Neurocomputing 458 (2021) 468–477

Contents lists available at ScienceDirect

Neurocomputing

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

A lightweight propagation path aggregating network with neural topic model for rumor detection

Pengfei Zhang [a], Hongyan Ran [a], Caiyan Jia [a,*], Xuanya Li [b,*], Xueming Han [a]

[a] School of Computer and Information Technology and Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing 100044, China
[b] Baidu Inc., , Beijing 100085, China

ARTICLE INFO

Article history:
Received 18 December 2020
Revised 4 April 2021
Accepted 20 June 2021
Available online 22 June 2021
Communicated by Zidong Wang

Keywords:
Rumor detection
Wasserstein autoencoder
Neural topic model
Propagation structure
Lightweight

ABSTRACT

The structure information associated with message propagation has been proved to be effective to distinguish false and true rumors. However, existing methods lack an efficient way to learn the representation of the whole rumors which captures the intrinsic mechanism of rumor propagation structures and semantics. In this study, we propose a lightweight propagation path aggregating (PPA) neural network for rumor embedding and classification. In the network, we first model the propagation structure of each rumor as an independent set of propagation paths in which each path represents the source post in a different talking context. We then aggregate all paths to obtain the representation of the whole propagation structure. Besides, we utilize a neural topic model in the Wasserstein autoencoder (WAE) framework to capture event insensitive stance patterns in response propagation trees where no source post is included. Empirical studies demonstrate that 1) PPA achieves the state-of-the-art performance with much less parameters and training time, 2) PPA can further benefit from the pre-trained neural topic model which enables to fully use unlabeled data, thus improves the performance of PPA especially when labeled samples are limited or rumors are spreading at early stage. Meanwhile, this topic model offers an explicit interpretation of stance patterns in the form of topics, consequently improves interpretability of the PPA network. The source code can be available at https://github.com/zperfet/PathFakeGit.

1. Introduction

With the rapid development of open social media platforms, such as Twitter and Facebook, rumors can quickly reach countless audience and lead to unexpected and disastrous consequences with low cost. However, it is labor-intensive and time-consuming for ordinary people to distinguish false news from massive amounts of online news. Therefore, it is highly necessary to develop automatic approaches to debunking rumors effectively and efficiently.

Recently, a tree-structured neural network RvNN [1] is proposed to bridge the content semantics and propagation clues. To the best of our knowledge, it is the first to learn the representation of propagation structures coupled with textural information, achieves significant performance. However, this tree-structured neural network recursively learns the representation of rumors with gated recurrent unit [2]. This makes the model have high computational burden and converge slowly. Therefore, in this study, we aim to develop a more efficient and effective lightweight-model to learn the representation of propagation structures of rumors, meanwhile make the model be able to capture the implicit semantic features hidden in the propagation process.

As is well-known, audiences' views towards a news vary from person to person. Some people may focus on details of news, while some tend to express their personal stances. To illustrate our intuition, Fig. 1 exemplifies different attitudes towards a source tweet, which describes the president Obama's opinion about stay-at-home-moms. Some people believe it is fake, while some start to attack Obama for his personal behavior or on unemployment rate. From the perspective of a source post, every propagation path corresponds to the source post in a different context. Each propagation path from a root node to a leaf node is intuitively a natural clustering of posts from different aspects. Therefore, it is natural to represent or embed propagation trees of rumors based on their independent propagation paths.

In addition, it is claimed that Twitter could "self-correct" some inaccurate information as users share opinions, conjectures and evidences [3]. People's attitudes towards a source post will reflect the truthness of the post. At this point of view, DTR [4] utilizes

* Corresponding authors.
E-mail addresses: cyjia@bjtu.edu.cn (C. Jia), lixuanya@baidu.com (X. Li).

https://doi.org/10.1016/j.neucom.2021.06.062
0925-2312/

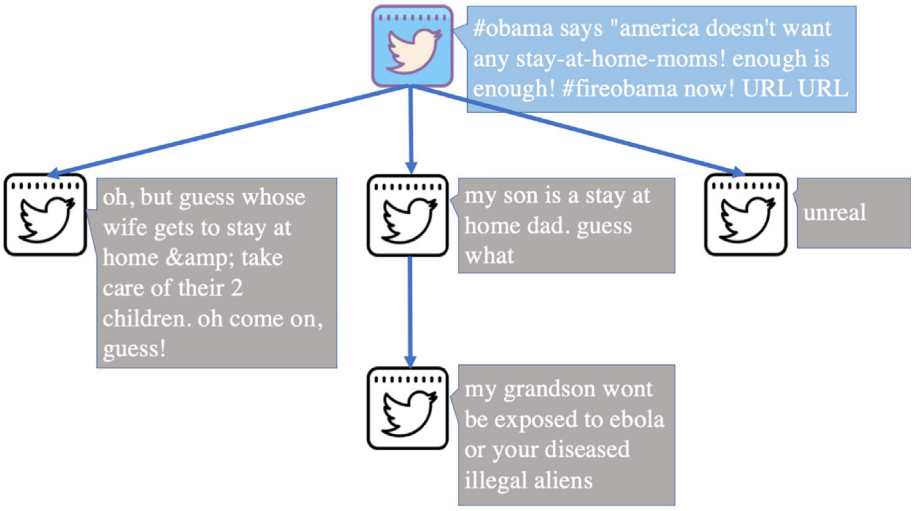

Fig. 1. An example of a propagation tree.

hand-crafted regular expressions (such as "really?", "not true") to find questing and denying tweets, but the method is oversimplified and suffers from very low recall. The studies [5,6] mutually reinforce stance detection and rumor classification in a neural multi-task learning framework. The framework focuses on learning the embedding vectors of posts and the whole rumors to boost the performance of the both tasks. However, it is important to reveal people's diverse stance distribution for the whole rumors, meanwhile understand people's concerns towards rumors.

In this paper, we investigate (1) how to generate powerful representations for propagation trees considering both their semantics and structures, and (2) how to find event-insensitive stance patterns automatically in propagation paths to characterize people's opinions. Therefore, we present a novel propagation path aggregating neural network (PPA) and utilize a neural topic model to characterize the topic distribution of stances. First, we represent propagation trees by aggregating the vector representations of their propagation paths, and parallel train the unsupervised topic model WAE to reveal stance topics of the corresponding response trees. We then integrate the vector representation of each tree and its pre-trained hidden topic vector to jointly predict each rumor.

The contributions of this paper can be summarized as follows.

- This is the first study that adopts a simple light-weighted feed-forward neural network to learn the embedding of rumors by organizing post content based on their propagation paths.
- This is the first work that introduces neural topic models for rumor detection. The hidden vectors learnt by neural topic model WAE on response propagation paths could explicitly interpret people's stance in terms of topics, further enhances the performance of PPA and improves interpretability of the proposed model.
- We conduct a series of experiments on four real-world datasets. The empirical studies demonstrate that our model achieves the state-of-the-art performance with much less training time and parameters, especially when labeled data is limited or rumors are spreading at early stage.

