Rumor Detection in Social Networks via Deep Contextual Modeling

Amir Pouran Ben Veyseh
apouranb@cs.uoregon.edu
Department of Computer and Information Science,
University of Oregon
Eugene, Oregon

My T. Thai
mythai@cise.ufl.edu
Department of Computer and Information Science and
Engineering, University of Florida
Gainesville, Florida

Thien Huu Nguyen
thien@cs.uoregon.edu
Department of Computer and Information Science,
University of Oregon
Eugene, Oregon

Dejing Dou
dou@cs.uoregon.edu
Department of Computer and Information Science,
University of Oregon
Eugene, Oregon

ABSTRACT

Fake news and rumors constitute a major problem in social networks recently. Due to the fast information propagation in social networks, it is inefficient to use human labor to detect suspicious news. Automatic rumor detection is thus necessary to prevent devastating effects of rumors on the individuals and society. Previous work has shown that in addition to the content of the news/posts and their contexts (i.e., replies), the relations or connections among those components are important to boost the rumor detection performance. In order to induce such relations between posts and contexts, the prior work has mainly relied on the inherent structures of the social networks (e.g., direct replies), ignoring the potential semantic connections between those objects. In this work, we demonstrate that such semantic relations are also helpful as they can reveal the implicit structures to better capture the patterns in the contexts for rumor detection. We propose to employ the self-attention mechanism in neural text modeling to achieve the semantic structure induction for this problem. In addition, we introduce a novel method to preserve the important information of the main news/posts in the final representations of the entire threads to further improve the performance for rumor detection. Our method matches the main post representations and the thread representations by ensuring that they predict the same latent labels in a multi-task learning framework. The extensive experiments demonstrate the effectiveness of the proposed model for rumor detection, yielding the state-of-the-art performance on recent datasets for this problem.

ACM Reference Format:
Amir Pouran Ben Veyseh, My T. Thai, Thien Huu Nguyen, and Dejing Dou. 2019. Rumor Detection in Social Networks via Deep Contextual Modeling.

ASONAM '19, August 27–30, 2019, Vancouver, BC, Canada

ACM ISBN 978-1-4503-6868-1/19/08…$15.00
https://doi.org/10.1145/3341161.3342896

In *International Conference on Advances in Social Networks Analysis and Mining (ASONAM '19), August 27–30, 2019, Vancouver, BC, Canada.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3341161.3342896

1 INTRODUCTION

With the expansion of social networks, the amount of data accessible to users is ever increasing. Social Networks (e.g., Facebook and Twitter) make it possible for their users to reach out thousands to millions of users. New contents are uploaded constantly to social networks, reflecting a near real-time view about the events in the real world. However, besides the benefit of effective information propagation, social networks present a unique challenge to verify the accuracy of the posts (i.e., fake news and rumors) (dubbed as rumor detection in this work). This problem has attracted much attention from the society recently due to its devastating implication on political, social and economical movements. For instance:

- In U.S. 2016 presidential election almost 529 different rumours about candidates were propagated via Facebook and Twitter which had influence on voters [7].
- A rumour about two explosions in White House in April 2013 in Twitter resulted in social panic and a dramatic drop in stock market [5].
- Rumors about the lost Malaysian airplane in March 2014 in Weibo, a Chinese micro blog service, made it difficult for people to follow the true news and hurt the families of the passengers [8].

It is thus the utmost interest of the social network platforms to develop effective strategies to combat against fake news and rumors. The challenges for this problem come from the fact that various sources of information might be required to recognize the misinformation, and extensive analysis and reasoning of such information sources might be needed to accurately make a decision. In order to solve this problem, the early effort from some social network platforms has relied on the reports from their users to identify the suspicious posts that are then further verified by outside fact checkers (e.g., Facebook) or automatic systems (e.g., Twitter). This approach is not very efficient as it involves substantial human effort and might need much time for the systems to recognize the fake news before they appear. Such long response time might be

already sufficient for the fake news to cause serious problems on the public, calling for more automatic methods for rumor detection to minimize the necessary human effort and expedite the recognition time. Automatic rumor detection is being studied actively in the literature and our work in this paper would contribute to this line of research by introducing a novel method to better address this problem.

