Rumor Detection on Social Media with Hierarchical Adversarial Training

Shiwen Ni, Jiawen Li and Hung-Yu Kao*

Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan
{P78083033, P78073012}@gs.ncku.edu.tw, hykao@mail.ncku.edu.tw

Abstract

The proliferation of rumors on social media has a huge impact on society. However, natural language text is high-dimensional and sparse, and the same rumor may be expressed in hundreds of ways on social media. As such, the robustness and generalization of the current rumor detection model are put into question. We propose a new hierarchical model called HAT-RD, which is divided into two categories: post-level modules and event-level modules. HAT-RD adopts a novel hierarchical adversarial training method based on gradient ascent by adding adversarial perturbations to the embedding layers both of post-level modules and event-level modules to deceive the detector. At the same time, the detector uses stochastic gradient descent to minimize the adversarial risk to learn a more robust model. In this way, the post-level and event-level sample spaces are enhanced, and experiments indicate that the model drift into an area with a flat loss landscape that leads to better generalization. Experiments on two real-world datasets demonstrate that our model achieves better results than state-of-the-art methods.

Introduction

Today, social media has become a popular news source for many. However, without automatic rumor detection systems, social media can be a breeding ground for rumors. Rumors can seriously affect people's lives. For instance, during the early outbreak of the current COVID-19 pandemic, rumors about it appeared and spread on social media, leading to some serious consequences. For example, rumors about a national lockdown in the United States fueled panic buying in groceries and toilet papers, disrupting the supply chain, exacerbating the demand-supply gap and worsening the issue of food insecurity among the socioeconomically disadvantaged and other vulnerable populations (Tasnim, Hossain, and Mazumder 2020). Setting up automatic rumor detection is therefore essential.

Automatic rumor detection is extremely challenging, and the greatest difficulty lies in spotting camouflaged rumors. As the saying goes, "Rumour has a hundred mouths." These words indicate that the ways rumors are expressed constantly change as they spread. Some malicious rumormongers may deliberately modify rumor text information to escape manual detection. Variability and disguise are the main characteristics of rumors, which means that a robust automatic rumor detection model is necessary. This is, therefore, one of the main reasons why the current rumor detection model is difficult to apply in practice.

Unfortunately, most current rumor detection models are not robust enough to spot the various changes and disguises used during the rumor propagation process. As shown in figure 1, we simulated the constantly changing process of rumors during their propagation and found that the general deep learning model was too sensitive to sentence changes and disguise. A Bert-base model trained on the data set PHEME 17 has a prediction confidence of 0.85 for the rumor *"Police say shots fired at 3 #ottawa sites National War Memorial, Parliament Hill, and now Rideau shopping centre"*, but when the input is changed to *" According to the government authority report: The shootings took place at three #ottawa locations the National War Memorial parliament Hill and now the Rideau shopping centre"* model's prediction confidence decreased from 0.85 to 0.47. The main meaning and label of the input rumor text did not change, but the model predicted incorrectly. The robustness and generalization of the model are not enough, and the changes of a few words and sentences may cause significant changes in the prediction results. This is one of the main reasons why current rumor detection models perform well in rumor detection data sets but cannot maintain high accuracy in real-world environments.

We designed a novel rumor detection model called HAT-RD to enhance the generalization ability and robustness of an automatic rumor detection model. In order to make full use of available information on social media, our model detected rumors base on an event. An event in this task means a tweet object that includes a source post and a certain number of replies. To make full use of the tweet object information and obtain a high-level representation, we took a hierarchical architecture as the skeleton of our model. Using more adversarial data to train the model can enhance the robustness and generalization of the model. However, natural language text space is sparse, and it is impossible to exhaust all possible changes manually to train a robust model. We attacked the sample space of post-level and event-level respectively to comprehensively improve the robustness of the model against changes in the text. The main contributions of this paper can be summarized as follows:

*Corresponding author

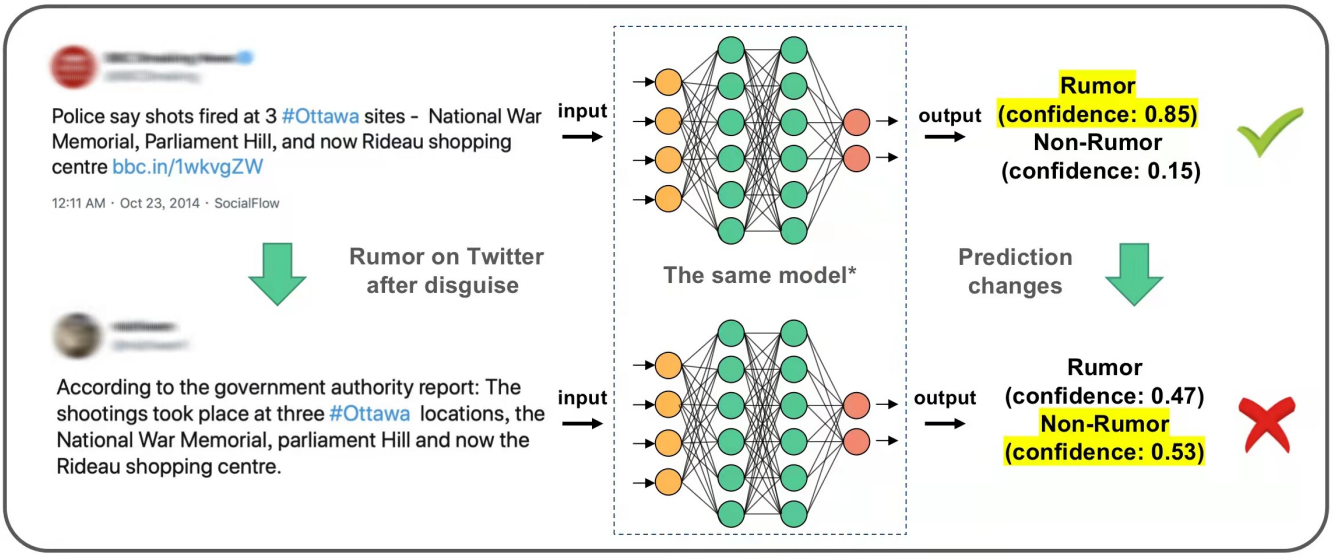


Figure 1: The robust of rumor detection models. Rumor dtection model* is a well trained BERT on PHEME 17, and this rumor example also comes from the real dataset PHEME 17.

- We proposed an end-to-end hierarchical framework that jointly exploits post-level and event-level information for rumor detection.
- We developed a hierarchical adversarial learning method that encourages the model to provide robust predictions under the perturbed post-level and event-level embeddings.
- We evaluated the proposed model HAT-RD on two real-world datasets. The results demonstrate that our model outperforms state-of-the-art models.
- We prove through experiments that the proposed hierarchical adversarial learning method can enhance the robustness and generalization of the model and prevent the model from being deceived by disguised rumors.

Related Work

Rumor Detection

Scientists around the world have been paying more attention to rumor detection in recent years. With the development of artificial intelligence, existing automated rumor detection methods are mainly based on deep neural networks. MA et al. (2016) are the first to use a deep learning network, an RNN-based model, for automatic misinformation detection. Chen et al. (2018); Yu et al. (2019) proposed an attention mechanism into an RNN or CNN model to process a certain number of sequential posts for debunking rumors. Ajao, Bhowmik, and Zargari (2018) proposed a framework combining CNN and LSTM to classify rumors. Shu et al. (2019) delved into an explainable rumor detection model by using both news content and user comments. Guo et al. (2018); Sujana, Li, and Kao (2020) detected rumors by creating a hierarchical neural network to obtain higher-level textual information representations. Yang et al. (2018) proposed a rumor detection model that can handle both text and images. Ruchansky, Seo, and Liu (2017) analyzed articles and extracted user characteristics to debunk rumors. (Ma, Gao, and Wong 2018) constructed a recursive neural network to handle conversational structure. Their model was presented as a bottom-up and top-down propagation tree-structured neural network. Li, Sujana, and Kao (2020); Li, Ni, and Kao (2020) used a variable-structure graph neural network to simulate rumor propagation and obtain more precise information representations in the rumor detection task. Ni, Li, and Kao (2021) used multi-view attention networks to simultaneously capture clue words in the rumor text and suspicious users in the propagation structure. Li, Ni, and Kao (2021) combined objective facts and subjective views for an evidence-based rumor detection.