2. Related work

Most previous feature-based rumor detection methods intend to learn a supervised classifier based on a wide range of information sources, such as text content [7], user profiles [7,8] and propagation patterns [9,10]. Except for heavy preprocessing, hand-crafted features used in these methods are often inadequate or unreliable for short, informal and ungrammatical social media texts.

Deep neural networks are then applied to learn effective features automatically in recent years. Ma et al. [11] utilizes recurrent neural networks to learn post content representations by modeling all posts of a source rumor as time series. Liu and Wu [12] models propagation paths characterized by users' properties as multivariate time series, and learns to capture the variations of user profiles on rumors using recurrent and convolutional neural networks. Ruchansky et al. [13] fuses features from multiple information sources using deep neural networks to further improve detection performance. However, these methods fail to embed the complex and dynamic structure information of rumors. Yuan et al. [14] proposes a neural attention network GLAN that integrates and learns the semantic of posts and the global structural information on a heterogeneous graph composed of user-and-user, user-and-post and post-and-post relationships, has showed good performance. Nevertheless, its efficiency is limited for a large scale graph and at some scenario, no all these three kinds of relationships are available.

More recent rumor detection approaches use deep neural networks to learn the representation of propagation structures of rumors with content information, have been proved to be effective. By tracing back to earlier propagation tree related methods, they focus on comparing the information diffusion similarities of true and false news. For example, Wu et al. [15] applies a random walk graph kernel to model propagation trees and build a SVM classifier for rumor detection. Ma et al. [10] proposes a tree kernel to compute the similarity of propagation trees by counting similar sub-structures. While deep neural network based methods intend to learn embedding vectors for the whole propagation structures of rumors. Collecting all this kind of methods in the literature, Ma et al. [1] proposes a tree-structured recursive neural network RvNN to integrate both structure and content semantics information of propagation trees of rumors. Khoo et al. [16] proposes a dual-attention (structure aware hierarchical token attention and post-level attention) network using two heavy transformer structures [17] to differentiate between the community's responses of real and fake claims. Bian et al. [18] proposes a novel bi-directional graph model Bi-GCN to explore propagation and dispersion characteristics by operating on both top-down and bottom-up propagation trees. However, these models have no ability to capture the

event-insensitive stance words and some of them, especially RvNN and PLAN, are heavy and inefficient.

Probabilistic topic models, especially LDA [19], have been widely used to discover the underlying topics in unlabeled documents with strong interpretability. The distributions of topic words and document clusters have been characterized in the models. To avoid the expensive iterative inference based on variational Bayesian or collapsed Gibbs sampling, neural topic models, such as NVDM [20], ProdLDA [21] and NTM-R [22], which incorporate variational autoencoders (VAE [23]) into the inference procedure via a forward pass of a recognition network, has been developed and achieved great success.

However, VAE relies on a reparameterization trick that only works with the "location-scale" family of distributions, where Dirichlet distribution commonly used in classical probabilistic topic models does not belongs to. Therefore, Nan et al. [24] proposes a neural topic model W-LDA in the WAE (Wasserstein autoencoder) framework to make the latent document-topic vectors of the model follow Dirichlet prior instead of Gaussian approximation. This topic model is better than LDA and ProdLDA with Laplacian approximation at generating texts. In addition, VAE is successfully used in semi-supervised text classification [25] since the topic distribution of words can been learned on unlabeled documents. While for rumor detection, unlabeled data are easier to be obtained than labeled data and if we known the stance distribution of rumors and non-rumors, it may help to judge the veracity of a rumor at its early stage and improves model's accuracy by using both labeled and unlabeled data. Thus, neural topic models (e.g. W-LDA) is promising to be used in rumor detection.

3. Method

3.1. Problem statement

Suppose that a rumor dataset is a set of N claims $D = \{C_1, C_2, \ldots, C_N\}$, where each claim C_i corresponds to a source tweet r_i and all its M reply tweets x_{i*}, i.e. $C_i = \{r_i, x_{i1}, x_{i2}, \ldots, x_{iM}\}$ and $x_{ij} \neq x_{ik}$ for $j \neq k$. Therefore, each C_i corresponds to a propagation tree with r_i being its root node, and in which node j has a child node k if the tweet k directly responses to the tweet j.

The goal is to learn a supervised classifier from labeled news stories, that is $f : C_i \rightarrow Y_i$, where Y_i belongs to four classes: non-rumor, false rumor, true rumor and unverified rumor concerned in the literature [10].

3.2. The proposed model

The proposed model consists of three major components: propagation path aggregating (PPA) neural network, unsupervised pre-training using the neural topic model WAE, integrating PPA embedding vectors and the latent document-topic vectors of the neural topic model to jointly predict the classes of rumors. The whole architecture of the proposed model is showed in Fig. 2.

3.2.1. Propagation path aggregating neural network

We define propagation paths of a propagation tree as $P_i = \{P_{i1}, P_{i2}, \ldots, P_{ip}\}$, where p is the number of paths and P_{ij} corresponds to a path from source tweet r_i to the j-th leaf node, i.e. $P_{ij} = [r_i; x_{ij1}; x_{ij2}; \ldots; x_{ijn}]$, where x_{ijk} is the k-th tree node of the j-th path in the i-th propagation tree; n is the number of nodes along the path. Each path content is composed of words from tree nodes along the current propagation path starting from a root node.

The existing methods [1,16] obtain the representation of a single tree node first, then encode structure information of the corresponding propagation tree. These methods have two disadvantages. First, these methods separate the propagation structure information of a propagation tree from the text representation of tree nodes. Therefore, besides encoding the text of nodes along propagation paths, we intent to naturally embed the structure information into the text representations. Second, the algorithm complexity of tree representation based on tree nodes is $O(N_T)$ (N_T is the number of tree nodes of a propagation tree), which can be very large. While the algorithm complexity of propagation path based embedding is only $O(N_L)$ (N_L is the leaf node number of a propagation tree). Thus, if we use propagation path sampling to obtain the embedding vector of a propagation tree, we can greatly reduce the complexity.

We compared several propagation path embedding methods, including bag-of-words (BoW) encoding, CNN, LSTMs and BERT [26]. In experiments, we found that the simple embedding model (BoW model) can achieve comparable performance as the complex models (CNN, LSTMs, BERT) for datasets with propagation structure and post content. Therefore, unlike previous models using complex models, such as Khoo et al. [16], we use the simplest (BoW) model to encode each propagation path.