There are different definitions for fake news and rumors in the literature. In this work, we will employ the definitions for these concepts as specified in [2]. In general, we can categorize rumors in two different categories:

- The rumors that contain true or false information and they would be indeed proved to be correct or incorrect (respectively) by other authorized sources later (i.e., after the time they are posted).
- The rumor that cannot be verified as presenting true or false information by the authorized sources. However, the users have identified those posts as rumors.

In other words, any piece of information whose veracity status is questionable by the time of posting would be considered as a rumor [2]. Based on this definition for rumors, rumor detection is defined as follows: Given a piece of information from social networks (e.g., user posts), we would like to predict if this piece of information is a rumor or not (i.e., cannot be verified as the true news). Due to the nature of social networks, the information that we can employ in the rumor predictions involves the conversations that the piece of information trigger (e.g., the replies from the users) and the profiles of the users who participate into such conversations. We consider the triggered conversations as the contextual information, and call a post from social networks along with its contextual information as a thread for convenience.

Different approaches based on feature engineering [3] [22], propagation pattern [6], and neural networks [10] have been proposed for rumor detection. It has been demonstrated in these works that besides the content presented in the posts and their corresponding contexts (e.g., the replies), it is also crucial to model the relations between those elements to boost the performance for rumor detection [14]. Unfortunately, these prior work have only focused on the structural relations inherited directly from the social networks (e.g., the reply relation between the main posts and the replies) and failed to exploit the implicit relations that can be induced via the semantics of the posts and the contexts. For instance, the recent work on rumor detection [14] employs the direct tree structures between user posts and replies in Twitter to guide the computation of recursive neural networks to perform prediction.

In this work, we argue that the implicit semantic relations between user posts and replies are also important for rumor detection and the models should capture them appropriately to boost the performance. Consequently, we propose a novel model for rumor detection that explicitly learns the semantic similarities between pairs of the main posts and the contextual replies using the self-attention mechanism. In the proposed model, such semantic similarities are used as the weights to compute representation vectors for the main post and the replies in a thread via the weighted combinations of the other elements in the same thread. The representation vectors are eventually aggregated into a final representation vector to represent the thread and perform the rumor prediction. Finally, as the main posts in the threads are the most important pieces of information in the threads for rumor detection, we propose to augment the final representations of the threads with the representations of the main posts to avoid the confusion for the proposed model. In particular, we introduce a novel method to ensure that the information in the final representations for the threads also involves the main information in the representation for the main posts. This is achieved by enforcing that such thread and main post representations would predict the same latent labels in a multi-task learning framework for rumor detection. The extensive experiments demonstrate the effectiveness of the proposed model and lead the the state-of-the-art performance for rumor detection on two recent benchmark datasets. In summary, our contributions in this paper include:

- We introduce a novel method for rumor detection based on the explicit modeling of semantic relations between the main posts and the contextual replies in social networks.
- We propose a novel multi-task learning framework to emphasize the main posts in the threads for rumor detection.
- We conduct extensive experiments on recent rumor detection datasets and achieve the state-of-the-art performance on those datasets.

In the following, we will first present the related work and then provide a formal definition for the task of rumor detection. Afterward, we will describe the proposed model and the experiments, followed by a conclusion in the end of this paper.

2 RELATED WORK

In general we can categorize the previous work into three categories [2]:

- Feature engineering approach: In this category, a set of features is hand-designed to transform the posts into feature representations that are then sent to some statistical model to perform classification. The typical features in this approach include the textual information, the structural evidences [3, 22], image/media content [6], and the propagation patterns of the information diffusion in social networks [9]. The success of this approach depends crucially on the quality of the hand-designed feature sets that might be suboptimal once being applied to different social networks and domains.
- Propagation based approach: In this approach, it is assumed that the propagation pattern of the rumors is different from those for non-rumor posts and such difference can be exploited to detect rumors in social networks [12]. However, One drawback of this approach is that it does not consider the textual and visual information from the post content.
- Deep learning approach: In contrast to feature engineering, this approach automatically learns effective features for the posts from data via deep learning architectures [10, 14]. The features induced by deep learning often capture the underlying representations of data, thus improving the generalization performance as well as the adaptability to new domains/social networks for rumor detection [10]. Our work in this paper follow this approach to develop a novel deep learning model for rumor detection.

