with who deny a false rumor or support a true rumor. Meanwhile, a reply, rather than directly responding to the source tweet (i.e., the root), is usually responsive to its immediate ancestor (Lukasik et al., 2016; Zubiaga et al., 2016a), suggesting obvious local characteristic of the interaction. The recursive network naturally models such structures for learning to capture the rumor indicative signals and enhance the representation by recursively aggregating the signals from different branches.

To this end, we extend the standard RvNN into two variants, i.e., a *bottom-up* (BU) model and a *top-down* (TD) model, which represent the propagation tree structure from different angles, in order to visit the nodes and combine their representations following distinct directions. The important merit of such architecture is that the node features can be selectively refined by the recursion given the connection and direction of all paths of the tree. As a result, it can be expected that the discriminative signals are better embedded into the learned representations.



(a) False rumor (b) True rumor

Figure 1: Propagation trees of two rumorous source tweets. Nodes may express stances on their parent as commenting, supporting, questioning or denying. The edge arrow indicates the direction from a response to its responded node, and the polarity is marked as '+' ('-') for support (denial). The same node color indicates the same stance on the veracity of root node (i.e., source tweet).

We evaluate our proposed approach based on two public Twitter datasets. The results show that our method outperforms strong rumor detection baselines with large margin and also demonstrate much higher effectiveness for detection at early stage of propagation, which is promising for real-time intervention and debunking. Our contributions are summarized as follows in three folds:

- This is the first study that deeply integrates both structure and content semantics based on tree-structured recursive neural networks for detecting rumors from microblog posts.
- We propose two variants of RvNN models based on bottom-up and top-down tree structures to generate better integrated representations for a claim by capturing both structural and textural properties signaling rumors.
- Our experiments based on real-world Twitter datasets achieve superior improvements over state-of-the-art baselines on both rumor classification and early detection tasks. We make the source codes in our experiments publicly accessible [2].

2 Related Work

Most previous automatic approaches for rumor detection (Castillo et al., 2011; Yang et al., 2012; Liu

[1]False (true) rumor means the veracity of the rumorous claim is false (true).

[2]https://github.com/majingCUHK/Rumor_RvNN

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 (Friggeri et al., 2014; Hannak et al., 2014) characterized by comments with links to debunking websites. Kwon et al. (2013) introduced a time-series-fitting model based on the volume of tweets over time. Ma et al. (2015) extended their model with more 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) used recurrent neural networks (RNN) to learn automatically the representations from tweets content based on time series. Recently, they studied to mutually reinforce stance detection and rumor classification in a neural multi-task learning framework (Ma et al., 2018). However, the approaches cannot embed features reflecting how the posts are propagated and requires careful data segmentation to prepare for 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. Compared to their studies, our model can learn the useful features via a more natural and general approach, i.e., the tree-structured neural network, to jointly generate representations from both structure and content.

RvNN has demonstrated state-of-the-art performances in a variety of tasks, e.g., images segmentation (Socher et al., 2011), phrase representation from word vectors (Socher et al., 2012), and sentiment classification in sentences (Socher et al., 2013). More recently, a deep RvNN was proposed to model the compositionality in natural language for fine-grained sentiment classification by stacking multiple recursive layers (Irsoy and Cardie, 2014). In order to avoid gradient vanishing, some studies integrated Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) to RvNN (Zhu et al., 2015; Tai et al., 2015). Mou et al. (2015) used a convolutional network over tree structures for syntactic tree parsing of natural language sentences.

3 Problem Statement

We define a Twitter rumor detection dataset as a set of claims $\mathcal{C} = \{C_1, C_2, \cdots, C_{|\mathcal{C}|}\}$, where each claim C_i corresponds to a source tweet r_i which consists of ideally all its relevant responsive tweets in chronological order, i.e., $C_i = \{r_i, x_{i1}, x_{i2}, \cdots, x_{im}\}$ where each x_{i*} is a responsive tweet of the root r_i. Note that although the tweets are notated sequentially, there are connections among them based on their reply or repost relationships, which can form a propagation tree structure (Wu et al., 2015; Ma et al., 2017) with r_i being the root node.