At first, we use (BoW) model to get the vector of each word in the propagation path. The representation vector of each word is initialized randomly and is trainable during model training process. In order to fully consider the information of each word, we then take the sum operator to obtain the representation of a path in the following.

$$p_{ij} = \sum_{v=1}^{|r_i|} E(r_i)_v + \sum_{k=1}^{n} \sum_{v=1}^{|x_{ijk}|} E(x_{ijk})_v. \tag{1}$$

where $E(x)_v$ means the word embedding of the v-th word in x (a source post or a reply tweet) initialized by a random vector or a pre-trained word embedding vector.

Although the number of leaf nodes is much smaller than that of tree nodes, it can still be very large, so we choose propagation path with the highest quality to further reduce the complexity of the model. Specifically, we first sort the propagation paths by the number of tree nodes in the paths and the reply time of leaf nodes. Then, we keep the top 200 paths in the top order. In order to prevent over fitting, we use dropout on the representation of the top 200 propagation paths. In our experiment, the performance of the model trained based on the top 200 paths with proper dropout is consistent with that of all paths. Then we use max pooling to capture the most discriminative features of all propagation paths, then obtain the embedding of tree P_i:

$$\bar{p}_i = \text{MaxPooling}(\text{dropout}((p_{ij}))). \tag{2}$$

We name the above rumor detection network as Propagation Path Aggregating Neural Network (PPA). We also test min pooling and average pooling. The performance of using the min pooling is close to the max pooling, but the average pooling is much worse than the max pooling.

We then predict the label of tree P_i using softmax in the output layer:

$$\bar{y}_i = \text{Softmax}(W\bar{p}_i + b). \tag{3}$$

In the better version of RvNN model with top-down learning process [1], a propagation tree is represented as the max-pooling of hidden states of leaf nodes which aggregates the information of each propagation path. However, the vectors of leaf nodes, which are likely far from a source post, mainly focus on response tweets thanks to recursive neural networks. Therefore, leaf nodes cannot learn an effective representation for the corresponding source tweet. If we view every propagation path as a talking context of source post, then our model is trying to learn a robust representation of a source post in different contexts.

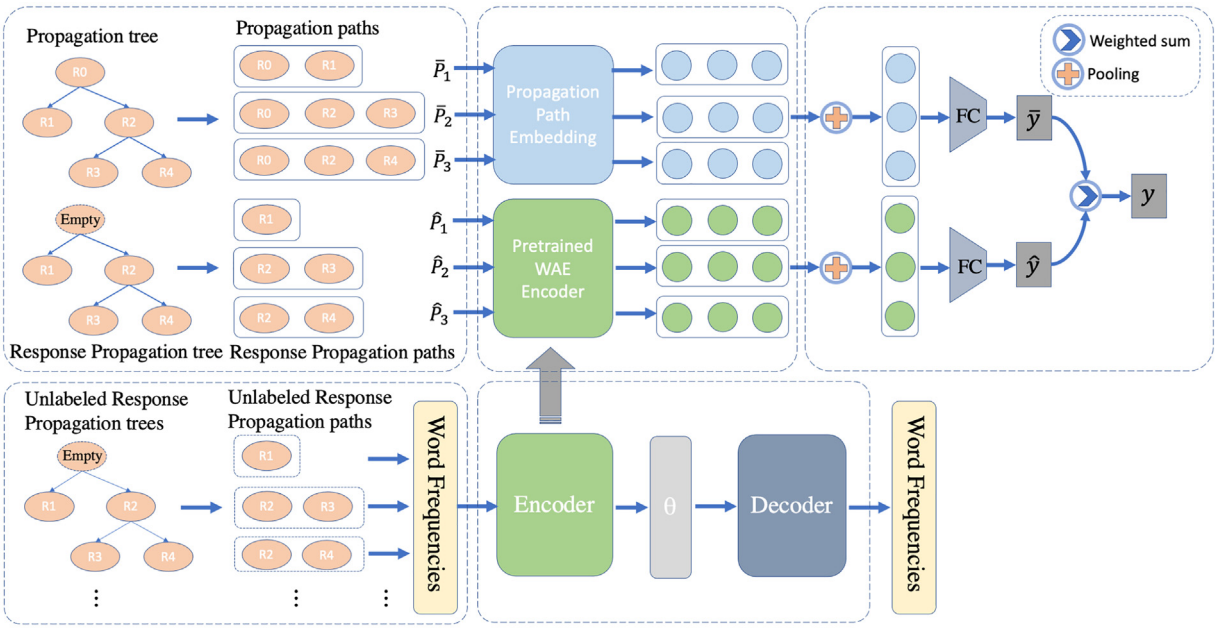

Fig. 2. The Architecture of PPA Model with WAE.

3.2.2. Pre-trained Wasserstein autoencoder

As claimed before, a propagation path may be a discussion of some specific aspects of its source tweet, and reply tweets usually contain event-insensitive stance patterns. Hence we utilize neural topic models to explore hidden stance topics in propagation paths. Different from the previous step, we remove the source tweet in each path to eliminate the impact of source tweet, consequently focus on event-insensitive stance words. We call a tree (described in the above section) without source tweet "response propagation tree" with an empty node as its root, and name its paths "response propagation paths". In this study, we modify the state-of-the-art neural topic model WAE [24], which is based on the WAE (Wasserstein autoencoder) framework, to learn the stance topics on the response propagation trees of rumors.

To generate a target document w, WAE first samples a topic vector θ from a prior distribution P_Θ and then pass through a decoder network. The resulting distribution in the target domain is P_{dec} with density:

$$p_{dec}(w) = \int_\theta p_{dec}(w|\theta)p(\theta)d\theta. \tag{4}$$

According to [27], minimizing the optimal transport distance between P_{dec} and target distribution P_w is equivalent to minimizes a reconstruction term and a distribution term as follows.

$$\inf_{Q(\theta|w)} E_{P_w} E_{Q(\theta|w)}[c(w, dec(\theta))] + \lambda \cdot \mathscr{D}_\Theta(Q_\Theta, P_\Theta), \tag{5}$$

where $Q_\Theta := E_{P_w} Q(\theta|w)$ is the aggregated posterior and λ is a scalar factor, $\mathscr{D}(Q_\Theta, P_\Theta)$ is an arbitrary divergence between Q_Θ and P_Θ.

Specifically, in our scenario, suppose that there are V words in the vocabulary, each response propagation path is represented as a BoW $w = (w_1, w_2, \ldots, w_V)$, where w_i is the number of occurrences of the i-th vocabulary word. The encoder of WAE maps w to a hidden state z with dimension K (K is often viewed as topic number) using Multi-Layer Perceptron (MLP) (Eq. 6). Document-topic vectors θ, representing topic distribution of propagation paths, can then be obtained by applying softmax over z (Eq. 7). The decoder maps θ to an output layer of V units using a single layer neural network. Then $\hat{w}$, which is a probability distribution over the words in the vocabulary, can be obtained by applying softmax over the neural output h (Eq. 8).

$$z = \text{MLP}(w). \tag{6}$$

$$\theta = \text{Softmax}(z). \tag{7}$$

$$\hat{w}_i = \frac{\exp h_i}{\sum_{j=1}^{V} \exp h_j}, h = \beta\theta + b, \tag{8}$$

where $\beta = [\beta_1, \ldots, \beta_K]$ is topic-word weight matrix.