Adversarial Training

Adversarial training is an important method to enhance the robustness of neural networks. Szegedy et al. (2014) first proposed the theory of adversarial training by adding small generated perturbations on input images. The perturbed image pixels were later named as adversarial examples. Goodfellow, Shlens, and Szegedy (2015) proposed a fast adversarial example generation approach to try to obtain the perturbation value that maximizes adversarial loss. Jia and Liang (2017) were the first to adopt adversarial example generation for natural language processing tasks. Jia and Liang (2017) were the first to adopt adversarial example generation for natural language processing tasks. Zhao, Dua, and Singh (2018) found that when adopting the gradient-based adversarial training method on natural language processing tasks, the generated adversarial examples were invalid characters or word sequences. Gong et al. (2018) utilized word vectors as the input for deep learning models, but this also generated words that could not be matched with any words in the word embedding space.

Problem Definition

Original rumors on social media are generally composed of a limited number of words and a few emojis. However, limited text information alone cannot accurately predict rumors. We, therefore, treat the original post and its reply posts together as an event for rumor detection. A whole event as the final decision-making unit contains a wealth of internal logic and user stance information. And the proposed hierarchical structure model starts with word embedding, forms post-embedding, event embedding, and finally predicts whether the event is a rumor through a fully connected layer.

Multiple events in the dataset are defined as $D = \{E_1, E_2, ..., E_{|e|}\}$. An event consists of a source post and several reply posts, $E_j = \{P_s, P_1, P_2, ..., P_{|p|}\}$. It should be noted that different events are composed of different numbers of posts, and a post is composed of different words, meaning our model needs to be able to process variable-length sequence information with a hierarchical structure. We consider the rumor detection task a binary classification problem. The event-level classifier can perform learning via labeled event data, that is, $E_j = \{P_s, P_1, P_2, ..., P_{|p|}\} \rightarrow y_j$. In addition, because an event contains multiple posts, we make the posts within the same event share labels. The post-level classifier $P_n = \{x_1, x_2, ..., x_{|x|}\} \rightarrow y_n$ can therefore be established, and all the data will be predicted as the two labels: rumor or non-rumor.

The Proposed Model HAT-RD

Rumors in social media have a hierarchical structure of post-level and event-level. Figure 2 shows a real-world rumor on Twitter. In response to this special data structure, we built the HAT-RD model based on the hierarchical BiLSTM, which can be divided into post-level modules and event-level modules, as shown in Figure 3. Hierarchical Adversarial Training (HAT) is a novel adversarial training method based on the hierarchical structure model. Taking the text of all posts under the event as input, we calculated the embedding of each word to obtain the input of post-level BiLSTM first. The formula is as follows:

$$I_p = \{x_1, x_2, ..., x_n\} \tag{1}$$

where I_p is the input of post-level BiLSTM, and all the vectors with the posts as the unit pass through the post-level BiLSTM layer in proper order. For each time point t, the formula is as follows:

$$h_t^p = \mathrm{BiLSTM_p}(x_i, h_{t-1}^p) \tag{2}$$

The cell state h_t^p of the uppermost LSTMp at the last time point is used as the result of the post encoding. Due to the use of the bidirectional structure, the final state of both directions is joint, and an event can be represented by a matrix in which each column is a vector representing a post. The formula is as follows:

$$O_p = [h_s^p, h_1^p, h_2^p, ..., h_{|p|}^p] \tag{3}$$

where h_s^p is the result of the post-level BiLSTM, that is, the embedding of the source post. h_i^p is the embedding of a reply

Figure 2: A real-world rumor on Twitter. The entire rumor is an event at the event level, an original post and 6 reply posts at the post level.

post, O_p the output of post-level BiLSTM, and I_e the input of event-level BiLSTM. The formula is as follows:

$$I_e = O_p = [h_s^p, h_1^p, h_2^p, ..., h_{|p|}^p] \tag{4}$$

For the next module, the event-level BiLSTM encoding process is similar to post-level BiLSTM. The difference can be seen in the input data unit; post-level BiLSTM uses a post vector composed of word vectors, while event-level BiLSTM uses an event vector composed of post vectors. The formula is as follows:

$$h_t^e = \mathrm{BiLSTM_e}(h_t^p, h_{t-1}^e) \tag{5}$$

In the rumor detection classification task, the state h_t^e of the event-level BiLSTM, the last layer at the last time point can be understood as a comprehensive representation of all posts.