We formulate this task as a supervised classification problem, which learns a classifier f from labeled claims, that is $f : C_i \rightarrow Y_i$, where Y_i takes one of the four finer-grained classes: *non-rumor*, *false rumor*, *true rumor*, and *unverified rumor* that are introduced in the literature (Ma et al., 2017; Zubiaga et al., 2016b).

An important issue of the tree structure is concerned about the direction of edges, which can result in two different architectures of the model: 1) a bottom-up tree; 2) a top-down tree, which are defined as follows:

- **Bottom-up tree** takes the similar shape as shown in Figure 1, where responsive nodes always point to their responded nodes and leaf nodes not having any response are laid out at the furthest level. We represent a tree as $\mathcal{T}_i = \langle V_i, E_i \rangle$, where $V_i = C_i$ which consists of all relevant posts as nodes, and E_i denotes a set of all directed links, where for any $u, v \in V_i$, $u \leftarrow v$ exists if v *responses to* u. This structure is similar to a citation network where a response mimics a reference.

- **Top-down tree** naturally conforms to the direction of information propagation, in which a link $u \rightarrow v$ means the information flows from u to v and v sees it and provides a response to u. This structure reverses bottom-up tree and simulates how information cas-

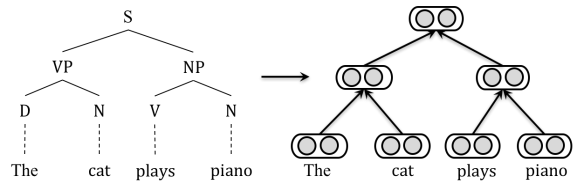


Figure 2: A binarized sentence parse tree (left) and its corresponding RvNN architecture (right).

cades from a source tweet, i.e., the root, to all its receivers, i.e., the decedents, which is similar as (Wu et al., 2015; Ma et al., 2017).

4 RvNN-based Rumor Detection

The core idea of our method is to strengthen the high-level representation of tree nodes by the recursion following the propagation structure over different branches in the tree. For instance, the responsive nodes confirming or supporting a node (e.g., "I agree", "be right", etc) can further reinforce the stance of that node while denial or questioning responses (e.g., "disagree, "really?!) otherwise weaken its stance. Compared to the kernel-based method using propagation tree (Wu et al., 2015; Ma et al., 2017), our method does not need pairwise comparison among large number of subtrees, and can learn much stronger representation of content following the response structure.

In this section, we will describe our extension to the standard RvNN for modeling rumor detection based on the bottom-up and top-down architectures presented in Section 3.

4.1 Standard Recursive Neural Networks

RvNN is a type of tree-structured neural networks. The original version of RvNN utilized binarized sentence parse trees (Socher et al., 2012), in which the representation associated with each node of a parse tree is computed from its direct children. The overall structure of the standard RvNN is illustrated as the right side of Figure 2, corresponding to the input parse tree at the left side.

二值化语义解析树

Leaf nodes are the words in an input sentence, each represented by a low-dimensional word embedding. Non-leaf nodes are sentence constituents, computed by recursion based on the presentations of child nodes. Let p be the feature vector of a parent node whose children are c_1 and c_2, the representation of the parent is computed by $p = f(W \cdot [c_1; c_2] + b)$, where $f(\cdot)$ is the activation function with W and b as parameters. This computation is done recursively over all tree nodes; the learned hidden vectors of the nodes can then be used for various classification tasks.

