Adversary-Aware Rumor Detection

Yun-Zhu Song,[1] **Yi-Syuan Chen,**[1] **Yi-Ting Chang,**[1] **Shao-Yu Weng,**[2] **Hong-Han Shuai,**[1]
[1]National Yang Ming Chiao Tung University, Taiwan
[2]National Tsing Hua University, Taiwan
{yunzhusong.eed07g, yschen.eed09g, joshchang0111.eed06}@nctu.edu.tw
nthu106071077@gapp.nthu.edu.tw, hhshuai@nctu.edu.tw

Abstract

While social media becomes a primary source of news now, it also becomes more challenging for people to distinguish the rumors and non-rumors, which attracts malicious manipulation and may lead to public health harm or economic loss. Consequently, many rumor detection models have been proposed to automatically detect the rumors based on the contents and propagation path. However, most previous works are not aware of malicious attacks, e.g., framing. Therefore, we propose a novel rumor detection framework, Adversary-Aware Rumor Detection including Weighted-Edge Transformer-Graph Network and Position-aware Adversarial Response Generator, to improve the vulnerability of detection models. To the best of our knowledge, this is the first work that can generate the adversarial response with the consideration of the response position. Experimental results show that our model achieves the state-of-the-art on various rumor detection tasks by the proposed Weighted-Edge Transformer-Graph Network and can maintain the performance under the adversarial response attack after the adversarial learning by Position-aware Adversarial Response Generator.[1]

1 Introduction

With the popularity and accessibility of social media, social media becomes the primary source for obtaining information.[2] Compared with traditional news, posts on social media are usually with shorter lengths and faster transmission speed, which also increases the difficulty of message verification. As such, social media are increasingly targeted for manipulation, leading to tremendous economic losses,

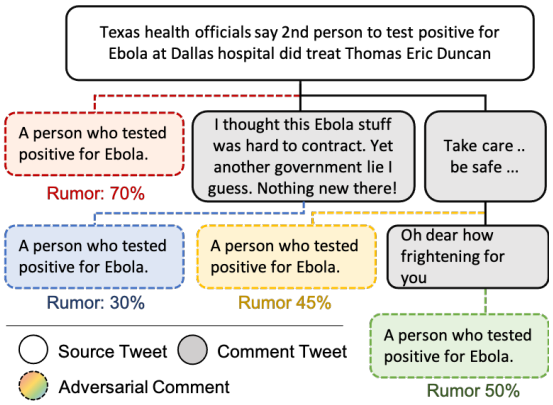

Figure 1: Position test for an adversarial response. The reply position can make an influence on the detection model.

and even deaths.[3] Take the COVID-19 pandemic as an example, a newly published study shows that at least some 800 people died because of a rumor about drinking highly-concentrated alcohol can disinfect their bodies (Islam et al., 2020). Therefore, fighting against misinformation in social networks gains a great deal of attention and becomes essential and inevitable.

In this paper, we study the problem of rumor detection on social media, where a rumor is defined as an unverified and instrumentally relevant information statement in circulation (DiFonzo and Bordia, 2007; Zubiaga et al., 2018). Given a conversation thread, including a source post and related responses, the rumor detection task aims to determine whether the source post is a rumor or not. Previous works of rumor detection can be categorized into three classes according to the data usage. Content-based approaches only use the textual information of the source posts and the user responses (Ma et al., 2016, 2019), while graph-

[1]The codes are released as a public download at https://github.com/yunzhusong/AARD.
[2]https://pewrsr.ch/3nzYpQd
[3]https://s3.amazonaws.com/media.mediapost.com/uploads/EconomicCostOfFakeNews.pdf

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1371–1382

based methods (Ma et al., 2017, 2018; Bian et al., 2020) consider the message propagation paths or model the propagation paths as a tree. In addition to considering the textual information and propagation paths, user-based methods take user profiles into consideration (Giudice, 2010; Liu et al., 2015).

However, two challenges of detecting rumors have not been completely addressed. 1) *Robustness to different responses.* Previous works of rumor detection take all the responses in the conversation thread into consideration and extract important information in a data-driven manner. However, not all responses can help detect rumors, especially for the malicious framing responses, i.e., the responses promote a particular misleading interpretation. As such, it is necessary to provide a learning mechanism to enable the selection of important responses. 2) *Vulnerability to malicious attack.* Most of the existing methods only rely on datasets, which may be vulnerable to the adversarial attack, e.g., the attack of Twitter bots. (Ma et al., 2019) make the first attempt to utilize a GAN-based approach to produce adversarial text. Nevertheless, it does not consider the graph structure of the conversation thread, i.e., the generator cannot determine which responses it should reply to. In fact, the reply position can make an influence on the detection model. As shown in Figure 1, the predicted probability that the source post is a rumor ranges from 30% to 70% according to different positions of attached adversarial responses. It is challenging to generate an adversarial attack by simultaneously considering both structural and textual information since the gradient-based methods cannot be directly applied due to the discrete nature of text and structure.

To address these two challenges, we propose a novel framework, namely, Adversary-Aware Rumor Detection (AARD), which includes i) Weighted-Edge Transformer-Graph Network (WETGN) and ii) Position-aware Adversarial Response Generator (PARG). Specifically, given a source post, responses, and propagation structure as an input, we use a transformer-based encoder to encode each token in the whole conversation thread to exploit the existing pre-trained knowledge. Each token can jointly attend to different tokens regardless of the token position, which gives the model the flexibility to break the distance limit in the sequence. Since the transformer layer only takes the responses in the conversation thread as a sequential input, the propagation path is not considered. Therefore, a Graph Convolutional Network (GCN) is applied to embed the structure by taking the token embeddings as node features, and aggregates the features according to the propagation paths. Inspired by (Veličković et al., 2018), we construct the edge features from the incident nodes and build an edge filter before the GCN layers to address the first challenge. As such, we can leverage the advantages of both transformer and graph neural networks.