The reconstruction loss is the negative cross-entropy between w and the decoder output $\hat{w}$.

$$c(w, \hat{w}) = -\sum_{i=1}^{V} w_i log(\hat{w}_i). \tag{9}$$

For measuring the distribution term, we use MMD-based divergence [28] to unbiasedly estimate $\text{MMD}_\mathbf{k}(P_\Theta, Q_\Theta)$ with m samples as follows.

$$\widehat{\text{MMD}}_\mathbf{k}(Q_\Theta, P_\Theta) = \frac{1}{m(m-1)} \sum_{i \neq j} \mathbf{k}(\theta_i, \theta_j) + \frac{1}{m(m-1)} \sum_{i \neq j} \mathbf{k}\left(\theta'_i, \theta'_j\right) - \frac{2}{m^2} \sum_{i,j} \mathbf{k}\left(\theta_i, \theta'_j\right), \tag{10}$$

where $\mathbf{k}(\theta, \theta')$ is information diffusion kernel [29] with geodesic distance defined as:

$$\mathbf{k}(\theta, \theta') = \exp\left(-\arccos^2\left(\sum_{k=1}^{K} \sqrt{\theta_k \theta'_k}\right)\right),$$

$\{\theta_1, \ldots, \theta_m\}$ are sampled from Dirichlet prior P_Θ and $\{\theta'_1, \ldots, \theta'_m\}$ are sampled from posterior estimation Q_Θ obtained by Eq. 7.

In experiments, the reconstruction loss is usually magnitude larger than the distribution term, a scaling factor $1/(s \log V)$ (s is document length) is used to balance the two terms [24].

3.2.3. Integrating PPA and the pre-trained WAE for rumor detection

Gururangan et al. [25] augments the vector representation of each document through a weighted sum over topic vector θ and

the internal states of the MLP encoder. While in our experiments, weight sum fails to provide obvious performance improvement, perhaps because information contained in MLP encoder is dispensable for rumor detection. Instead, we fine-tune the encoder weights of WAE with labeled data and represent a response propagation path with a document-topic vector. Namely, the topic vector of a response propagation path can be computed by

$$\theta_{ij} = \text{Encoder}(w_{ij}). \tag{12}$$

Where w_{ij} indicates the word sequence vector of the j-th response propagation path in the i-th response propagation tree. Then we obtain the representation of a response propagation tree and predict the label of the corresponding rumor in the same way as PPA with formulas:

$$\hat{p}_i = \text{MaxPooling}(\text{dropout}(\theta_{i*})), \tag{13}$$
$$\hat{y}_i = \text{Softmax}(\text{MLP}(\hat{p}_i)). \tag{14}$$

Finally, we integrate the prediction result of PPA and that of WAE for each rumor by mixing $\bar{y}_i$ and $\hat{y}_i$:

$$y_i = \alpha\bar{y}_i + (1-\alpha)\hat{y}_i, \tag{15}$$

where $\alpha \in [0, 1]$ is the mixing proportion. We name the model PPA-WAE.

In our experiments, we also try to concatenate p_{ij} and θ_{ij} as a long vector and then go through Maxpooling and Softmax to predict the class of rumor C_i. But the model tends to be unstable even with dropout or batch normalization, perhaps because simply concatenating p_{ij} with θ_{ij} is likely to introduce noise to the model, thus increases the variance of the experimental results. Besides, additive models are good to determine the combined value of each alternative [30].

4. Experiments and results

4.1. Datasets

The proposed model is evaluated on four real-world datasets: Twitter15 [10], Twitter16 [10], Weibo [31] and PHEME 5 events dataset [6]. To the best of our knowledge, this is the first work that compares different models on all the above four data sets. Twitter15, Twitter16, Weibo and PHEME respectively contains 1490, 818, 4664 and 1972 source tweets. The Weibo dataset is a Chinese dataset, and the others are English datasets. Each source post and its corresponding replies and retweets are provided in the form of a propagation tree. The difference between reply and retweet of a source tweet is that reply contains a comment, while retweet is just a copy of source tweet. All retweets in this kind of trees are removed since they do not contribute new information to the model. For Twitter15 and Twitter16, since the original datasets do not include reply texts, we crawl all the reply texts according to given reply IDs via Twitter API[1] and we found that about 70% reply texts are still available. We named all replies we could get as valid posts. Twitter15 and Twitter16 contain four classes labels: non-rumor (NR), false rumor (FR), true rumor (TR) and unverified rumor (UR). The Weibo dataset only contains two binary labels: false rumor (FR) and true rumor (TR). PHEME dataset contains three labels: false rumor (FR), true rumor (TR) and unverified rumor(UR). PHEME 5 events dataset is composed of 5 different events. We mix PHEME dataset and shuffle all events rumors to form the similar dataset as Twitter15, Twitter16 and Weibo. For fair comparison, we conduct 5-fold cross validation following Ma et al. [1], Bian et al. [18]. Table 1 shows the statistics of the four datasets.

[1] https://dev.twitter.com/rest/public.

4.2. Baseline models

For making comprehensive comparison of our models with the state-of-the-art methods in the literature, we conduct a series experiments on four datasets. The compared methods are in the following.

-DTC A decision tree-based classifier based on various statistics features of tweets [7].

-DTR A decision tree-based ranking model proposed by Zhao et al. [4] to detect fake news by searching for enquiry phrases.

-RFC A random forest classifier that utilizes a set of handcrafted features from user, linguistic and structure properties [9].

-SVM-TS A linear SVM model that uses time-series to model the variation of news characteristics [32].

-PTK An SVM model with a propagation tree kernel [10] that identifies rumors on Twitter by capturing their similarities between pairs of propagation trees.

-RvNN A top-down or bottom-up tree-structured recursive neural network [1] for learning the propagation of rumors. The top-down network is selected since it has better performance.

-RvNN* An improved version of RvNN[2] by replacing Momentum Gradient Descent Algorithm with AdaGrad algorithm [33].

-PLAN A structure-aware hierarchical self-attention model by learning embedding vectors of propagation structures [16].

-Bi-GCN A novel bi-directional graph convolutional model by operating on both top-down and bottom-up propagation trees of rumors [18].