4.2 Bottom-up RvNN

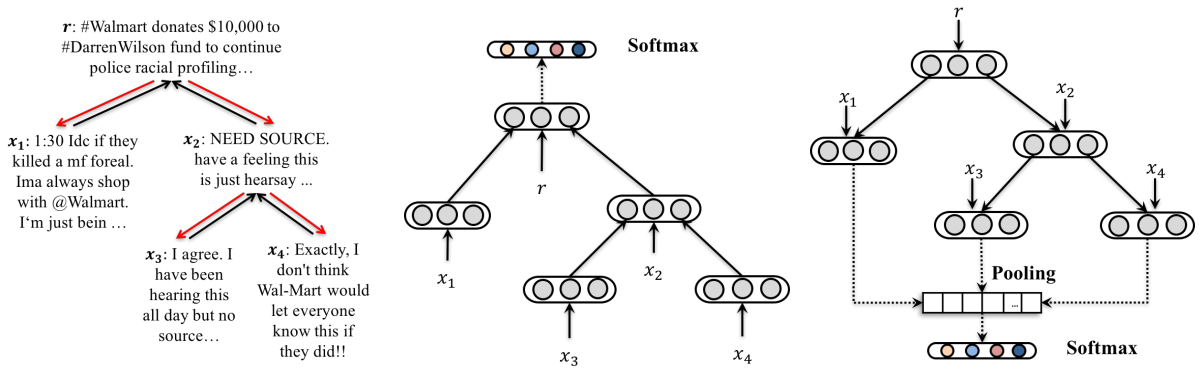
The core idea of bottom-up model is to generate a feature vector for each subtree by recursively visiting every node from the leaves at the bottom to the root at the top. In this way, the subtrees with similar contexts, such as those subtrees having a denial parent and a set of supportive children, will be projected into the proximity in the representation space. And thus such local rumor indicative features are aggregated along different branches into some global representation of the whole tree.

For this purpose, we make a natural extension to the original RvNN. The overall structure of our proposed bottom-up model is illustrated in Figure 3(b), taking a bottom-up tree (see Figure 3(a)) as input. Different from the standard RvNN, the input of each node in the bottom-up model is a post represented as a vector of words in the vocabulary in terms of $tfidf$ values. Here, every node has an input vector, and the number of children of nodes varies significantly[3].

In rumor detection, long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent units (GRU) (Cho et al., 2014) were used to learn textual representation, which adopts memory units to store information over long time steps (Ma et al., 2016). In this paper, we choose to extend GRU as hidden unit to model long-distance interactions over the tree nodes because it is more efficient due to fewer parameters. Let $\mathcal{S}(j)$ denote the set of direct children of the node j. The transition equations of node j in the bottom-up model are formulated as follows:

$$
\begin{aligned}
\tilde{x}_j &= x_j E \\
h_{\mathcal{S}} &= \sum_{s \in \mathcal{S}(j)} h_s \\
r_j &= \sigma\left(W_r \tilde{x}_j + U_r h_{\mathcal{S}}\right) \\
z_j &= \sigma\left(W_z \tilde{x}_j + U_z h_{\mathcal{S}}\right) \\
\tilde{h}_j &= tanh\left(W_h \tilde{x}_j + U_h (h_{\mathcal{S}} \odot r_j)\right) \\
h_j &= (1 - z_j) \odot h_{\mathcal{S}} + z_j \odot \tilde{h}_j
\end{aligned}
\tag{1}
$$

[3]In standard RvNN, since an input instance is the parse tree of a sentence, only leaf nodes have input vector, each node representing a word of the input sentence, and the non-leaf nodes are constituents of the sentence, and thus the number of children of a node is limited.