Moreover, to address the second challenge, we build a Position-aware Adversarial Response Generator (PARG) to train the detector by adding an adversarial response to the conversation thread. Specifically, based on a transformer-based encoder-decoder framework, PARG takes the source post with part of the corresponding responses as input to generate an adversarial response. Nevertheless, choosing the attached position for the structure-aware detection model is also crucial. PARG is trained to select the position by considering the correlation between the generated response and each of the existing posts. However, the position selection involves the argmax function, which is a non-differentiable operation. Therefore, to enable the backpropagation of gradients from the detector, PARG instead predicts the probabilities of attaching the generated response to each existing response. When updating the edge weights of the attached edges in the detection model, the generator can use the gradient to correct the predicted probabilities.

By fine-tuning WETGN with the adversarial data generated by PARG, WETGN is equipped with a certain degree of resistance to attack and maintains the performance on clean datasets. Nevertheless, although an attacker can generate adversarial examples with the detection model, it may create nonsense sentences, which can be manually excluded (noticeability). On the other hand, imposing constraints on the generated examples decreases the possibility of finding effective adversarial examples (success rate). This paper designs a training pipeline to strike a balance between success rate and noticeability. As such, the attacker is trained to decrease the detection performance and approach the real responses simultaneously.

Extensive experimental results manifest that the proposed WETGN outperforms state-of-the-art approaches on three rumor detection benchmarks by at least 4.9%, 2.89%, and 3.87% on Twitter15,

Twitter16, and Pheme datasets. At the same time, AARD can resist the adversarial attack. Moreover, the proposed PARG can successfully attack existing detection models with a success rate of at least 25.08%. The success rate can be significantly reduced after fine-tuning, which shows the compatibility and usefulness of PARG.

2 Related Work

Early works rely on textual content to verify the authenticity of social media posts. For example, Badaskar et al. (2008) quantify the frequency of uncommon phrases in the articles and syntactic and semantic checking, while Potthast et al. (2018) detect the truthfulness of news by analyzing its writing style. Ma et al. (2016) use recurrent neural networks to learn both the temporal and textual representation of the source posts and user responses, which highly improves prior methods that utilize hand-crafted features. Also, Volkova et al. (2017) extract text features with LSTM and CNN structures to make the prediction.

On the other hand, a recent line of studies focuses on automatically detecting rumors based on the tree structure of the conversation thread (Ma et al., 2017; Wei et al., 2019; Kumar and Carley, 2019; Lu and Li, 2020). For instance, Ma et al. (2018) build a tree-structured recursive neural network to catch the hidden features from either top-down or bottom-up propagation structure and text content. However, it can only obtain the information of one propagation structure and ignore the other. To solve this problem, Bian et al. (2020) use the GCN-based model to embed both propagation and dispersion structures and enable the proposed method to process graph/tree structures and learn higher-level representation more conductive to rumor detection. Besides, by utilizing the hierarchical structure in the conversation thread (i.e., parent, child, before, after and self), Khoo et al. (2020) adopt the idea in Shaw et al. (2018) to perform structure-aware self-attention.

In addition, stance and user information are also used in several studies. By using stance prediction as the auxiliary task with multi-task learning, Wei et al. (2019); Li et al. (2019); Kumar and Carley (2019) have demonstrated that stance prediction plays a vital role in rumor detection. Furthermore, Li et al. (2019) incorporate the collected user credibility to supervise the detection model. Lu and Li (2020) construct the propagation network by using retweet sequences of users with user profiles to capture the correlation between user propagation and its source post. The uniqueness of our work lies in improving the vulnerability of detection models.

Due to the small or non-diversified training data, a recent line of studies utilizes the adversarial learning (Ma et al., 2019; Yang et al., 2020) or data augmentation to improve the detectors. For example, Ma et al. (2019) propose an RNN-based GAN model, where the generator aims to generate conflicting information in the conversation thread, and the discriminator is forced to learn more robust features. On the other hand, Han et al. (2019) augment data by using semantic relatedness to assign pseudo labels to unlabeled tweets. However, the structural information is important but not considered in these previous works.

3 Methodology

3.1 Problem Formulation

Given a conversation thread comprised of a source post and the corresponding responses, rumor detection aims to determine whether the claim of the source post is a rumor or not. Let $X = \{x_0, x_1, \cdots, x_i, \cdots, x_N\}$ denote a conversation thread, where x_0 represents the source post and $\{x_i\}_{i=1}^{N}$ represents the N responses. A graph $G = \langle V, E \rangle$ is constructed by taking each element in X as a node and the interactions between elements as the edge connections to form the node set V and the edge set E, respectively. For example, if nodes x_u and x_v have a direct interaction (e.g., commenting or retweeting) in the same conversation thread, an edge $(x_u, x_v) \in E$ is constructed accordingly. Due to the nature of social media, the graph G is an acyclic tree. Let $y \in \{rumor, non\text{-}rumor\}$ be the class label. Rumor detection aims to predict y given the graph G.

3.2 Rumor Detection Model

Transformer Encoder: To obtain the representation of text contents, we adopt the transformer-based encoder to explore the pre-trained knowledge. We first flatten the tree-structured graph in the chronological order, which constitutes a source post followed by a sequence of responses. Specifically, the source post and each response are started by a special token [CLS] and ended by another special token [SEP] to indicate the separation of nodes (Liu and Lapata, 2019). In this setting, we