The most related work to ours is the deep learning model to capture contextual information in Twitter for rumor detection in [14]. In particular, Recursive Neural Networks (RvNN) are used to compose the tree-like structures of the posts and the corresponding replies in Twitter based on their *tf-idf* representations. However, such prior work only exploits the explicit relations between the main posts and their replies from the network structures (i.e., the direct reply relations). Our work in this paper is different from the prior work as we go beyond the explicit structural relations in social networks and model the implicit relations among the posts based on their semantics to perform rumor detection.

In addition to the main task of rumor detection, there are other approaches that attempt to detect the stance of the replies toward the main post and then detect the rumor [20, 24]. It has been shown that for this task the evolution of the people's stance toward the main post is very helpful and considering time series is important for this problem. Motivated by such characteristics, [13] proposes a multi-task learning framework to simultaneously predict the stance and classify the main post for rumor detection.

3 PROBLEM DEFINITION

Following the recent work on rumor detection [14], we use Twitter as the social network in this work. Formally, an input I for rumor detection consists of the main tweet R_0 along with a set of reply tweets for this main tweet $R_1, R_2, \ldots, R_T$ (where T is the number of reply tweets): $I = (R_0, R_1, R_2, \ldots, R_T)$. I is called a thread for convenience. The goal of rumor detection is to predict whether the input thread represents a rumor or not. In order to perform such prediction, following [14], we attempt to classify I into one of the following four labels: 1) Without Rumor, (2) True Rumor, (3) False Rumor and (4) Unrecognizable. If the main tweet R_0 can be proved to be false, the label for I is "False Rumor" while the label for I is "True Rumor" if R_0 can be shown to be true.

4 MODEL

4.1 Word & Tweet Representation

We consider the input $I = (R_0, R_1, R_2, \ldots, R_T)$ as sequence of tweets where each tweet, in turn, is a sequence of words. The main tweet R_0 is put at the beginning of the tweet sequence. As the tweets might involve different numbers of words, we pad the tweets with a special token to ensure that all the tweets have the same word length N (i.e., the maximum word length of the tweets in the dataset).

In order to represent the posts and replies, for the i-th tweet R_i, we first convert its words $W_{i1}, W_{i2}, \ldots, W_{iN}$ (i.e., $R_i = W_{i1}, W_{i2}, \ldots, W_{iN}$) into their pre-trained word embeddings $e_{i1}, e_{i2}, \ldots, e_{iN}$ respectively. Afterward, we apply the max-pooling operation over such word embeddings along each dimension to obtain the representation vector h_i for R_i:

$$h_i = Elementwise\_Max(e_{i1}, e_{i2}, \ldots, e_{iN}) \quad (1)$$

This tweet-vector transformation procedure would convert the the input thread $I = (R_0, R_1, R_2, \ldots, R_T)$ into a sequence of representation vectors $(h_0, h_1, h_2, \ldots, h_T)$ (respectively) that are then fed into the following steps for further computation.

4.2 Contextualizing Tweet Representations

Due to the nature of social networks, the main tweet and the replies are not independent and the content of the main tweet or replies has substantial influence on other tweets in the same thread. In the previous work, it has been shown that capturing such relations among the main tweets and replies can help to boost the performance of rumor detection [14]. However, such previous work has only considered the explicit relation between the main tweet and its replies (i.e., the reply trees in Twitter from the network structures), neglecting the implicit relations among the tweets based on their semantic similarities. In this work, we propose to exploit such implicit semantic relations to further improve the performance for rumor detection.