(a) Bottom-up/Top-down tree (b) Bottom-up RvNN model (c) Top-down RvNN model

Figure 3: A bottom-up/top-down propagation tree and the corresponding RvNN-based models. The black-color and red-color edges differentiate the bottom-up and top-down tree in Figure 3(a).

where x_j is the original input vector of node j, E denotes the parameter matrix for transforming this input post, $\tilde{x}_j$ is the transformed representation of j, $[W_*, U_*]$ are the weight connections inside GRU, and h_j and h_s refer to the hidden state of j and its s-th child. Thus h_S denotes the sum of the hidden state of all the children of j assuming that all children are equally important to j. As with the standard GRU, $\odot$ denotes element-wise multiplication; a reset gate r_j determines how to combine the current input $\tilde{x}_j$ with the memory of children, and an update gate z_j defines how much memory from the children is cascaded into the current node; and $\tilde{h}_j$ denotes the candidate activation of the hidden state of the current node. Different from the standard GRU unit, the gating vectors in our variant of GRU are dependent on the states of many child units, allowing our model to incorporate representations from different children.

After recursive aggregation from bottom to up, the state of root node (i.e., source tweet) can be regard as the representation of the whole tree which is used for supervised classification. So, an output layer is connected to the root node for predicting the class of the tree using a softmax function:

$$\hat{y} = Softmax(Vh_0 + b) \tag{2}$$

where h_0 is the learned hidden vector of root node; V and b are the weights and bias in output layer.

4.3 Top-down RvNN

This model is designed to leverage the structure of top-down tree to capture complex propagation patterns for classifying rumorous claims, which is shown in Figure 3(c). It models how the information flows from source post to the current node. The idea of this top-down approach is to generate a strengthened feature vector for each post considering its propagation path, where rumor-indicative features are aggregated along the propagation history in the path. For example, if current post agree with its parent's stance which denies the source post, the denial stance from the root node down to the current node on this path should be reinforced. Due to different branches of any non-leaf node, the top-down visit to its subtree nodes is also recursive. However, the nature of top-down tree lends this model different from the bottom-up one. The representation of each node is computed by combining its own input and its parent node instead of its children nodes. This process proceeds recursively from the root node to its children until all leaf nodes are reached.

Suppose that the hidden state of a non-leaf node can be passed synchronously to all its child nodes without loss. Then the hidden state h_j of a node j can be computed by combining the hidden state $h_{\mathcal{P}(j)}$ of its parent node $\mathcal{P}(j)$ and its own input vector x_j. Therefore, the transition equations of node j can be formulated as a standard GRU:

$$\begin{aligned}
\tilde{x}_j &= x_j E \\
r_j &= \sigma\left(W_r \tilde{x}_j + U_r h_{\mathcal{P}(j)}\right) \\
z_j &= \sigma\left(W_z \tilde{x}_j + U_z h_{\mathcal{P}(j)}\right) \\
\tilde{h}_j &= tanh\left(W_h \tilde{x}_j + U_h (h_{\mathcal{P}(j)} \odot r_j)\right) \\
h_j &= (1 - z_j) \odot h_{\mathcal{P}(j)} + z_j \odot \tilde{h}_j
\end{aligned} \tag{3}$$

Through the top-down recursion, the learned representations are eventually embedded into the hidden vector of all the leaf nodes. Since the num-

ber of leaf nodes varies, the resulting vectors cannot be directly fed into a fixed-size neural layer for output. Therefore, we add a max-pooling layer to take the maximum value of each dimension of the vectors over all the leaf nodes. This can also help capture the most appealing indicative features from all the propagation paths.

Based on the pooling result, we finally use a softmax function in the output layer to predict the label of the tree:

$$\hat{y} = Softmax(Vh_{\infty} + b) \quad (4)$$

where h_{∞} is the pooling vector over all leaf nodes, V and b are parameters in the output layer.

Although both of the two RvNN models aim to capture the structural properties by recursively visiting all nodes, we can conjecture that the top-down model would be better. The hypothesis is that in the bottom-up case the final output relies on the representation of single root, and its information loss can be larger than the top-down one since in the top-down case the representations embedded into all leaf nodes along different propagation paths can be incorporated via pooling holistically.