Based on the principle of multi-task learning, rumor post classification and rumor event classification are highly related, and the parameters of the post-level module are shared in the two tasks. A post-level auxiliary classifier and an event-level primary classifier were therefore included in the hierarchical model. The post-level auxiliary classifier is mainly for accelerating training and preventing "vanishing gradient". Two classifiers were used to obtain post-level prediction results and event-level prediction results. The formula is as follows:

$$\hat{y}_p = \mathrm{softmax}(W_p \cdot h_t^p + b_p) \tag{6}$$

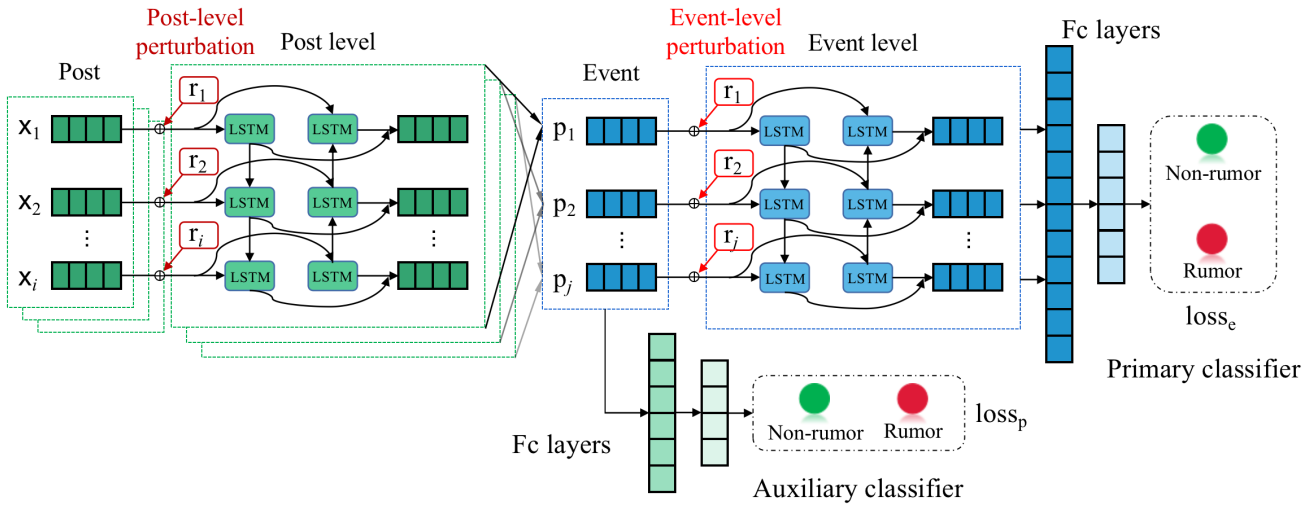


Figure 3: The robust of rumor detection models. * is a well trained BERT on PHEME 17, and this rumor example also comes from the real dataset PHEME 17.

$$\hat{y}_e = \text{softmax}(W_e \cdot h_t^e + b_e) \tag{7}$$

where $\hat{y_p}$ and $\hat{y_e}$ are the post and event classification results, respectively, W_p and W_e are the weights of the fully connected layers, and b_p and b_e are the biases. The goal of each training process is to minimize the standard deviation between the predicted and output values using the following loss function:

$$L_p = -ylog(\hat{y}_{p_r} - (1-y_p)log(1-\hat{y}_{p_n})) \tag{8}$$

$$L_e = -ylog(\hat{y}_{e_r} - (1-y_e)log(1-\hat{y}_{e_n})) \tag{9}$$

$$L_t = \alpha L_p + (1-\alpha)L_e \tag{10}$$

where L_p and L_e are the post-level loss and event-level loss, respectively. α is the loss coefficient weight to control L_p and L_e. L_t is the total loss of the entire rumor detection model used to update the parameters. y is the real label; $\hat{y}_r$ and $\hat{y}_n$ are the two labels predicted by the model -rumor and non-rumor. The gradient of the model was calculated according to Loss L_{total}. The formula is as follows:

$$g = \nabla_\theta L_t(\theta, x, y) \tag{11}$$

Hierarchical Adversarial Training

The above is a forward propagation under standard training of the model. We needed to make the model perform hierarchical adversarial training. The overall hierarchical adversarial training procedure is shown in Algorithm 1. This adversarial optimization process was expressed with the following Min-Max formula:

$$\min_\theta \mathbb{E}_{(x,y)\sim D}\{\max_{r_p, r_e \in S}[L_t(\theta, x_p + r_p, (y_p, y_e)) + L_e(\theta, x_e + r_e, y_e)]\} \tag{12}$$

where r_p and r_e are the perturbations of the post-level input x_p and event-level input x_e under maximization of the internal risk. We respectively estimated these values by linearizing $\nabla_{x_p} L_t(\theta, x_p, (y_p, y_e))$ and $\nabla_{x_e} L_e(\theta, x_e, y_e)$ around x_p and x_e. Using the linear approximation $\nabla_{x_p} L_t(\theta, x_p, (y_p, y_e))$ and $\nabla_{x_e} L_e(\theta, x_e, y_e)$ in equation (13), (14) and the L2 norm constraint, the resulting adversarial perturbations are:

$$r_p = \epsilon_p \cdot \frac{\nabla_{x_p} L_t(\theta, x_p, (y_p, y_e))}{||\nabla_{x_p} L_t(\theta, x_p, (y_p, y_e))||_2} \tag{13}$$

$$r_e = \epsilon_e \cdot \frac{\nabla_{x_e} L_e(\theta, x_e, y_e)}{||\nabla_{x_e} L_e(\theta, x_e, y_e)||_2} \tag{14}$$

where ϵ_p and ϵ_e are the perturbation coefficients. Note that the value of the perturbation r_p is calculated based on the back-propagation of the total Loss instead L_t of L_p, because the addition of the perturbation r_p makes L_p and Le increase at the same time.

Post-Level Adversarial Training After a normal forward and backward propagation, and were calculated according to the gradient. Using post-level adversarial training, we added word-level perturbation to the word vector to obtain the input of post-level BiLSTM, and the formula is as follows:

$$I_{p_{adv}} = \{x_1 + r_1^p, x_2 + r_2^p, ..., x_n + r_n^p\} \tag{15}$$

where $I_{p_{adv}}$ is the adversarial input of post-level BiLSTM, and r_n^p is the post-level perturbation added to the word vector x_n. All the vectors with the posts as the unit then pass through the post-level BiLSTM layer in proper order. For each time point t, the formula is as follows:

$$h_t^{p_{adv}} = \text{BiLSTM}_\text{p}(x_i + r_i^p, h_{t-1}^{p_{adv}}) \tag{16}$$

The adversarial cell state $h_t^{p_{adv}}$ of the uppermost LSTM_p at the last time point is used as the result of the post encoding. Due to the use of the bidirectional structure, the final state of both directions is joint, and an event can be represented by a matrix in which each column is a vector representing a

Algorithm 1: Hierarchical adversarial training algorithm

Input: Training samples $\mathcal{X}$, perturbation coefficient ϵ_p and ϵ_e, Loss coefficient weight α, Learning rate τ
Parameter: θ