In order to capture the semantic relations between the tweets, we learn the pairwise similarities among them based on the self-attention mechanism. In particular, inspired by the transformer architecture in [19], we first compute the key and query vectors for each tweet based on its representation h_i:

$$\begin{aligned} k_i &= W_k * h_i + b_k \\ q_i &= W_q * h_i + b_q \end{aligned} \quad (2)$$

Given these key and query vectors for the tweets, we obtain the similarity a_{ij} between the i-th tweet and the j-the tweet in the input thread I via the dot product:

$$a_{i,j} = k_i \cdot q_j / \gamma \quad (3)$$

where γ is a normalization factor. Once all the similarities a_{ij} for all the tweet pairs in a thread have been computed, we exploit these similarities as the weights to compute more abstract representations for the tweets based on the weighted sums:

$$h'_i = \Sigma_j a_{i,j} * h_j \quad (4)$$

In the next step, we obtain the overall representation vector h' for the input thread I by applying the max-pooling operation one more time over all the tweet representations h'_i:

$$h' = Elementwise\_Max(h'_0, h'_1, h'_2, \ldots, h'_T) \quad (5)$$

Afterward, a 2-layer feed-forward neural network followed by a softmax layer is employed to produce a probability distribution $P(y|R_0, R_1, R_2, \ldots, R_T; \theta)$ over the possible labels for rumor prediction for I (θ is the model parameter). Finally, we optimize the negative log-likelihood function to train the proposed model for rumor detection:

$$L_{label} = -\log P(y * |R_0, R_1, R_2, \ldots, R_T; \theta) \quad (6)$$

where $y*$ is the correct label for I.

4.3 Information Preservation

The current model treats the main tweets as equally important as the replies. This is undesirable as the main tweets involve the most important content in the threads over which the model should emphasize to produce good performance. Such emphasis can be done by ensuring that the overall final representations of the threads also cover the core information represented in the main tweet representations (i.e., preserving the information in the main tweets).

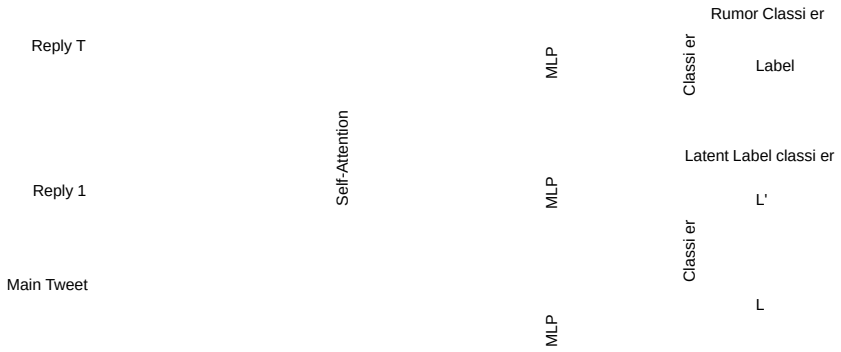


Figure 1: The Proposed Model. Inputs to the model are word embeddings. Green vectors represent replies and red vectors represent the main tweet. The self-attention component incorporates contextual information into the representation of each tweet. Max-pooled representations are fed into the rumor and latent label classi ers.

In this work, we propose to achieve this goal by enforcing the same latent labels induced by the thread representations and the main tweet representations. In particular, we rst transform the thread representation h^0 and the main tweet representation h_0 into two probability distributions over the same number K of possible latent labels $P^0(L^0|R_0, R_1, R_2, ..., R_T) = F^0(h^0)$ and $P^1(L|R_0) = F^1(h_0)$ (respectively) using two di erent 2-layer feed-forward neural networks with softmax layers in the end (i.e., F^1 and F^0 are di erent 2-layer feed-forward networks). In order to ensure that h^0 covers the information in h_0, the latent label L^0 that $P^0(L^0|R_0, R_1, R_2, ..., R_T)$ predicts should be the same as the latent label predicted by $P^1(L|R_0)$. To this end, we additionally optimize the following negative log-likelihood function for the latent labels when we train the proposed model:

$$L = \arg\max_L P^1(L|R_0) \quad (7)$$

$$L_{info} = -\log P^0(L|R_0, R_1, R_2, ..., R_T) \quad (8)$$

Consequently, the loss function to train the model in this work is the weighted sum of the rumor label loss and the information preservation loss:

$$Loss = L_{Lablel} + \alpha L_{info} \quad (9)$$

where α is the hyper-parameter controlling the contribution of the information preservation loss to the nal loss function. Figure 1 shows the main building blocks of the proposed model.