For Englist datasets, we tokenize propagation paths with spaCy,[3] exclude stopwords[4] and tokens shorter than two characters and those with digits or punctuation. For Weibo dataset, Jieba[5] (Chinese for "to shutter") Chinese text tokenization is used to tokenize weibo propagation paths. For the above four datasets, we use a vocabulary with 5000 most common tokens in all propagation paths for PPA to characterize the content of all paths. For response propagation paths without source tweets learnt by WAE, we use 4980 most common tokens and 20 emojis with the highest frequency to represent all response paths since emojis are very popular in social platforms (such as Twitter or Weibo) and usually convey people's emotional information and attitudes.

PPA and the encoder of WAE are jointly trained using AdaGrad algorithm and the learning rate is set to 0.05. Embedding vector of each word is randomly initialized with size of 50. The We also test pre-trained word embedding, such as Glove [34], WordVec [35] or BERT [26], but achieve no obvious performance improvements. In our experiments, pre-trained word embedding could improve the initial accuracy of the model with large margin, but in the end there is no improvement in accuracy. The mixing proportion α is set to 0.6. For pre-training WAE, we set the Dirichlet parameters to 0.1 and 0.2. We use Adam optimizer with momentum $\beta_1 = 0.99$. Learning rate is set to 0.003 to overcome initial local minimum. The dimension of latent document-topic vectors is also 50, the same as the two hidden layers of WAE encoder. Dropout over the latent document-topic vector θ_{ij} is set to 0.5, and dropout over propagation path p_{ij} is set to 0.3.

4.3. Rumor classification performance

Tables 2–4 show the performance of all compared models on Twitter15, Twitter16 and Weibo datasets. The accuracy of feature-based methods are drawn from Ma et al. [1] and Bian et al. [18]. We use the mean of 5 runs generated from different

[2] https://github.com/majingCUHK/Rumor_RvNN/pull/7.
[3] https://spacy.io/.
[4] http://snowball.tartarus.org/algorithms/english/stop.txt.
[5] https://github.com/fxsjy/jieba.

Table 1
Statistics of the datasets.

Statistics	Twitter15	Twitter16	Weibo	PHEME
Total trees	1490	818	4664	1972
Total valid posts	42756	20194	2011057	31430
True	374	205	2351	1008
False	370	205	2313	393
Unverified	374	203	0	571
Non-rumor	372	205	0	0
Propagation paths	37339	17543	1837779	20194
Avg words of paths	27.87	28.04	79.28	46.04
Avg words of response paths	13.65	13.48	14.99	30.39

Table 2
Rumor detection results on Twitter15 datasets (N: Non-Rumor; F: False Rumor; T: True Rumor; U: Unverified Rumor). We report F1-score for each individual class and accuracy (Acc.) for all testing samples.

Method	Acc.	NR F1	FR F1	TR F1	UR F1
DTC	0.454	0.733	0.355	0.317	0.415
DTR	0.409	0.501	0.311	0.364	0.473
RFC	0.565	0.810	0.422	0.401	0.543
SVM-TS	0.544	0.796	0.472	0.404	0.483
PTK	0.741	0.778	0.705	0.753	0.728
RvNN	0.717	0.673	0.734	0.794	0.665
RvNN*	0.778	0.742	0.809	0.804	0.758
PLAN	0.840	0.821	0.846	0.874	0.818
Bi-GCN	0.864	0.847	0.861	**0.917**	0.829
PPA	0.862	0.891	0.857	0.831	**0.869**
PPA-WAE	**0.873**	**0.899**	**0.881**	0.869	0.843

Table 3
Rumor detection results on Twitter16 datasets (N: Non-Rumor; F: False Rumor; T: True Rumor; U: Unverified Rumor). All measures are the same as Table 2.

Method	Acc.	NR F1	FR F1	TR F1	UR F1
DTC	0.465	0.643	0.393	0.419	0.403
DTR	0.414	0.394	0.273	0.630	0.344
RFC	0.585	0.752	0.415	0.547	0.563
SVM-TS	0.574	0.755	0.420	0.571	0.526
PTK	0.728	0.721	0.714	0.784	0.693
RvNN	0.731	0.672	0.723	0.828	0.701
RvNN*	0.788	0.763	0.778	0.853	0.761
PLAN	0.859	0.855	0.842	0.869	**0.876**
Bi-GCN	0.877	0.834	0.874	**0.927**	0.873
PPA	0.874	0.871	0.867	0.892	0.866
PPA-WAE	**0.887**	**0.882**	**0.903**	0.921	0.842

seeds for the other models for better comparison. We bold the best results of each column in the Tables.

We can observe that the performance of all feature-based methods (DTC, DTR, RFC and SVM-TS) is very poor, indicating that feature engineering is difficult to generalize well for rumor detection.

PTK utilizes SVM with a propagation tree kernel to compute the similarities between pairs of propagation trees, thus PTK is inherently inefficient for large datasets. Our proposed models outperform RvNN and RvNN* with a large margin. RvNN represents propagation tree by max-pooling the hidden vectors of leaf nodes. Each leaf node is heavily influenced by its near ancestor tweets, thus leaf nodes may contain little information of the corresponding source tweet when the path's length is long.

Our proposed model PPA-WAE is better than the newly developed propagation tree based methods, PLAN and Bi-GCN. Although PLAN achieves comparable performance with our models, the transformer networks used in the model are vary heavy and the training of PLAN is very time-consuming. Bi-GCN is very efficient, but its performance is not stable. We will compare our models with Bi-GCN in a more comprehensive way in the later discussion.

The common feature of Twitter15, Twitter16 and Weibo datasets is that the reply posts contain a lot of emotional words. PHEME 5 events dataset is a dataset of conversations around rumours associated with 5 different breaking news stories. So we choose PHEME 5 events dataset, which contains very few emotional words in reply posts as discussed in Khoo et al. [16], to check whether our models still work when propagation paths contain few emotional words. According to Table 5, PPA and PPA-WAE show comparable performance, indicating that the proposed PPA-WAE is superior to PPA when propagation path contains rich emotional words. The F1 scores of PLAN and PPA-WAE are nearly the same, because the stacked transformer blocks where PLAN used are more suitable for encoding complex sentences in PHEME dataset. Contrary to the extraordinary results achieved by PLAN on PHEME dataset, Bi-GCN performs very poorly and only gets 72.2% Macro F1 score. The possible reason is that the dataset is not balanced and Bi-GCN has the poor performance on false rumor class with only 57.0% F1 score.

Table 4
Rumor detection results on Weibo (F: False Rumor; T: True Rumor). We report precision (Prec.), recall (Rec.) and F1-score for each class and Acc. for all testing samples.