4.4 Model Training

The model is trained to minimize the squared error between the probability distributions of the predictions and the ground truth:

$$L(y, \hat{y}) = \sum_{n=1}^{N} \sum_{c=1}^{C} (y_c - \hat{y}_c)^2 + \lambda ||\theta||_2^2 \quad (5)$$

where y_c is the ground truth and $\hat{y}_c$ is the prediction probability of a class, N is the number of training claims, C is the number of classes, $||.||_2$ is the L_2 regularization term over all model parameters θ, and λ is the trade-off coefficient.

During training, all the model parameters are updated using efficient back-propagation through structure (Goller and Kuchler, 1996; Socher et al., 2013), and the optimization is gradient-based following the Ada-grad update rule (Duchi et al., 2011) to speed up the convergence. We empirically initialize the model parameters with uniform distribution and set the vocabulary size as 5,000, the size of embedding and hidden units as 100. We iterate over all the training examples in each epoch and continue until the loss value converges or the maximum epoch number is met.

5 Experiments and Results

5.1 Datasets

For experimental evaluation, we use two publicly available Twitter datasets released by Ma et al. (2017), namely Twitter15 and Twitter16[4], which respectively contains 1,381 and 1,181 propagation trees (see (Ma et al., 2017) for detailed statistics). In each dataset, a group of wide spread source tweets along with their propagation threads, i.e., replies and retweets, are provided in the form of tree structure. Each tree is annotated with one of the four class labels, i.e., non-rumor, false rumor, true rumor and unverified rumor. We remove the retweets from the trees since they do not provide any extra information or evidence content-wise. We build two versions for each tree, one for the bottom-up tree and the other for the top-down tree, by flipping the edges' direction.

5.2 Experimental Setup

We make comprehensive comparisons between our models and some state-of-the-art baselines on rumor classification and early detection tasks.

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

- DTC: The information credibility model using a Decision-Tree Classifier (Castillo et al., 2011) based on manually engineering various statistical features of the tweets.

- RFC: The Random Forest Classier using 3 fitting parameters as temporal properties and a set of handcrafted features on user, linguistic and structural properties (Kwon et al., 2013).

- SVM-TS: A linear SVM classifier that uses time-series to model the variation of handcrafted social context features (Ma et al., 2015).

- SVM-BOW: A naive baseline we built by representing text content using bag-of-words and using linear SVM for rumor classification.

- SVM-TK and **SVM-HK**: SVM classifier uses a Tree Kernel (Ma et al., 2017) and that uses a Hybrid Kernel (Wu et al., 2015), respectively, both of which model propagation structures with kernels.

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

[4] https://www.dropbox.com/s/7ewzdrbelpmrnxu/rumdetect2017.zip?dl=0

(a) Twitter15 dataset

Method	Acc.	NR F_1	FR F_1	TR F_1	UR F_1
DTR	0.409	0.501	0.311	0.364	0.473
DTC	0.454	0.733	0.355	0.317	0.415
RFC	0.565	**0.810**	0.422	0.401	0.543
SVM-TS	0.544	0.796	0.472	0.404	0.483
SVM-BOW	0.548	0.564	0.524	0.582	0.512
SVM-HK	0.493	0.650	0.439	0.342	0.336
SVM-TK	0.667	0.619	0.669	0.772	0.645
GRU-RNN	0.641	0.684	0.634	0.688	0.571
BU-RvNN	0.708	0.695	0.728	0.759	0.653
TD-RvNN	**0.723**	0.682	**0.758**	**0.821**	**0.654**

(b) Twitter16 dataset

Method	Acc.	NR F_1	FR F_1	TR F_1	UR F_1
DTR	0.414	0.394	0.273	0.630	0.344
DTC	0.465	0.643	0.393	0.419	0.403
RFC	0.585	0.752	0.415	0.547	0.563
SVM-TS	0.574	**0.755**	0.420	0.571	0.526
SVM-BOW	0.585	0.553	0.556	0.655	0.578
SVM-HK	0.511	0.648	0.434	0.473	0.451
SVM-TK	0.662	0.643	0.623	0.783	0.655
GRU-RNN	0.633	0.617	0.715	0.577	0.527
BU-RvNN	0.718	0.723	0.712	0.779	0.659
TD-RvNN	**0.737**	0.662	**0.743**	**0.835**	**0.708**

Table 1: Results of rumor detection. (NR: non-rumor; FR: false rumor; TR: true rumor; UR: unverified rumor)

- BU-RvNN and **TD-RvNN**: Our bottom-up and top-down RvNN models, respectively.