4.4 Baseline Models

The goal of the information preservation component in the proposed model is to emphasize main tweet R_0 in the nal representation of the input thread h^0. In order to demonstrate the bene ts of the latent label prediction mechanism, we explore two other methods to achieve this goal, serving as the baselines for the proposed model.

In the rst baseline, we emphasize the main tweet R_0 in h^0 by directly imposing that the thread representations h^0 and the main tweet representations h_0 are similar. In particular, we rst transform h^0 and h_0 into more abstract representation vectors l^0 and l (respectively) with the same dimension using di erent 2-layer feed-forward networks. We would then enforce that l^0 and h_0 are similar by replacing L_{info} in the proposed model with the squared di erence loss between l^0 and l:

$$L^1_{info} = ||l - l^0||^2 \quad (10)$$

where $||.||^2$ is the L_2 norm of the vector. We denote this model by Di in our experiments.

In the second baseline for the information preservation component, we observe that the thread representation h^0 is computed based on both the main tweet representation h_0 and the reply tweet representations $h_1, h_2, ..., h_T$. In order to establish a stronger influence of the main tweet on the thread representation, we can ensure that the thread representation h^0 is more similar to the main tweet representation h_0 than the reply tweet representations $h_1, h_2, ..., h_T$. In particular, we rst aggregate the reply tweet representations $h_1, h_2, ..., h_T$ into a single vector h_{rep} to facilitate the comparison with h^0 via the max-pooling operation:

$$h_{rep} = Elementwise\_Max(h_1, h_2, ..., h_T) \quad (11)$$

Afterward, we estimate the similarity s_0 between the thread representation h^0 and the main tweet representation h_0 as well as the similarity s_{rep} between the thread representation h^0 and the reply tweet representation h_{rep} by applying a feed-forward neural network FF with a single output unit (i.e., for the similarity) on the concatenation of the representations: $s_0 = FF(h^0, h_0)$; $s_{rep} = FF(h^0, h_{rep})$. Note that we ensure the similarities s_0 and s_{rep} are between 0 and 1 by introducing the sigmoid function in the end of FF. Consequently, we replace the loss function L_{info} in Equation 7 of the proposed model with the following margin loss function to push h^0 closer to h_0 than h_{rep}:

$$L^2_{info} = \max(0, s_{rep} - s_0 + \gamma) \quad (12)$$

We name this model as Discriminator in the experiments. Note that all the three variants of the information preservation component (i.e., latent label prediction, Di and Discriminator) aim to retain more information of the main tweet representation h_0 in the

thread representation h^0. However, the label prediction mechanism approaches this via the implicit constraint of the same predicted latent label from the representations while D_i targets a more extreme and direct method of similar representations. D_s discriminator, on the other hand, considers the information preservation for the main tweet in the context with the reply tweet representations and the margin loss function.

As the input for rumor detection is a sequence of tweets, a common approach in natural language processing is to handle such sequential data with Recurrent Neural Networks (RNN) (e.g., Long-short Term Memory Networks - LSTM). Consequently, in addition to the baseline models for information preservation, we investigate another baseline model to fine tune the tweet representations based on LSTM. In particular, for this baseline model, we employ a LSTM layer over the sequence of tweet representations $h_0; h_1; h_2; :::; h_T$ before the self-attention component. The hidden states of the LSTM layer $\hat{h_0}; \hat{h_1}; \hat{h_2}; :::; \hat{h_T}$ would then replace the tweet representations $h_1; h_2; :::; h_T$ respectively in the proposed model. This baseline is called *RNN* in the experiments.

5 EXPERIMENTS

5.1 Datasets, Resources & Parameters

Following the previous work on rumor detection, we use the Twitter datasets (described in [18]) for the evaluations in this paper (i.e., the Twitter 15 and Twitter 16 datasets). There are 1381 and 1118 main tweets in these datasets respectively for which a main tweet would correspond to one tree of replies.