for epoch $= 1 \dots N_{ep}$ **do**
  for $(x, y) \in \mathcal{X}$ **do**
    Forward-propagation calculation Loss:
      $L_p \leftarrow -ylog(\hat{y}_{p_r} - (1 - y_p)log(1 - \hat{y}_{p_n}))$
      $L_e \leftarrow -ylog(\hat{y}_{e_r} - (1 - y_e)log(1 - \hat{y}_{e_n}))$
      $L_t \leftarrow \alpha L_p + (1 - \alpha)L_e$
    Backward-propagation calculation gradient:
      $g_p \leftarrow \nabla_{x_p} L_t(\theta, x_p, (y_p, y_e))$
      $g_e \leftarrow \nabla_{x_e} L_e(\theta, x_e, y_e)$
    Compute perturbation:
      $r_p \leftarrow \epsilon_p \cdot g_p / ||g_p||_2$
      $r_e \leftarrow \epsilon_e \cdot g_e / ||g_e||_2$
    Forward-Backward-propagation calculation adversarial gradient:
      $g^p_{adv} \leftarrow \nabla_\theta L^p_{t_{adv}}(\theta, x_p + r_p, (y_p, y_e))$
      $g^e_{adv} \leftarrow \nabla_\theta L^e_{e_{adv}}(\theta, x_e + r_e, y_e)$
    Update parameter:
      $\theta \leftarrow \theta - \tau(g + g^p_{adv} + g^e_{adv})$
  end for
end for
Output: θ

post. The formula is as follows: time point t, the formula is as follows:

$$O_{p_{adv}} = [h_s^{p_{adv}}, h_1^{p_{adv}}, h_2^{p_{adv}}, ..., h_{|p|}^{p_{adv}}] \quad (17)$$

where $h_s^{p_{adv}}$ is the adversarial result of the post-level BiLSTM, that is, the embedding of the source post. $h_i^{p_{adv}}$ is the adversarial embedding of the reply post, and $O_{p_{adv}}$ is the adversarial output of post-level BiLSTM and input of event-level BiLSTM. The formula is as follows:

$$h_t^{e_{adv}} = \mathrm{BiLSTM_p}(h_t^{p_{adv}} + r_t^e, h_{t-1}^{e_{adv}}) \quad (18)$$

Finally, h_t^e was replaced with $h_t^{e_{adv}}$ and the adversarial loss $L^p_{p_{adv}}$, $L^p_{e_{adv}}$ and $L^p_{t_{adv}}$ of post-level perturbation can be calculated using equations (6)-(9). The post-level adversarial gradient g^p_{adv} is calculated based on the result of backpropagation. The formula is as follows:

$$g^p_{adv} = \nabla_\theta L^p_{t_{adv}}(\theta, x_p + r_p, (y_p, y_e)) \quad (19)$$

Event-Level Adversarial Training We next performed event-level adversarial training and repeated the process of equations (1), (2) and (3) to obtain the posts vector. Event-level perturbation was then added to the post vector to obtain the adversarial input of event-level BiLSTM, and the formula is as follows:

$$I_{e_{adv}} = \{h_s^p + r_s^p, h_1^p + r_1^p, h_2^p + r_2^p, ..., h_{|p|}^p + r_{|p|}^p\} \quad (20)$$

In the same way, input $I_{e_{adv}}$ into the event-level BiLSTM to get the final event representation vector $h_t^{e_{adv}}$, replace h_t^e with $h_t^{e_{adv}}$ and calculate the adversarial loss $L^e_{e_{adv}}$ of event-level perturbation through equations (6)-(9). Finally, the post-level adversarial gradient g^e_{adv} is calculated based on backpropagation. The formula is as follows:

$$g^e_{adv} = \nabla_\theta L^e_{e_{adv}}(\theta, x_e + r_e, y_e) \quad (21)$$

Finally, the gradient is calculated by the standard training; the gradient calculated by the post-level adversarial training and the gradient calculated by the event-level adversarial training were used to update the model parameters. The parameter update process is expressed as:

$$\theta \leftarrow \theta - \tau(g + g^p_{adv} + g^e_{adv}) \quad (22)$$

where τ is the learning rate.

Statistic	PHEME 2017	PHEME 2018
Users	49,345	50.593
Posts	103,212	105,354
Events	5,802	6,425
Avg words/post	13.6	13.6
Avg posts/event	17.8	16.3
Max posts/event	346	246
Rumor	1972	2402
Nonrumor	3830	4023
Balance degree	34.00%	37.40%

Table 1: Dataset statistics.

Experiments

Datasets

Two extensive public rumor datasets, PHEME 2017 and PHEME 2018 (Kochkina, Liakata, and Zubiaga 2018), are used to evaluate our method HAT-RD. Each dataset contains a series of topics, and each topic is divided into several events. Each event is composed of a source post and several reply posts, as shown in Table 1.