Method	Class	Acc.	Prec.	Rec.	F1
DTC	F	0.831	0.847	0.815	0.831
	T		0.815	0.824	0.819
DTR	F	0.789	0.784	0.801	0.793
	T		0.794	0.777	0.785
RFC	F	0.855	0.810	0.929	0.866
	T		0.916	0.779	0.842
SVM-TS	F	0.885	0.950	0.932	0.938
	T		0.124	0.047	0.059
PTK	F	0.891	0.876	0.913	0.894
	T		0.907	0.868	0.887
RvNN	F	0.908	0.912	0.897	0.905
	T		0.904	0.918	0.911
RvNN*	F	0.929	0.949	0.909	0.928
	T		0.911	0.950	0.930
PLAN	F	0.943	0.939	0.948	0.943
	T		0.946	0.937	0.942
Bi-GCN	F	0.961	0.961	0.964	0.961
	T		**0.962**	**0.962**	0.960
PPA	F	0.953	0.956	0.952	0.954
	T		0.949	0.953	0.951
PPA-WAE	F	**0.962**	**0.963**	**0.965**	**0.964**
	T		0.961	0.960	**0.961**

Table 5
Rumor detection results on PHEME 5 events dataset (F: False Rumor; T: True Rumor; U: Unverified Rumor). We report F1-score for each class. Besides Marco-F1 score (Khoo et al. [16]), we also provide accuracy for a more comprehensive comparison.

Method	Acc.	Macro F1	FR F1	TR F1	UR F1
RvNN	0.728	0.749	0.761	0.717	0.769
RvNN*	0.743	0.758	**0.770**	0.745	0.759
PLAN	0.785	0.772	0.753	0.828	0.735
Bi-GCN	0.722	0.677	0.570	0.792	0.675
PPA	0.802	0.773	0.699	0.852	0.767
PPA-WAE	**0.803**	**0.774**	0.687	**0.854**	**0.781**

4.4. Ablation study

To better understand each component of PPA-WAE, we compare PPA-WAE with PPA-Merge, PPA and WAE on four datasets. PPA represents our baseline model trained on propagation trees, PPA-Merge merges BOW representations of each propagation tree's all propagation paths, and WAE represents the pre-trained WAE encoder only trained on response propagation trees. The empirical results are summarized in Fig. 3.

By Fig. 3, we have the following conclusions. First, the performance of the simplest PPA-Merge is already pretty good. The possible reason is that Twitter15, Twitter16 and Weibo datasets contain rich emotional words as discussed in Khoo et al. [16]. Second, PPA is significantly superior to the PPA-Merge, indicating that organizing response tweets according to propagation paths is simple but effective. In our ablation experiments, we observe that PPA-Merge and PPA quickly reach high performance almost at the same speed. But during the training process, the performance of PPA-Merge starts to oscillate at its later stage, while the performance of PPA steady increases as time goes by. Third, WAE achieves 72.8%, 75.2% and 90.5% without utilizing source posts on Twitter15, Twitter16 and Weibo, respectively, indicating that the stance topics are important supplement for rumor detection. Weibo is a simple binary classification dataset, so WAE achieves 90.5% accuracy using only response information. Fourth, reply posts in PHEME are composed of event related conversations and very similar to source post, so PPA and WAE perform similar to each other. Finally, PPA-WAE shows the best performance which means that PPA can still benefit from WAE on Twitter15, Twitter16 and Weibo datasets.

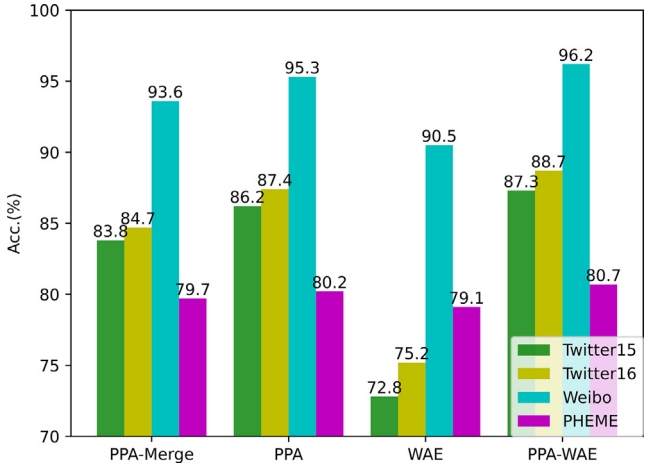

Fig. 3. Comparison of PPA-Merge, PPA, WAE and PPA-WAE on Twitter15, Twitter16, Weibo and PHEME datasets. PPA-Merge merges BoW representations of each propagation tree's all propagation paths, PPA-Merge is a simplified version of PPA without considering the propagation structure information.

Table 6
Test accuracy on Twitter15, Twitter16 and Weibo datasets under varying scales of labeled training data (200, 400, 600, 800 and 1000 documents). The training data of Twitter16 is less than 600, so we only list part of accuracy on Twitter16.

Datasets	Model	200	400	600	800	1000
Twitter15	PPA	0.706	0.777	0.825	0.845	0.862
	PPA-WAE	0.743	0.806	0.843	0.860	0.873
	Bi-GCN	0.672	0.759	0.819	0.847	0.864
Twitter16	PPA	0.752	0.840	0.876	–	–
	PPA-WAE	0.792	0.862	0.885	–	–
	Bi-GCN	0.738	0.838	0.879	–	–
Weibo	PPA	0.799	0.826	0.855	0.871	0.900
	PPA-WAE	0.839	0.864	0.869	0.885	0.913
	Bi-GCN	0.706	0.777	0.825	0.845	0.862

To further explore the effectiveness of WAE, we compare vanilla PPA and PPA-WAE at different scales of training samples (200, 400, 600, 800 and 1000 samples) on Twitter15, Twitter16 and Weibo in Table 6. As mentioned before, PHEME contains few emotional words, thus WAE has almost no performance contribution to the PHEME dataset. Therefore, we do not list the performance comparison on the PHEME dataset in Table 6. By Table 6, the less training samples, the more benefits WAE gain. Moreover, we have compared Bi-GCN with PPA-WAE in Table 6. We can see that PPA-WAE significantly outperforms Bi-GCN when training data is limited since PPA-WAE is able to learn the information from both labelled data and unlabelled data.

4.5. Topics learned by WAE

After pre-training WAE on response propagation paths, we can extract top words on each topic by ranking each column of topic-word weight matrix $\beta = [\beta_1, \ldots, \beta_K]$. Examples of topics learned on Twitter15 are provided in Table 7, and each column contains top 10 words of a particular topic. We have similar results on Twitter16 and Weibo datasets. But we just list the topics learned on Twitter15 as an illustration.

From Table 7, We can see topic words are composed of many stance words, and are similar to regular expressions designed by DTR, such as "really", "not true", while being much richer and more diverse. Moreover, the popular emojis appear in topic words frequently, thus enhancing our model's ability to infer the stance of response propagation paths.