Regarding the parameters and resources for the proposed model, we use the Glove [16] embedding (of size 300) to initialize the word vectors. 300 hidden units are employed for the key and query vectors in Equations 2. The feed-forward layer for the rumor classifier has two layers with 200 hidden units. The feed-forward layer for the information preservation component, on the other hand, has two layers with 100 hidden units and employ three latent labels for prediction. We use the Adagrad optimizer with initial learning rate of 0.3 with the trade-off parameter of $= 1$ for the loss function. Following the previous work [14], we use the 5-fold cross validation procedure to tune the parameters and obtain the performance for the models in this work.

5.2 Comparing to the state-of-the-art

This section compares the proposed model (called *Semantic Graph*) with the state-of-the-art model on our datasets. For performance measure we use the accuracy on all classes and F1 score per class for each dataset. We compare with two types of models: 1) Feature based models: These models rely on feature engineering to extract features for such statistical models as Decision Tree, SVM and Random Forest [3, 9, 11, 12, 21, 23]. 2) Deep learning models: These models use deep learning models to learn features for rumor detection (i.e., Recurrent Neural Networks or Recursive Neural Networks with GRU-RNN, BU-RvNN and TD-RvNN in [10, 14]). Tables 2 and 3 show the results on Twitter 15 and Twitter 16 respectively.

Table 2: Model Performance on Twitter 15.

Model	Accuracy	F1 NR	F1 FR	F1 TR	F1 UR
DTR [23]	0.409	0.501	0.311	0.364	0.473
DTC [3]	0.454	0.733	0.355	0.317	0.415
RFC [9]	0.565	0.810	0.422	0.401	0.543
SVM-TS [11]	0.544	0.796	0.472	0.404	0.483
SVM-BOW [14]	0.548	0.564	0.524	0.582	0.512
SVM-HK [21]	0.493	0.650	0.439	0.342	0.336
SVM-TK [12]	0.667	0.619	0.669	0.772	0.645
GRU-RNN [10]	0.641	0.684	0.634	0.688	0.571
BU-RvNN [14]	0.708	0.695	0.728	0.759	0.653
TD-RvNN [14]	0.723	0.682	0.758	0.821	0.654
Semantic Graph	0.770	0.814	0.764	0.775	0.743

Table 3: Model Performance on Twitter 16.

Model	Accuracy	F1 NR	F1 FR	F1 TR	F1 UR
DTR [23]	0.414	0.394	0.273	0.630	0.344
DTC [3]	0.465	0.643	0.393	0.419	0.403
RFC [9]	0.585	0.752	0.415	0.547	0.563
SVM-TS [11]	0.574	0.755	0.420	0.571	0.526
SVM-BOW [14]	0.585	0.553	0.655	0.582	0.578
SVM-HK [21]	0.511	0.648	0.434	0.473	0.451
SVM-TK [12]	0.662	0.643	0.623	0.783	0.655
GRU-RNN [10]	0.633	0.617	0.715	0.577	0.527
BU-RvNN [14]	0.718	0.723	0.712	0.779	0.659
TD-RvNN [14]	0.737	0.662	0.743	0.835	0.708
Semantic Graph	0.768	0.825	0.751	0.768	0.789

These tables show that deep learning models outperform feature based models for rumor detection due to the capacity of the deep learning models to automatically learn effective features from data. In addition, comparing *Semantic Graph* and RvNN models with GRU-RNN, we see that the structural information (e.g., the reply or semantic relations) helps to improve the performance for rumor detection. Finally, by incorporating implicit semantic relations among all the tweets in a thread, *Semantic Graph* achieves the state-of-the-art performance on both datasets in terms of accuracy, and outperforms all the other models in three out of four classes in terms of F1 Score. This clearly demonstrates the effectiveness of the proposed method for rumor detection.