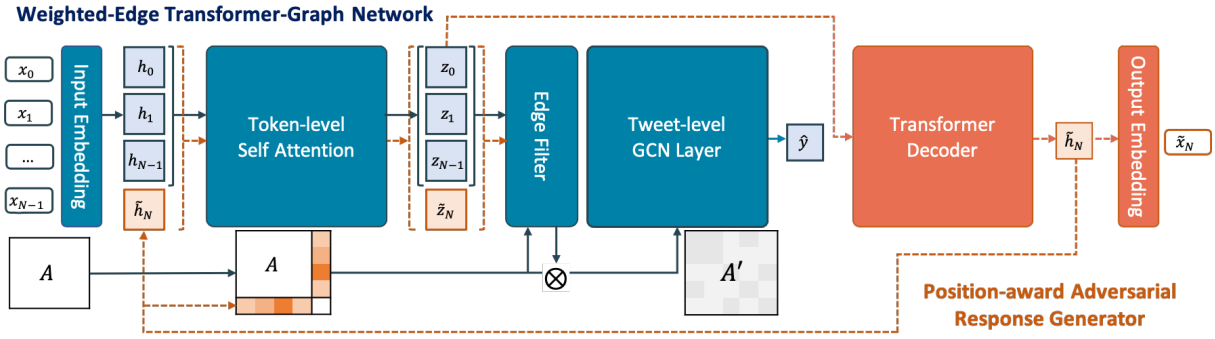

Figure 2: Overview of adversary-aware rumor detection framework.

allow each token to jointly attend to nodes in different positions for better capturing semantics. Let $h_i^{(0)} \in \mathbb{R}^{|x_i| \times d}$ denotes the d-dimensional embedding of a node x_i, which is constructed by token embedding, segment embedding, and position embedding (Devlin et al., 2019). The embedding of a conversation thread $H^{(0)}$ can thus be obtained as follows:

$$h_i^{(0)} = E_{in}(x_i) = E_{tok}(x_i) + E_{seg}(x_i) + E_{pos}(x_i),$$

$$H^{(0)} = [h_0^{(0)} \,||\, h_1^{(0)} \,||\, ... \,||\, h_N^{(0)}] \in \mathbb{R}^{M \times d},$$

where $||$ is the concatenation operation, and $M = |x_1| + |x_2| + ... + |x_N|$ indicates the length of input sequence. The embedding is passed through several transformer layers. At layer $l+1$, the features from previous layer $H^{(l)}$ is transformed by three linear layers to form the query Q, key K, and value V matrices, and the output $H^{(l+1)}$ is computed as follows:

$$Q = H^{(l)} W_q, K = H^{(l)} W_k, V = H^{(l)} W_v,$$

$$H^{(l+1)} = softmax(\frac{QK^T}{\sqrt{d_k}} V),$$

where W_q, W_k, and W_v are trainable parameters and d_k is the scaling factor to prevent small gradients. The feature of token [CLS] from the last layer is taken to represent each node, which are denoted as $Z = [z_0 \,||\, z_1 \,||\, ... \,||\, z_N] \in \mathbb{R}^{N \times d}$.

GCN Classifier: The interactions between responses, e.g., commenting or retweeting, are essential information for the detection model to judge the source post (Castillo et al., 2011). The responses not only contain the users' opinions but also reveal the propagation paths through social media. Since Graph Convolutional Network (GCN) is one of the most effective models for graph-structured data modeling, we leverage GCN to consider the propagation path. The message propagation function of a multi-layer GCN defined in the first-order approximation of Chebyshev polynomials is derived as follows:

$$Z^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} Z^{(l)} W^{(l)}),$$

where $Z^{(l)} \in \mathbb{R}^{N \times d}$ is a hidden feature matrix at the l-th layer, $\widetilde{A} = A + I$ is a binary adjacency matrix with self-connection, $\widetilde{D}_{u,u} = \sum_v \widetilde{A}_{u,v}$ is a degree matrix, $W^{(l)}$ is a learnable matrix, and $Z^{(0)} = Z$ is the node features. Although GCN has been proved to be effective for extracting structural information, the information may not be faithful after aggregation with specific nodes, e.g. framing responses. Therefore, considering the potential existence of various redundant or adversarial messages, we propose to filter the edges by learning the importance of edges before the aggregation.[4] Specifically, based on the node features extracted by the transformer encoder, the importance of an edge $e_{u,v}$ between nodes x_u and x_v is constructed as follows:

$$e_{u,v} = f_{edge}(z_u, z_v) = \sigma([z_u \,||\, z_v] W_{edge} + b_{edge}),$$

where W_{edge} and b_{edge} are trainable parameters. The predicted importance are then used to construct a weighted adjacency matrix as follows:

$$A'_{u,v} = A_{u,v} + I_{u,v} + e_{u,v}.$$

For the final prediction, the model considers the entire graph by taking mean pooling over all convolved node features instead of only taking the root

[4] We also simulate the generation process of adversarial responses in the next subsection and assess the attack performance in the experiment.

node's attribute. The prediction is calculated from the feature matrix of the last GCN layer L:

$$\hat{y} = softmax(Mean(z_0^L, z_1^L, .., z_N^L)W_o + b_o),$$

where W_o and b_o are trainable parameters.

3.3 Adversarial Response Generation

To further improve the vulnerability of the detection model, we explore adversarial learning under the setting of white-box attack, i.e., the parameters and gradients of the detector are exposed when updating the attacker. Specifically, we design a response generation model that attaches new adversarial responses to the conversation threads as an attacker against the detection model. For the text generation model, we adopt an encoder-decoder framework with transformer layers, which shows outstanding performance in text generation (Liu and Lapata, 2019). However, the gradients cannot be backpropagated from the detector to the generator for updating, due to the non-differentiable argmax function (de Masson d'Autume et al., 2019) in generation. To solve this problem, we tie the generator's output layer E_{out} with the embedding layer E_{in}, which means the weights of the two layers are mutually transposed. In this way, the features before the argmax function can be treated as the embedding of the generated response. Given an input sequence $\{x_i\}_{i=0}^{n-1}$, a response is generated as follows:

$$h_i = E_{in}(x_i),$$

$$\widetilde{h}_n = f_{dec}(f_{enc}(h_0, \cdots, h_{n-1})),$$

$$\widetilde{x}_n = argmax(softmax(E_{out}(\widetilde{h}_n))),$$

$$E_{in}(\widetilde{x}_n) \sim \widetilde{h}_n,$$

where f_{enc} and f_{dec} are the encoder and decoder. The features $\widetilde{h}_n$ can thus be directly used in the detector without breaking the gradient path. Besides, to reduce the model complexity, the encoder is shared between the generator and detector.