We implement **DTC** and **RFC** using Weka[5], SVM-based models using LibSVM[6] and all neural-network-based models with Theano[7]. We conduct 5-fold cross-validation on the datasets and use accuracy over all the four categories and F1 measure on each class to evaluate the performance of models.

5.3 Rumor Classification Performance

As shown in Table 1, our proposed models basically yield much better performance than other methods on both datasets via the modeling of interaction structures of posts in the propagation.

It is observed that the performance of the 4 baselines in the first group based on handcrafted features is obviously poor, varying between 0.409 and 0.585 in accuracy, indicating that they fail to generalize due to the lack of capacity capturing helpful features. Among these baselines, SVM-TS and RFC perform relatively better because they use additional temporal traits, but they are still clearly worse than the models not relying on feature engineering. DTR uses a set of regular expressions indicative of stances. However, only 19.6% and 22.2% tweets in the two datasets contain strings covered by these regular expressions, rendering unsatisfactory result.

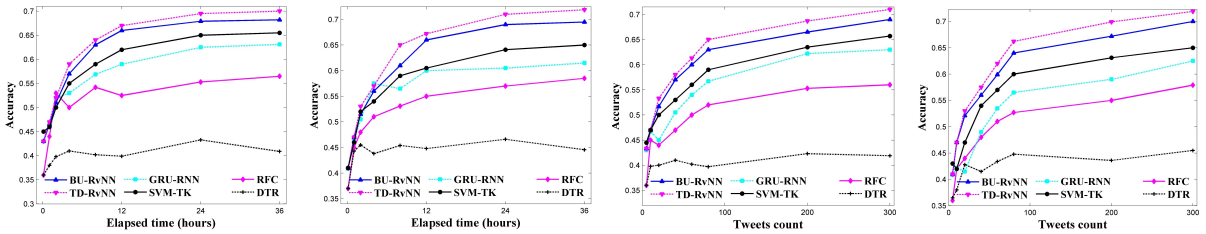
Among the two kernel methods that are based on comparing propagation structures, we observe that SVM-TK is much more effective than SVM-HK. There are two reasons: 1) SVM-HK was originally proposed and experimented on Sina Weibo (Wu et al., 2015), which may not be generalize well on Twitter. 2) SVM-HK loosely couples two separate kernels: a RBF kernel based on handcrafted features, plus a random walk-based kernel which relies on a set of pre-defined keywords for jumping over the nodes probabilistically. This under utilizes the propagation information due to such oversimplified treatment of tree structure. In contrast, SVM-TK is an integrated kernel and can fully utilize the structure by comparing the trees based on both textual and structural similarities.

It appears that using bag-of-words is already a decent model evidenced as the fairly good performance of SVM-BOW which is even better than SVM-HK. This is because the features of SVM-HK are handcrafted for binary classification (i.e., non-rumor vs rumor), ignoring the importance of indicative words or units that benefit finer-grained classification which can be captured more effectively by SVM-BOW.

The sequential neural model GRU-RNN performs slightly worse than SVM-TK, but much worse than our recursive models. This is because it is a special case of the recursive model where each non-leaf node has only one child. It has to rely on a linear chain as input, which missed out valuable structural information. However, it does learn high-level features from the post content via hidden units of the neural model while SVM-TK cannot which can only evaluates similarities based on the overlapping words among subtrees. Our recursive models are inherently tree-structured and take advantages of representation learning following the propagation structure, thus beats SVM-TK.