Compared with manually designed regular expressions, document representation of response propagation paths produced by WAE can be explicitly interpreted in terms of topics. Each response propagation path can be interpreted as a combinations of various stance topics, such as pray for misfortune, appreciation for special behaviours or arguments over opinions. Then PPA-WAE is able to model the relationships between different stances and corresponding rumor classes.

Table 7
Topics learned by WAE on Twitter15.

Pray	Appreciate	Argue	Ridiculous
prayers	thank	true	😂
hell	bless	fake	disgusting
bullshit	sharing	kidding	lmao
thoughts	thanks	heartbreaking	omg
🙏	god	hoax	mtv
insane	amazing	fucking	haha
haha	cute	check	hahaha
sorry	fucking	terrible	crazy
bless	interesting	got	awesome
condolences	☺️	really	usweekly

4.6. Early detection

Early rumor detection, which is one of the most crucial goals of rumor detection, aims to detect rumor at the early stage of propagation. Since Bi-GCN [18] outperforms RvNN and other models with large margins, we simply compare PPA-WAE with Bi-GCN. Also, detection deadlines follow the strategy in Bi-GCN for fair comparison. For each detection deadline, we reconstruct the propagation trees of our models by deleting reply tweets released after the deadline.

Figs. 4–6 show the performances of Bi-GCN and PPA-WAE at different time deadlines on Twitter15, Twitter16 and Weibo, respectively. The ratio of reply tweets at different deadlines to total

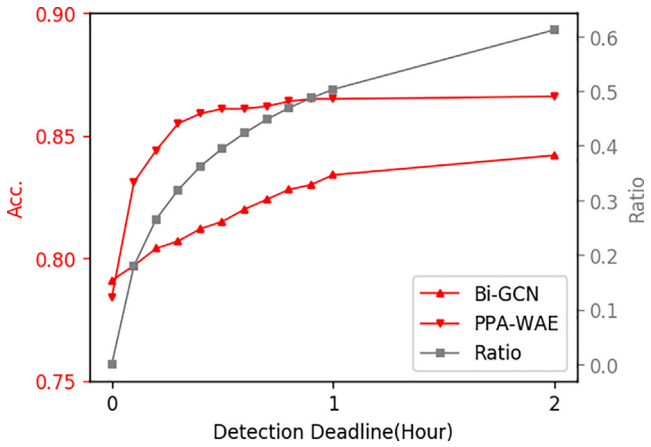

Fig. 4. Results of early detection on Twitter15.

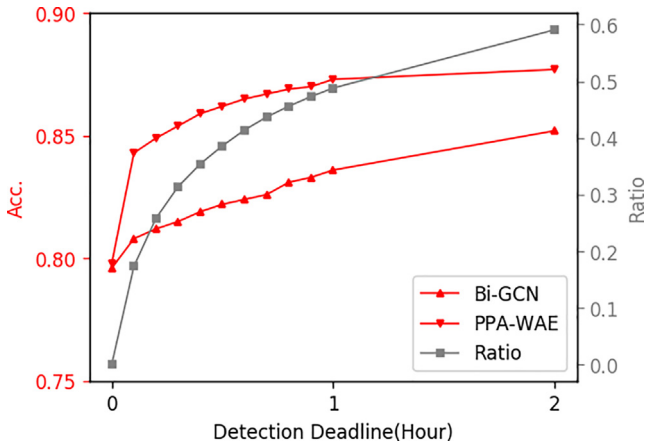

Fig. 5. Results of early detection on Twitter16.

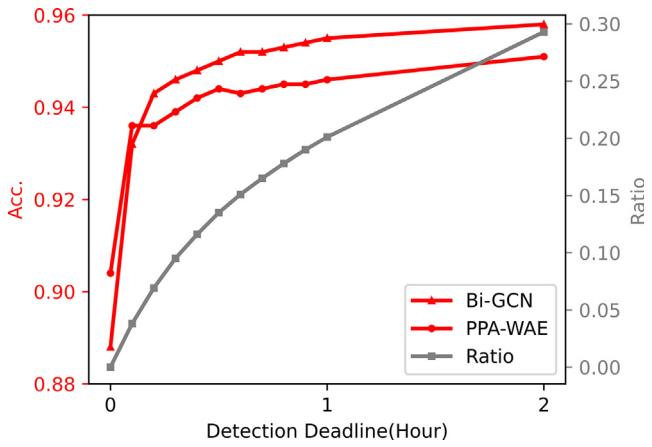

Fig. 6. Results of early detection on Weibo.

Table 8
Lightweight comparison of tree structured models on Weibo dataset. We compare each model from three aspects: 1) if the model could be trained in parallel; 2) the training time of the model; 3) the parameter size of each model.

Model	Parallel	Time(min)	Parameter(M)
RvNN	N	1259	0.56
RvNN*	N	345	0.56
PLAN	Y	2950	8.61
Bi-GCN	Y	69	0.63
PPA	Y	25	0.26
PPA-WAE	Y	84	0.51

reply tweets is also added. We can observe that the performance of PPA-WAE is superior to Bi-GCN at earlier stage, and the performance of Bi-GCN increases slowly as time goes by, while that of PPA-WAE increases quickly during the first 30 min, which is significant for rumor detection at early stage.

4.7. Lightweight comparison of tree structured models

In real application, we usually need to detect rumors online. Therefore, it is very necessary to develop a lightweight model which is efficient and portable. We then compare tree structure based models over three measurements: parallel, training time, total parameters on the largest dataset Weibo of the four datasets. The comparison results are showed in Table 8. For fair comparison, we conduct all experiments on the same Nvidia TiTAN GPU. All results are the average of 5 repeats with the same parameter settings.

First, PPA and PPA-WAE are easy to parallelize because each propagation tree is represented as a set of propagation paths. While RvNN and RvNN* can not run in parallel because of sequential property of RNN networks used in RvNN series and each tree is different. Second, PPA, PPA-WAE and Bi-GCN are comparable in terms of training speed and parameters, but Bi-GCN needs to calculate the adjacency matrix for each tree in advance. While PLAN is the heaviest model according to Table 8.

5. Conclusion

In this work, we propose a propagation path aggregating model, which integrates semantics and propagation structures of rumors in the form of propagation paths using a feed forward neural network. We train the neural topic model in Wasserstein autoencoder framework on response propagation paths for extracting stance patterns. Experimental results have shown the efficiency and effectiveness of PPA and PPA-WAE. In the future, we plan to fuse more information, such as user characteristics or common knowledge, for giving a more powerful rumor detection model.