Nevertheless, for rumor detectors that incorporate propagation paths, the location to attach the generated responses is also a crucial problem. Similar to text generation, the operation of choosing the attached position is also discrete. To enable the model to simultaneously learn the position for generating responses, the generation model additionally predicts the edge weights $\{e_{n,i}\}_{i=0}^{n-1}$ between the generated response $\widetilde{x}_n$ and all existing nodes in the training process with Gumble softmax function (Jang et al., 2016), i.e.,

$$\pi_i = [\widetilde{h}_n \,||\, h_i]W_p + b_p, \; \forall i \in (0, \cdots n-1),$$

$$e'_{n,j} = softmax(log(\pi_i + g_i)/\tau),$$

where g_i is i.i.d sampled from Gumble distribution (0,1), τ is a hyper-parameter for controlling the smoothness of output distribution, and $W_p \in \mathbb{R}^{2d \times 1}$ and b_p are trainable parameters. A higher edge weight indicates a higher possibility that the attack can succeed at a specific position. It is worth noting that we focus on generating only one response to attack the model for validating the performance of the proposed attack. The proposed model can be extended to iteratively generate adversarial responses at different positions in the conversation thread.

Table 1: Statistics of the datasets.

	Twitter15	Twitter16	Pheme
# of tree	1458	818	3720
# of node	41154	18618	67238
# of rumors	1086	613	1863
# of non-rumors	372	205	1863
Avg text length	15.84	15.87	17.79
Max text length	136	383	78
Min text length	2	2	2

3.4 Training Pipeline

The adversarial examples cannot only demonstrate the weakness of the detection model, but also provide the opportunity to improve the vulnerability. However, there is a trade-off between the attack success rate and the noticeability. To strike a balance between them, the attacker should generate a response that is close to the real ones. Base on this idea, we i) decompose the conversation threads into several subtrees for the attacker to predict the next real response and ii) design a three-stage learning pipeline to mutually learn the attacker and the detector.

Firstly, the generator is trained along with the detector to increase the detection accuracy. To generate quality responses, we provide the generator target sentence by decomposing one conversation tree into several subtrees. That is, given a subsequence of the conversation thread, the goal of the generator is to synthesize the next real response x. We only train the decoder layer θ_{dec} of the generation model while fixing the parameters of the encoder. For the detection model, the trainable layers are the encoder θ_{enc} layer, filter layer θ_{filter}

and the GCN layer θ_{gcn}. The objective function of generator is the binary-cross entropy for rumor classification and the cross entropy for text perplexity $L_{txt} = -\frac{1}{|x|}\sum_{m=1}^{|x|} logP_{gen}(w_m|w_{1:m-1})$, while the detector minimizes the rumor classification loss. The loss of the first stage (L^{1st}) is derived by summarizing L_{gen} and L_{det} with a weight λ:

$$L_{gen}(\theta_{dec}) = CE(\hat{y}, y) + L_{txt},$$

$$L_{det}(\theta_{enc}, \theta_{filter}, \theta_{gcn}) = CE(\hat{y}, y),$$

$$L^{1st} = \lambda L_{gen} + (1-\lambda)L_{det}.$$

The second training stage is to train the generator while fixing the detector. In this stage, the goal of the generator is to generate a response that can confuse the detector as an attacker. The detector takes the adversarial data as the input and makes a prediction, and the target label is reversed $\bar{y}$, i.e., the rumor becomes non-rumor and vice versa. To make the generated text unnoticeable, i.e., similar to human written sentences, the attacker is also trained to optimize L_{txt}. Therefore, the loss of the second stage is

$$L^{2nd} = L_{gen}(\theta_{dec}) = CE(\hat{y}, \bar{y}) + L_{txt}.$$

The third training stage is to fine-tune the detector under the fixed attacker. The detector is trained on the adversarial data and optimized to make the correct prediction. This training equips the detector with the ability to resist the attack and also learn to filter out the potential redundant or adversarial messages. The objective function is as follows:

$$L^{3rd} = L_{det}(\theta_{enc}, \theta_{filter}, \theta_{gcn}) = CE(\hat{y}, y).$$

4 Experimental Results

4.1 Experiment Settings

Datasets. We evaluate the proposed AARD on three public datasets including Pheme (Zubiaga et al., 2016), Twitter15 and Twitter16 (Ma et al., 2017) datasets since these datasets contain source posts, the corresponding responses, and the rumor labels. The original labels of Twitter15 and Twitter16 datasets include four classes, i.e., true rumor, false rumor, unverified rumor, and non-rumor. In this paper, we focus on differentiating rumors from non-rumors, and thus regard the first three classes as rumors. The Pheme dataset is collected based on five events with two classes, i.e., rumor and non-rumor. Due to the privacy protection policy of Twitter, the contents of responses are not included in the dataset. Therefore, we crawl the contents of responses by ourselves. If all contents have already been removed, we delete the empty tweet from the tree. Meanwhile, following the previous work (Khoo et al., 2020), we also eliminate retweets with an empty text description. The statistics are shown in Table 1.