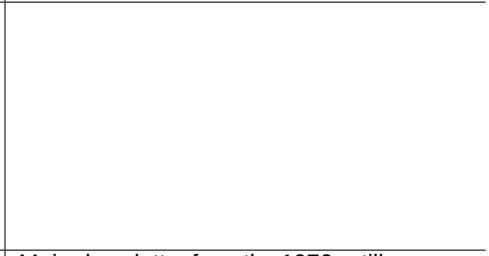
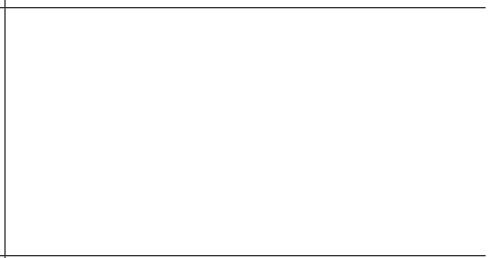
5.3 Word Embedding

The input to the proposed model is the embeddings of the words in the tweets. This section compares the performance of the proposed model when different word embeddings are employed. In particular, we investigate two types of word embeddings in this work:

1) Contextualized Embeddings : Contextualized word embeddings involve pre-trained models that can compute the embedding for a word based on its context. The following contextualized word embeddings are compared in this work:

ELMo: This is a bidirectional language model with multiple layers of LSTMs [17]. We use the ELMo embeddings of dimension 1024 in the experiments.

Table 1: Heatmap of attention. Numbers in front of each tweet show the index of the tweet in the heatmap. Numbers in heatmap show the attention weights between the correspnoding tweets indexed at the columns and rows.

False Rumor	True Rumor
 Main : really? amber alert website goes dark under government shutdown (6) Reply : How? They only need webmaster to run it (7) Reply : barry proves it is not (4) Reply : What's up with this. It's up nicely (13)	 Main : lego letter from the 1970s still offers a powerful message to parents 40 years later (8) Reply : lego lives on my brother. my daughter and my grandson all enjoyed it thro the generations (8) Reply : forty years later still screams from parents as we step on the bloody stuff (7) Reply : times may change but the truth of this lego letter from the 1970s never has in 40 years via (0)
Unverified	Non-Rumor
 Main : donald trump: " ... laziness is a trait in blacks. it really is, i believe that. itâ€™s not anything they can control.â€¦ (2) Reply : that is not from trump. it is found in a book by a former trump employee, he alleges he heard trump say that (0) Reply : now he is trying to win the african american vote and wonders why it is not working. the gop wed itself to terminal stupidity (5) Reply : it is appalling that this nonsensical revisionism is being forcefully persued (7)	 Main : colombian gov. and left-wing farc rebel movement agree cease re to end decades-long conict (11) Reply : hmmm, not exactly as promising as first reported, but a step in the right direction at least. (10) Reply : I really hope that it will last (9) Reply : funny how the farc still stealing children and bombing towns as i write this (9)

BERT: This is a bidirectional language model trained with a transformer [4]. We employ the vectors in the last layer of the BERT model (with dimension 768) for the word embeddings in the experiments.
GPT: Similar to BERT, GPT uses a transformer to train a language model. However, this is an unidirectional model [18]. In the experiments, we use the last layer of GPT with dimension of 768 for the word embeddings.

2) Non-Contextualized Embeddings : In contrast to the contextualized embeddings, the same embedding vector is produced for each word in the vocabulary regardless of its context. The following non-contextualized embeddings are considered in our experiments:

Word2Vec: This is a word embedding method based the skip gram model in [15]. The Word2Vec embeddings in our experiments have 300 dimensions.
fastText: This is a library for efficient text representation learning. We use the model trained on Wikipedia, UMBC webbase corpus and statmt.org news dataset [1]. The embedding dimension is 300.

Table 4: Model Performance on Twitter 15

Model	Accuracy	F1 NR	F1 FR	F1 TR	F1 UR
Word2Vec	0.635	0.648	0.650	0.663	0.641
fastText	0.628	0.621	0.663	0.645	0.632
ELMO	0.698	0.679	0.695	0.712	0.640
BERT	0.729	0.701	0.720	0.756	0.661
GPT	0.767	0.810	0.765	0.771	0.751
Glove	0.770	0.814	0.764	0.775	0.743

Table 5: Model Performance on Twitter 16

Model	Accuracy	F1 NR	F1 FR	F1 TR	F1 UR
Word2Vec	0.643	0.630	0.629	0.642	0.663
FastText	0.651	0.639	0.626	0.630	0.671
ELMO	0.711	0.683	0.703	0.699	0.715
BERT	0.742	0.801	0.755	0.779	0.770
GPT	0.761	0.821	0.742	0.768	0.783
Glove	0.768	0.825	0.751	0.768	0.789