CRediT authorship contribution statement

Pengfei Zhang: Data curation, Writing-original draft, Validation, Visualization. **Hongyan Ran:** Investigation, Conceptualization, Formal analysis. **Caiyan Jia:** Methodology, Writing-review & editing, Supervision, Funding acquisition. **Xuanya Li:** Resources, Project adiministration. **Xueming Han:** Investigation, Data curation, Validation

Declaration of Competing Interest

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

Acknowledgement

This work is supported in part by the National Nature Science Foundation of China (Nos. 61876016 and 61632004) and National Key R&D Program of China (2018AAA0100302).

References

[1] J. Ma, W. Gao, K.-F. Wong, Rumor detection on twitter with tree-structured recursive neural networks, in: Proceedings of the 56th Conference of the Association for Computational Linguistics, 2018, pp. 1980–1989.
[2] J. Chung, Ç. Gülçehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling (2014).
[3] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, R. Procter, Detection and resolution of rumours in social media: a survey, ACM Computing Surveys 51 (2018) 32:1–32:36.
[4] Z. Zhao, P. Resnick, Q. Mei, Enquiring minds: early detection of rumors in social media from enquiry posts, in: Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 1395–1405.
[5] J. Ma, W. Gao, K.-F. Wong, Detect rumor and stance jointly by neural multi-task learning, in: Proceedings of the 27th International Conference on World Wide Web, 2018, pp. 585–593.
[6] S. Kumar, K.M. Carley, Tree lstms with convolution units to predict stance and rumor veracity in social media conversations, in: A. Korhonen, D.R. Traum, L. Màrquez (Eds.), Proceedings of the 57th Conference of the Association for Computational Linguistics, 2019, pp. 5047–5058.
[7] C. Castillo, M. Mendoza, B. Poblete, Information credibility on twitter, in: Proceedings of the 20th International Conference on World Wide Web, 2011, pp. 675–684.
[8] F. Yang, Y. Liu, X. Yu, M. Yang, Automatic detection of rumor on sina weibo, In Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics (2012) 13.
[9] S. Kwon, M. Cha, K. Jung, Prominent features of rumor propagation in online social media, in: Proceedings of the 13th International Conference on Data Mining, 2013, pp. 1103–1108.
[10] J. Ma, W. Gao, K. Wong, Detect rumors in microblog posts using propagation structure via kernel learning, in: Proceedings of the 55th Conference of the Association for Computational Linguistics, 2017, pp. 708–717.
[11] J. Ma, W. Gao, P. Mitra, S. Kwon, B.J. Jansen, K. Wong, M. Cha, Detecting rumors from microblogs with recurrent neural networks, in: Proceedings of the 25th International Joint Conference on Artificial Intelligence, 2016, pp. 3818–3824.
[12] Y. Liu, B. Wu, Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks, in: Proceedings of the 34th Conference on Artificial Intelligence, 2018, pp. 354–361.
[13] N. Ruchansky, S. Seo, Y. Liu, Csi: a hybrid deep model for fake news detection, in: Proceedings of the 26th Conference on Information and Knowledge Management, 2017, pp. 797–806.
[14] C. Yuan, Q. Ma, W. Zhou, J. Han, S. Hu, Jointly embedding the local and global relations of heterogeneous graph for rumor detection, CoRR abs/1909.04465 (2019).
[15] K. Wu, S. Yang, K.Q. Zhu, False rumors detection on sina weibo by propagation structures, in: Proceedings of the 31st International Conference on Data Engineering, 2015, pp. 651–662.
[16] L.M.S. Khoo, H.L. Chieu, Z. Qian, J. Jiang, Interpretable rumor detection in microblogs by attending to user interactions, CoRR abs/2001.10667 (2020).
[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 30th Annual Conference on Neural Information Processing Systems, 2017, pp. 5998–6008.

[18] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, J. Huang, Rumor detection on social media with bi-directional graph convolutional networks, CoRR abs/2001.06362 (2020).
[19] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.
[20] Y. Miao, L. Yu, P. Blunsom, Neural variational inference for text processing, in: Proceedings of the 33nd International Conference on Machine Learning, 2016, pp. 1727–1736.
[21] A. Srivastava, C.A. Sutton, Autoencoding variational inference for topic models, in: Proceedings of the 5th International Conference on Learning Representations, 2017.
[22] R. Ding, R. Nallapati, B. Xiang, Coherence-aware neural topic modeling, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 830–836.
[23] D.P. Kingma, M. Welling, Auto-encoding variational bayes, in: Proceedings of the 2th International Conference on Learning Representations, 2013.
[24] F. Nan, R. Ding, R. Nallapati, B. Xiang, Topic modeling with wasserstein autoencoders, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, 2019, pp. 6345–6381.
[25] S. Gururangan, T. Dang, D. Card, N.A. Smith, Variational pretraining for semi-supervised text classification, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, 2019, pp. 5880–5894.
[26] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186.
[27] I. Tolstikhin, O. Bousquet, S. Gelly, B. Schoelkopf, Wasserstein auto-encoders, in: Proceedings of the 6th International Conference on Learning Representations, 2018.
[28] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Scholkopf, A.J. Smola, A kernel two-sample test, Journal of Machine Learning Research 13 (2012) 723–773.
[29] G. Lebanon, J. Lafferty, Information diffusion kernels, in: Proceedings of the 15th Annual Conference on Neural Information Processing Systems, 2002, pp. 391–398.
[30] G.D. Fensterer, Planning and assessing stability operations: a proposed value focus thinking approach, Ph.D. thesis, 2012.
[31] J. Ma, W. Gao, P. Mitra, S. Kwon, B.J. Jansen, K. Wong, M. Cha, Detecting rumors from microblogs with recurrent neural networks, in: S. Kambhampati (Ed.), Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, IJCAI/AAAI Press, New York, NY, USA, 2016, pp. 3818–3824.
[32] J. Ma, W. Gao, Z. Wei, Y. Lu, K. Wong, Detect rumors using time series of social context information on microblogging websites, in: Proceedings of the 24th International Conference on Information and Knowledge Management, 2015, pp. 1751–1754.
[33] J.C. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, Journal of Machine Learning Research (2011) 2121–2159.
[34] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: A. Moschitti, B. Pang, W. Daelemans (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1532–1543.
[35] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: Y. Bengio, Y. LeCun (Eds.), Proceedings of the 1th International Conference on Learning Representations, 2013.

Zhang Pengfei, born in 1995, post graduate. His main research interests include natural language inference and rumor detection.
Ran hongyan, born in 1992, post graduate. Her main research interests include data mining, social computing and natural language processing.

JIA Caiyan, born in 1976, professor. Her main research interests include data mining, social computing and natural language processing.
Xuanya Li received the PhD. degree from Beijing Institute of Technology, Beijing, China, in 2012. He is currently the director of Baidu Campus and the executive member of China Computer Federation. His main research interests include Internet of Things and artificial intelligence.

Han Xueming, born in 1998, post graduate. His main research interests include natural language inference and rumor detection.