Baselines. The selection of the baselines follows two criteria: 1) "rumor detection" or "rumor veracity classification" and 2) availability of source codes. Specifically, this paper designs a rumor detector and generator for the "rumor detection" task, which is a binary classification task. In contrast, the "rumor veracity classification" is a four-class classification task (non-rumor/true-rumor/false-rumor/unverified rumor). As Ma et al. (2019) also target the binary classification task between rumor and non-rumor, it is selected as baselines. For other works focusing on rumor veracity classification (Ma et al., 2018; Kumar and Carley, 2019; Yang et al., 2020; Khoo et al., 2020; Bian et al., 2020), one possible way for comparing with these works is to reimplement the models and change their settings to the binary classification. Therefore, Bian et al. (2020) and Ma et al. (2018) are used as baselines by reimplementing and changing the labels as binary classification. Unfortunately, it is hard to compare with some baselines that do not release the source code (Yang et al., 2020) or require additional information, e.g., user information and the stance of each response (Kumar and Carley, 2019). Finally, the baseline methods are listed: (1) RvNN (Ma et al., 2018), based on tree-structured recursive neural networks with GRU units to obtain representations from the propagation structure in the bottom-up (BURvNN) or top-down (TDRvNN) manners, (2) GAN-GRU (Ma et al., 2019), the GAN-style learning model where the discriminator and generator are recurrent neural networks with GRU units, (3) BiGCN (Bian et al., 2020), the GCN-based model that can embed both propagation and dispersion structures and enhance the root node features, and (4) GCAN (Lu and Li, 2020), which learns the retweet propagation features based on user features by a structure employed convolution and recurrent neural networks.

Implementation Details. We use the same hyperparameters for all datasets. Specifically, the batch

Table 2: Rumor/non-rumor detection results. The '-EF' means the model without the edge filter, and the '-DD' is trained without the data decomposition, while the '-PARG' indicates the detector without adversarial learning.

		Twitter15				Twitter16				Pheme			
Method	Class	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
BURvNN	NR	85.02	69.77	72.43	71.01	80.08	64.13	52.68	57.78	76.83	78.28	74.36	76.22
	R		90.51	89.31	89.89		84.47	89.60	86.95		75.62	79.30	77.38
TDRvNN	NR	86.53	69.19	72.23	76.10	84.83	58.05	65.57	76.11	70.43	67.43	81.77	73.42
	R		92.44	91.10	89.85		**93.78**	90.25	87.03		77.78	59.09	65.75
GAN-GRU	NR	83.50	67.46	67.57	67.46	80.74	64.16	54.27	58.51	75.13	75.61	74.19	74.94
	R		88.98	88.94	88.95		85.45	89.59	87.44		74.74	76.07	75.38
BiGCN	NR	88.16	84.15	62.83	70.49	87.30	**87.12**	52.17	63.34	80.97	79.14	84.18	81.46
	R		89.17	96.10	92.39		87.15	**98.03**	92.14		83.09	77.78	80.21
AARD	NR	93.06	85.63	**87.84**	86.50	89.73	83.49	74.63	78.73	83.93	82.59	**86.93**	84.45
	R		**95.87**	94.84	95.32		91.59	94.95	93.23		**86.52**	80.91	83.20
-EF	NR	92.99	87.93	84.59	85.93	89.17	82.63	75.12	78.05	84.11	84.83	83.44	83.99
	R		84.87	95.85	95.33		91.72	94.06	92.79		83.83	84.78	84.18
-DD	NR	90.86	80.24	87.03	83.05	89.19	79.22	**78.54**	78.85	83.09	85.56	79.79	82.50
	R		95.50	92.17	93.72		92.59	92.88	92.73		81.12	86.40	83.62
-PARG	NR	**93.47**	**89.06**	85.68	**86.93**	**90.19**	87.02	73.17	**79.29**	**84.84**	**86.31**	82.90	**84.48**
	R		95.26	**96.13**	**95.64**		91.22	96.08	**93.57**		83.74	**86.77**	**85.15**

Table 3: Detection results for True/False Rumor. The results of GCAN are from the original paper. ($\star$) Indicates the results are taken from the reference.

Twitter15					
Method	Class	Acc.	Prec.	Rec.	F1
GCAN*	TR	87.67	-	-	-
	FR		82.57	82.95	82.50
AARD	TR	**94.13**	95.54	92.78	94.11
	FR		92.87	95.49	94.14
Twitter16					
Method	Class	Acc.	Prec.	Rec.	F1
GCAN*	TR	90.84	-	-	-
	FR		75.94	76.32	75.93
AARD	TR	**92.01**	94.45	89.51	91.87
	FR		90.13	94.22	91.99

sizes of the detector and the generator are 48. The learning rate of the generator is 0.002 and warms up for 2000 steps. The learning rate of the decoder is set to 0.002. The token embeddings are initialized from BERT, therefore, the settings refer to the pretrained model bert-base-uncased (Devlin et al., 2019). The Transformer Encoder has 12 self-attention layers, and the layer number of GCN (L) is 2. The loss weight λ is 0.5.

Evaluation metrics. The evaluation metrics include accuracy, precision, recall and F1 score of two classes. We split each dataset into five-fold (80% for training and 20% for testing), and report the average results.

4.2 Rumor Detection

Overall Performance. Table 2 shows the performance of all models on the rumor detection task. The results manifest that the proposed AARD outperforms all the state-of-the-art models by at least 4.9%, 2.89%, and 3.87% on Twitter15, Twitter16, and Pheme datasets, respectively. Compared with the methods that use the recursive (BURvNN and TDRvNN) or recurrent (GAN-GRU) neural network, graph-network based models achieve better results, indicating that the propagation structure contains important information when detecting rumors. Different from the BiGCN, which uses the tf-idf vectors as the node features, AARD uses self-attention layers to encode the posts as the node features. The Transformer encoder enables the model to embed tokens across nodes, thus strengthening the node representation.

Moreover, the bottom rows of Table 2 show the ablation studies of the proposed AARD. The results show that both edge filter and data decomposition play important roles. On the other hand, the model can achieve a promising performance on the rumor detection task without adversarial learning. The goal of adversarial learning is to address the second challenge, i.e., vulnerability to malicious attacks. Accordingly, the Position-aware Adversarial Response Generator (PARG) is designed to improve the robustness under a malicious attack. As the original testing dataset is clean (without attacks) or only contains few manual attacks, the detection accuracy may not be significantly improved. However, when a detector is without adversarial learning (denoted by AARD-PARG in Table 2 and WETGN in Table 4), the performance drastically decreases when encountering an attack (adding one adversarial node to the conversation tree), which can be alleviated by fine-tuning on the adversar-

Table 4: Results of adversarial attack and training, where '-EF' indicates the detector without edge filter.