Table 4 and 5 show the performance of the proposed model with di erent word embeddings on Twitter 15 and Twitter 16 respectively. These tables show that the contextualized word embeddings in general outperform the non-contextualized word embeddings for rumor detection. This result is expected since the contextualized word embedding capture the appropriate meanings of the words based on their context while non-contextualized models use the same representations for all meanings of the words. However, among all embedding models, Glove actually achieves the best performance. Our hypothesis is that the language in our rumor detection datasets is very similar to those in the corpus used to train Glove (i.e., Wikipedia). This leads to better overlapping between the vocabularies of our datasets and Glove embeddings to provide more prior knowledge for the rumor detection models.

5.4 Ablation Study

There are two major components in the proposed model for rumor detection, i.e., seft-attention (called SA) and information preservation with latent label prediction (called Prediction). In order to see the contributions of these components for the proposed model, we take turns to exclude these components from the models. The rst section in Tables 6 and 7 shows the performance of the proposed model when the SA and/or Prediction component are removed from the model on the Twitter 15 and 16 datasets respectively. As we can see from the tables, when SA is excluded, the model performance drops dramatically. This shows that contextual information is important for representing each tweet. In addition, the information preservation component with latent label prediction is also necessary for the model to achieve the best performance. It suggests that the multi-task learning setting proposed by this work is e ective to preserve important information during the model computation for rumor detection.

In order to assess e ectiveness of the proposed latent label prediction for preserving information about the main tweet, we conduct experiments with the Di and Discriminator baselines. The results are presented in the second sections of Tables 6 and 7, showing that the proposed latent label prediction mechanism outperforms the baselines Di and Discriminator for information preservation. We attribute the poorer performance of Di and Discriminator to the fact that their constraint mechanisms for the similarity of the main tweet and the thread representations have an e ect to eliminate the information about the reply tweets on the nal thread representation. In particular Di enforces the main tweet representation and the thread representation to be similar over all the possible dimensions, leaving no space for the reply tweet representations in the thread representation Discriminator, on the other hand, explicitly separates the reply tweet representations from the overall thread representation. The limited reply tweet information in the thread representation hinders an important source of information and hurts the performance for rumor detection. The latent label prediction mechanism avoids this problem as it only imposes the similarity of the main tweet representation and the thread representation over the most important dimensions via the same latent label, reserving some space for the reply tweet information in the nal thread representation for better performance.

Finally the last rows in Tables 6 and 7 report the performance of the RNN baseline mentioned in Section IV.D. As we can see in the tables, although RNN is a common component in many NLP tasks, it hurts the performance of the proposed model for rumor detection. This is due to our consideration of the input thread $I = [R_0; R_1; R_2; \ldots; R_T]$ as a sequence of tweets that do not re ect its original tree structures in social networks. Better modeling approach that inherits such tree structures might help to improve the representation of the tweets and further advance the proposed model.

Table 6: Ablation study on Twitter 15

Model	Accuracy	F1 NR	F1 FR	F1 TR	F1 UR
Full Model	0.770	0.814	0.764	0.775	0.743
-SA	0.652	0.643	0.629	0.675	0.660
-Prediction	0.751	0.789	0.756	0.742	0.729
-SA -Prediction	0.601	0.612	0.603	0.663	0.549
Di	0.755	0.740	0.751	0.772	0.731
Discriminator	0.763	0.791	0.758	0.772	0.740
RNN	0.753	0.746	0.755	0.769	0.740

Table 7: Ablation study on Twitter 16

Model	Accuracy	F1 NR	F1 FR	F1 TR	F1 UR
Full Model	0.768	0.825	0.751	0.768	0.789
-SA	0.665	0.643	0.651	0.608	0.689
-Prediction	0.749	0.795	0.723	0.741	0.770
-SA -Prediction	0.592	0.604	0.624	0.598	0.651
Di	0.756	0.812	0.742	0.751	0.761
Discriminator	0.761	0.819	0.745	0.758	0.769
RNN	0.755	0.809	0.756	0.759	0.772