In the two recursive models, TD-RvNN outperforms BU-RvNN, which indicates that the bottom-up model may suffer from larger information loss than the top-down one. This verifies the hypothesis we made in Section 4.3 that the pooling layer

[5] www.cs.waikato.ac.nz/ml/weka
[6] www.csie.ntu.edu.tw/~cjlin/libsvm
[7] deeplearning.net/software/theano



(a) Twitter15 (elapsed time) (b) Twitter16 (elapsed time) (c) Twitter15 (tweets count) (d) Twitter16 (tweets count)

Figure 4: Early rumor detection accuracy at different checkpoints in terms of elapsed time (tweets count).

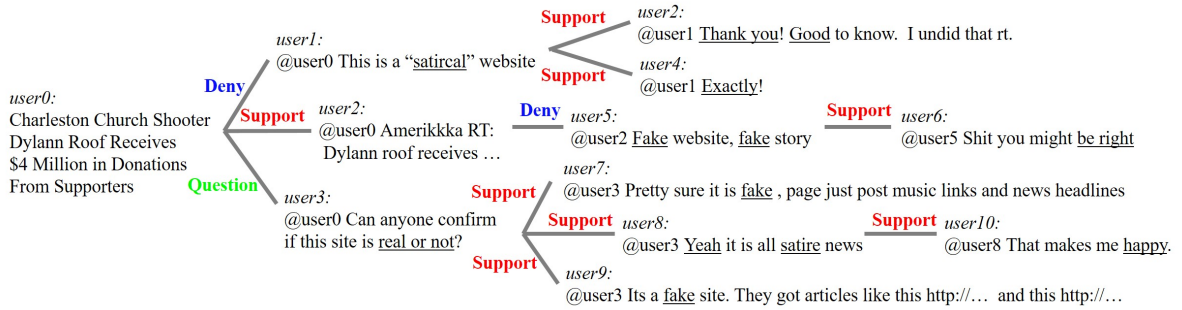


Figure 5: A correctly detected false rumor at early stage by both of our models, where propagation paths are marked with relevant stances. Note that edge direction is not shown as it applies to either case.

in the top-down model can effectively select important features embedded into the leaf nodes.

For only the non-rumor class, it seems that our method does not perform so well as some feature-engineering baselines. This can be explained by the fact that these baselines are trained with additional features such as user information (e.g., profile, verification status, etc) which may contain clues for differentiating non-rumors from rumors. Also, the responses to non-rumors are usually much more diverse with little informative indication, making identification of non-rumors more difficult based on content even with the structure.

5.4 Early Rumor Detection Performance

Detecting rumors at early state of propagation is important so that interventions can be made in a timely manner. We compared different methods in term of different time delays measured by either tweet count received or time elapsed since the source tweet is posted. The performance is evaluated by the accuracy obtained when we incrementally add test data up to the check point given the targeted time delay or tweets volume.

Figure 4 shows that the performance of our recursive models climbs more rapidly and starts to supersede the other models at the early stage. Although all the methods are getting to their best performance in the end, TD-RvNN and BU-RvNN only need around 8 hours or about 90 tweets to achieve the comparable performance of the best baseline model, i.e., SVM-TK, which needs about 36 hours or around 300 posts, indicating superior early detection performance of our method.

Figure 5 shows a sample tree at the early stage of propagation that has been correctly classified as a false rumor by both recursive models. We can see that this false rumor demonstrates typical patterns in subtrees and propagation paths indicative of the falsehood, where a set of responses supporting the parent posts that deny or question the source post are captured by our bottom-up model. Similarly, some patterns of propagation from the root to leaf nodes like "support→deny→support" are also seized by our top-down model. In comparison, sequential models may be confused because the supportive key terms such as "be right", "yeah", "exactly!" dominate the responses, and the SVM-TK may miss similar subtrees by just comparing the surface words.