	Twitter15			Twitter16			Pheme		
	Accuracy	Diff.	ASR	Accuracy	Diff.	ASR	Accuracy	Diff.	ASR
WETGN	93.47 → 71.13	-23.34	28.87	90.19 → 59.80	-30.39	40.20	84.84 → 74.92	-9.92	25.08
-EF	92.99 → 62.20	-30.79	37.80	89.17 → 51.49	-37.68	48.51	84.11 → 74.59	-9.52	25.41
	After Adversarial Training								
	Adversarial		Clean		Adversarial		Clean		Adversarial		Clean	
	Acc.	Diff.	Acc.	Diff.	Acc.	Diff.	Acc.	Diff.	Acc.	Diff.	Acc.	Diff.
AARD	92.44	-1.03	93.06	-0.41	87.94	-2.25	89.73	-0.46	82.53	-2.31	83.93	-0.91
-EF	90.72	-2.27	91.41	-1.58	86.70	-2.47	83.92	-5.25	78.49	-5.62	82.98	-1.13

Table 5: Generated adversarial examples of testing data, showing the source post, responses and the generated response. The user names are replaced to remain anonymous.

R	**Source:** st. Louis co police tell me ofcr shot a man who pointed handgun at him at chambers & sheffingdell at about 1 a.m **Response:** @name1 Uh! oh. this is serious and officially out of control. # @name1 thank you so much for getting this scoop. people have been on pins and needles at the Reddit live feed. @name1 how many weeks until police interview witnesses? @name2 @name3 or the video! **Generated Response:** @name1 @name2 is not a false alarm.
NR	**Source:** BBC reports that broadcaster Sir Terry Wogan has died of cancer aged 77 **Response:** @name1 A huge loss.. no more his whit and sing song voice.. tragic news @name1 Sad sad news, yet another British icon gone [UNK] @name1 @name2 another national treasure gone! so sad. @name1 RIP Wogan! thoughts and prayers with your loved ones! @name1 #prayforpudsey @name1 #terrywogan cancer is an absolute bitch , so many celebrated people have been taken by it in just a few weeks **Generated Response:** @name1 @name2 #I can not believe it is a true story.

ial examples. The detailed analysis of adversarial learning is discussed in Sec. 4.3.

To further analyze the impact of data quantity on model performance, we train the models under different quantities of data, ranging from 5% to 100%, and evaluate them on the same testing set. Figure 4 shows the results, which indicate that our model can still achieve leading performance even with minimal training data.

True/False Rumor Detection. We separately compare AARD with another graph-based model, GCAN, since GCAN focuses on true rumor/false rumor classification task and evaluates on Twitter15 and Twitter16.[5] Table 3 shows the results of true rumor/false rumor classification, which indicates that AARD also has an excellent performance in the rumor classification. We consider it is because differentiating the false rumor from true rumor also requires the model to carefully examine the responses.

Early Detection. Early detection aims to detect rumors in the early stage, which is an important indicator for evaluating the detection model. We refer (Bian et al., 2020) and (Ma et al., 2019) to construct the detection deadlines of Twitter15 and Pheme datasets and only use the responses released before the deadlines to evaluate the accuracy. Figure 3 compares the accuracy with different detection deadlines. At the early stage, i.e., when a post just came out with extremely few responses, the accuracy of different models is around 0.75 on the Twitter15 dataset. After just a few minutes, the accuracy of our model reaches 0.85, whereas the accuracy of baselines only approximates 0.8. For the Pheme dataset, we squeeze the time sequence and find that the performances of all models become stable but our model stably outperforms others.

4.3 Adversarial Attack

Table 4 shows the model performance under an adversarial attack generated by PARG. The notation "→" indicates the performance before and after the attack, while "Diff." and "ASR" represent the accuracy difference and the Attack Success Rate (ASR) of PARG, respectively. The results indicate that the proposed PARG significantly reduces the accuracy of the detectors. The ASR is lower on the Pheme dataset than on Twitter15 and Twitter16 datasets because Pheme is a much larger dataset than the others. Therefore, the detector can learn more indicated features from Pheme and be more robust. Moreover, by comparing the performance of the detector with (WETGN) and without (-EF) edge filter, adding the edge filter can help the detector resist the attack, that is, the "Diff." is lower on

[5]The GCAN requires user profiles for training, which are not crawled in our datasets. Therefore, GCAN is not compared in the R/NR classification.

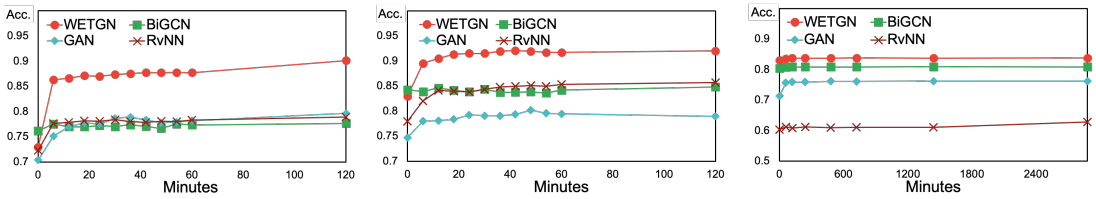

Figure 3: Early detection on Twitter15 (left), Twitter16 (middle) and Pheme (right).

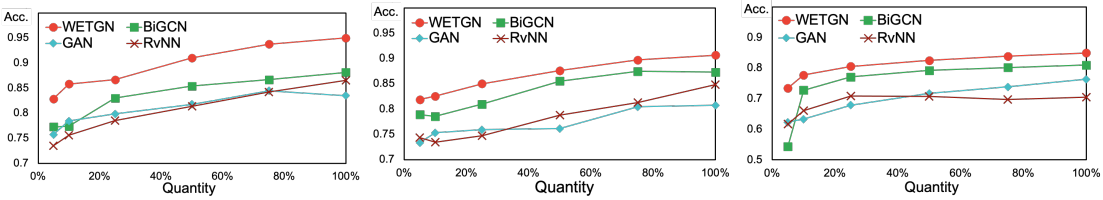

Figure 4: Data scarcity on Twitter15 (left), Twitter16 (middle) and Pheme (right).