6 Conclusions and Future Work

We propose a bottom-up and a top-down tree-structured model based on recursive neural networks for rumor detection on Twitter. The inher-

ent nature of recursive models allows them using propagation tree to guide the learning of representations from tweets content, such as embedding various indicative signals hidden in the structure, for better identifying rumors. Results on two public Twitter datasets show that our method improves rumor detection performance in very large margins as compared to state-of-the-art baselines.

In our future work, we plan to integrate other types of information such as user properties into the structured neural models to further enhance representation learning and detect rumor spreaders at the same time. We also plan to use unsupervised models for the task by exploiting structural information.

Acknowledgment

This work is partly supported by Innovation and Technology Fund (ITF) Project No. 6904333, and General Research Fund (GRF) Project No. 14232816 (12183516). We would like to thank anonymous reviewers for the insightful comments.

References

Gordon W Allport and Leo Postman. 1946. An analysis of rumor. *Public Opinion Quarterly* 10(4):501–517.

G.W. Allport and L.J. Postman. 1965. *The psychology of rumor*. Russell & Russell.

Adam J. Berinsky. 2017. Rumors and health care reform: Experiments in political misinformation. *British Journal of Political Science* 47(2):241262.

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

Tong Chen, Lin Wu, Xue Li, Jun Zhang, Hongzhi Yin, and Yang Wang. 2017. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. *arXiv preprint arXiv:1704.05973* .

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* .

Nicholas DiFonzo and Prashant Bordia. 2007. Rumor, gossip and urban legends. *Diogenes* 54(1):19–35.

Nicholas DiFonzo, Prashant Bordia, and Ralph L Rosnow. 1994. Reining in rumors. *Organizational Dynamics* 23(1):47–62.

Pamela Donovan. 2007. How idle is idle talk? one hundred years of rumor research. *Diogenes* 54(1):59–82.

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*.

Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, 1996., IEEE International Conference on*. IEEE, volume 1, pages 347–352.

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*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Ozan Irsoy and Claire Cardie. 2014. Deep recursive neural networks for compositionality in language. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'14, pages 2096–2104.

Marianne E Jaeger, Susan Anthony, and Ralph L Rosnow. 1980. Who hears what from whom and with what effect: A study of rumor. *Personality and Social Psychology Bulletin* 6(3):473–478.

Allan J Kimmel. 2004. *Rumors and rumor control: A manager's guide to understanding and combatting rumors*. Routledge.

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.

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.

Michal Lukasik, PK Srijith, Duy Vu, Kalina Bontcheva, Arkaitz Zubiaga, and Trevor Cohn. 2016. Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. volume 2, pages 393–398.

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, 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, 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. Detect rumor and stance jointly by neural multi-task learning. In *Companion Proceedings of the The Web Conference 2018*. WWW '18, pages 585–593.

Lili Mou, Hao Peng, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2015. Discriminative neural sentence modeling by tree-based convolution. *arXiv preprint arXiv:1504.01106* .

Vahed Qazvinian, Emily Rosengren, Dragomir R Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11, pages 1589–1599.

Jacob Ratkiewicz, Michael Conover, Mark Meiss, Bruno Gonçalves, Snehal Patil, Alessandro Flammini, and Filippo Menczer. 2011. Truthy: mapping the spread of astroturf in microblog streams. In *Proceedings of the 20th International Conference Companion on World Wide Web*. WWW '11, pages 249–252.

Ralph L Rosnow and Eric K Foster. 2005. Rumor and gossip research. *Psychological Science Agenda* 19(4).

Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. CIKM '17, pages 797–806.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP-CoNLL '12, pages 1201–1211.

Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. pages 129–136.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. pages 1631–1642.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075* .

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*. IEEE, pages 651–662.

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.

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.

Xiaodan Zhu, Parinaz Sobihani, and Hongyu Guo. 2015. Long short-term memory over recursive structures. In *Proceedings of the 32nd International Conference on Machine Learning*. pages 1604–1612.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2017. Detection and resolution of rumours in social media: A survey. *arXiv preprint arXiv:1704.00656* .

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.