WETGN for Twitter15 and Twitter16. In addition, we use the adversarial samples to fine-tune the detector (AARD). The bottom of Table 4 shows the results of the fine-tuned model, where the Adversarial/Clean indicates the accuracy tested on the dataset with/without adversarial attacks. The performance of the AARD without the edge filter is also provided, which also suggests the edge filter can improve the robustness.

When a detector is without adversarial learning (WETGN), the performance decreases by at least 20% on Twitter15 and Twitter16 datasets when encountering an attack. In contrast, the proposed detector with adversarial learning can maintain the performance even when the attacker has access to the model's parameters (white-box attack). It is worth noting that there may be a trade-off between the adversarial accuracy (test on adversary data) and the clean accuracy (test on clean data) (Raghunathan et al., 2019), depending on how we fine-tune the detection models, e.g., only using the adversarial data or using both kinds of data. The AARD is only fine-tuned on the adversarial data. By adjusting the experimental settings, the clean accuracy can be further improved in exchange for adversarial accuracy. Compared AARD to WETGN, it suggests that the fine-tuned detection model can resist the attack (adversarial accuracy increases from 71.13 to 92.44) while almost not affecting the clean accuracy (from 93.47 to 93.06) on the Twitter15 dataset.

Two examples of the generated adversarial responses that attack successfully are shown in Table 5. In the first example, the source post is a rumor, and PARG alters the prediction by inserting a response "not a false alarm", which conveys a signal that it is actually not a rumor. For the second example, which is a non-rumor, PARG attacks it with a certain attitude "can not believe" to deny that it is a "true" story. Similar responses can also be found in the real response written by human. If a rumor detector only captures simple patterns, it may easily misclassify the above examples and fail to adversarial attacks.

5 Conclusion

In this paper, we propose a novel rumor detection framework, AARD, to improve the vulnerability of detection models, which includes the Weighted-Edge Transformer-Graph Network (WETGN) and the Position-aware Adversarial Response Generator (PARG). Overall evaluation and ablation study results show the effectiveness of the proposed rumor detector on three public datasets. In addition, the adversarial attack results show the benefit of fine-tuning with the adversarial responses generated by PARG. In the future, we plan to further study the model generalization on rumor veracity classification tasks and combine the response stances.

Acknowledgements

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grants MOST-109-2221-E-009-114-MY3, MOST-109-2221-E-001-015, MOST-109-2221-E-009-097, and MOST-109-2218-E-009-016. We are grateful to the National Center for High-performance Computing for computer time and facilities.

References

Sameer Badaskar, Sachin Agarwal, and Shilpa Arora. 2008. Identifying real or fake articles: Towards better language modeling. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 549–556.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684.

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

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

Katherine Del Giudice. 2010. Crowdsourcing credibility: The impact of audience feedback on web page credibility. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–9.

S Han, J Gao, and F Ciravegna. 2019. Data augmentation for rumor detection using context-sensitive neural language model with large-scale credibility corpus. In *Learning from Limited Labeled Data: ICLR 2019 Workshop*. OpenReview.

Md Saiful Islam, Tonmoy Sarkar, Sazzad Hossain Khan, Abu-Hena Mostofa Kamal, SM Murshid Hasan, Alamgir Kabir, Dalia Yeasmin, Mohammad Ariful Islam, Kamal Ibne Amin Chowdhury, Kazi Selim Anwar, et al. 2020. Covid-19–related infodemic and its impact on public health: A global social media analysis. *The American Journal of Tropical Medicine and Hygiene*, 103(4):1621.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Ling Min Serena Khoo, Hai Leong Chieu, Zhong Qian, and Jing Jiang. 2020. Interpretable rumor detection in microblogs by attending to user interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8783–8790.

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

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

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

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Yi-Ju Lu and Cheng-Te Li. 2020. GCAN: Graph-aware co-attention networks for explainable fake news detection on social media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 505–514, Online. Association for Computational Linguistics.

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.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. Association for Computational Linguistics.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2019. Detect rumors on twitter by promoting information campaigns with generative adversarial learning. In *The World Wide Web Conference*, pages 3049–3055.

Cyprien de Masson d'Autume, Shakir Mohamed, Mihaela Rosca, and Jack Rae. 2019. Training language gans from scratch. In *Advances in Neural Information Processing Systems*, pages 4300–4311.

Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A stylometric inquiry into hyperpartisan and fake news. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 231–240, Melbourne, Australia. Association for Computational Linguistics.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C Duchi, and Percy Liang. 2019. Adversarial training can hurt generalization. *arXiv preprint arXiv:1906.06032*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas. 2017. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on Twitter. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 647–653, Vancouver, Canada. Association for Computational Linguistics.

Penghui Wei, Nan Xu, and Wenji Mao. 2019. Modeling conversation structure and temporal dynamics for jointly predicting rumor stance and veracity. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4787–4798, Hong Kong, China. Association for Computational Linguistics.

Xiaoyu Yang, Yuefei Lyu, Tian Tian, Yifei Liu, Yudong Liu, and Xi Zhang. 2020. Rumor detection on social media with graph structured adversarial learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI*, pages 1417–1423.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, 51(2):1–36.

Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2016. Learning reporting dynamics during breaking news for rumour detection in social media. *arXiv preprint arXiv:1610.07363*.

Appendix

.1 Case Study

Table 6 and Table 7 show more generated examples of our Position-aware Adversarial Response Generator (PARG) for non-rumors and rumors, respectively. We highlight the possible detection signals for the rumor detectors. The detector may learn a better representation, when there are more controversial responses. These examples also show the good generation quality, which may help the noticeability.

Table 6: Generated adversarial examples of non-rumors. The table shows the source post, responses and the generated response for one data. The user names are replaced to remain anonymous.

Source: Videos show suffering of starving in Syrian town of #Madaya. **Response:** @name1 How about videos that show how many are benefiting. People would like to hear that side as well. @name1 DJ Trump and his stormtrumpers say let buddy Putin do whatever. @name1 You know what they r starving people/children as well in my Canadian city I pity the people but I take care of our own @name1 And who pray tell is starving these poor unfortunate people? @name1 If some of these countries truly loved their people.. I think we might be a better world... what a concept eh @name1 This war crime has been going on for months #cnn or do you break news on a need to know bases? @name1 Praying for their relief @name1 So sad her husband got refugee status in Germany and never looked back. @name1 This breaks my heart! @name1 The USA has the same problem. **Generated Response:** @name1 @name2: The world is not a better world.
Source: Breaking: patient being tested for Ebola in Kansas **Response:** @name1 Quick send him to the Westboro baptist church! @name1: Breaking: patient being tested for Ebola in Kansas this shit is getting out of hand @name1 Lock the door to all non-Americans attempting to flee from there to here until this is over. All except Americans!!! @name1 Why not just prevent travel from west Africa before all of this crap started happening @name1 But wait I thought there was no danger and we shouldn't be worried about the US, oh look like everything else that's ... @name1: Patient being tested for Ebola in Kansas sure this is how the walking dead began. @name1 This is getting serious by the minute in Kansas it is spreading fast @name1 You believe the government can handle this go talk to the Katrina survivors @name1: Breaking: patient being tested for Ebola in Kansas oh hell no @name1 How is these people are being infected with Ebola why are these people allowed to travel **Generated Response:** @name1 @name2: The first person to be tested for Ebola in the US.
Source: Elvis Presley was born on this day in 1935 **Response:** @name1 I have been to his birthplace, Tupelo, a week ago... @name1 @name2 What a beautiful voice @name1 @name2 And he's still dead. @name1 @name2 If Elvis was alive today he would be... 81 hundred pounds. @name1 @name2 Wow it's also David Bowie's birthday great day for music lovers? @name1 Elvis Presley.. I salute you the maestro of country music and rock.. Happy birthday and forever you are remembered. **Generated Response:** @name1 @name2 I'm still alive!

Table 7: Generated adversarial examples of rumors. The table shows the source post, responses and the generated response for one data. The user names are replaced to remain anonymous.

Source: The world is running out of chocolate, world's largest chocolate manufacturer warns **Response:** @name1 Let them eat buiscuits!! @name1 Try something else or chocs are gone for good @name1 To quote a great man. 'I don't believe it!' nooo... time to hoard #chocoholic @name1: The world is running out of chocolate @name1 Ukraine president has lots for sale... @name2 This jst sounds like a re-packaged story from the last 3 years. 'blame Asia' - 'blame poor harvest @name1 @name2 nooo! @name1 Where is congress on the really important stuff like this? where do republicans stand? how about the democrats? **Generated Response:** @name1: the world is running out of chocolate?
Source: 'Nine Britons, 23 US citizens and 80 children' feared dead after #MH17 jet 'shot down' [URL] **Response:** @name1 230 + dead in #gaza! Murdered openly by #Israel #Genocideingaza @name1 Shot down-that's not confirmed yet-just speculation-two crashes of a Malaysian airline 777 within 4 months that's a fact @name2 now that's disturbing! @name1 The added horror here will be if no-one does anything about it. If it was a Russian missile Putin will need to answer for it @name1 It is so very tragic and saddening that so many lives were lost, no matter whether they were American or Briton or neither. @name1 Why comment on the US citizen number? Are there lives worth more than other nations? @name1 I bet each one blames the other, but my thoughts are the rebels/Russians may be in line as the suspects. @name1 A tragic day-my deepest condolences and thoughts go to the relatives and loved ones of the 295 people **Generated Response:** @MH17 @name1: A list of the dead in the Malaysia airlines plane shot down.
Source: CDC has confirmed that the patient in Dallas has tested positive for the Ebola virus. We'll have more coming up on Khou 11 news **Response:** @name1 What?! I need to read about this. Is it saying this originated here in the states or is this someone who traveled abroad @name1: CDC has confirmed that the patient in Dallas has tested positive for the Ebola virus. Basically everyone in Texas is dead... RT @name1 CDC has confirmed that the patient in Dallas has tested positive for the Ebola virus. We'll have more... @name3 @name4 @name1 following strict medical guidelines is probably more effective than praying. right about less... @name1 We in the US are safe, we have the needs to treat Ebola at small numbers here. Plus he was traveling from west @name1: CDC has confirmed that the patient in Dallas has tested positive for the Ebola virus. g2g leaving the country @name5 @name1: CDC has confirmed that the patient in Dallas has tested positive for the Ebola virus... **Generated Response:** @name1 @name2: A patient in Texas has tested positive for the Ebola virus.
Source: Scientist releases this horrifying picture of a puppy-sized spider he found in the rainforest [URL][URL] **Response:** @name1 Oh my god @name1 @name2 enjoy @name3 I'd die... @name1: scientist releases pic of puppy-sized spider in rainforest @name1 love of god @name1 I hate spiders @name1 Oh hell no ! ! @name1:... puppy-sized spider he found in the rainforest Ummm @name4 @name1: puppy-sized spider found in the rainforest @name1 @name5 yo don't RT this shit on my tl wtf my nigga why does this exist?? @name1 Paper trained yet? @name1 @name6 that's it, burning down the rainforest. @name1 Does it come when you call it by name? @name1: scientist releases this horrifying picture of a puppy-sized spider @name7. @name1 Remind me to never visit the rainforest. **Generated Response:** @name1 @name2:a spider found in rainforest in the rainforest is not the same thing.