Experimental Settings

The datasets were randomly split for our experiment: 80% for training, 10% for validation, and 10% for testing. Similar to the work of (Li, Ni, and Kao 2021), we calculated the accuracy, precision, recall and F1-score to measure the rumor detection performance. In the data preprocessing phase, our data were subjected to the following processes: text standardization, deletion of useless network labels, and so on. Stop words were retained because they contained words that could reflect the writer's emotions. We trained all the models by employing the derivative of the loss function through backpropagation and used the Adam optimizer (Kingma and Ba 2014) to update the parameters. We use Glove's (Pennington, Socher, and Manning 2014) pre-trained 300-dim word vector. For the hyperparameters, the maximum value of vocabulary is 80000; the batch size is 64, the dropout rate is 0.5, the BiLSTM hidden size unit is 512, the loss coefficient weight α is 0.1, the learning rate is 0.0001, and the perturbation coefficient ϵ_p and ϵ_e are 1.0 and 0.3. Our proposed model was finally trained for 100 epochs with early

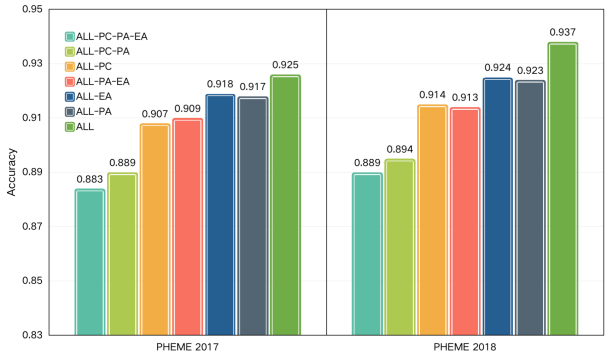


Figure 4: HAT-RD ablation analysis in accuracy.

stopping. In addition, all experiments are run under the following hardware environment: CPU: Intel(R) Core(TM) i7-8700 CPU@3.20GHz, GPU: GeForce RTX 2080, 10G.

Performance Comparison

Our HAT-RD model was compared with other well-known rumor detection models to evaluate our model's rumor debunking performance. Note that some of them are the current state-of-the-art models:

- SVM-BOW: a rumor detection naive baseline, predicting rumors by an SVM classifier that using bag-of-words for word representation (Ma, Gao, and Wong 2018).
- CNN: a rumor detection naive baseline based on convolutional neural networks that represent each suspicious statement and classify rumors with a fully connected layer (Chen, Liu, and Kao 2017).
- BiLSTM: a bidirectional RNN-based model that detects rumor by considering the bidirectional suspicious statement words (Augenstein et al. 2016).
- BERT: a well-known model in many NLP tasks. We fine-turned a BERT model to detect rumors (Devlin et al. 2019).
- RDM: a rumor detection model that integrates reinforcement learning and deep learning for early rumor detection (Zhou et al. 2019).
- CSRD: a rumor detection model that classifies rumors by simulating comments' conversation structure using GraphSAGE and BiLSTM (Li, Sujana, and Kao 2020).
- LOSIRD: a state-of-the-art rumor detection model that leverages objective facts and subjective views for interpretable rumor detection (Li, Ni, and Kao 2021).
- HAT-RD: Our proposed model which uses post-level and event-level hierarchical adversarial training to enhance the model.

Main Results and Thought

The results of different rumor detection models are compared in Table 5; the HAT-RD clearly performs the best in terms of rumor detection compared to the other methods based on the two datasets with 92.5% accuracy on PHEME 2017 dataset and 93.7% on PHEME 2018. In addition, the

Dataset	Method	Acc	Pre	Rec	F1
PHEME 2017	SVM-BOW	0.669	0.535	0.524	0.519
	CNN	0.787	0.737	0.702	0.71
	BiLSTM	0.795	0.763	0.691	0.706
	BERT	0.865	0.859	0.851	0.855
	RDM	0.873	0.817	0.823	0.82
	CSRD	0.900	0.893	0.869	0.881
	LOSIRD*	0.914	0.915	0.900	0.906
	HAT-RD⊛	**0.925**	**0.925**	**0.911**	**0.917**
PHEME 2018	SVM-BOW	0.688	0.518	0.512	0.504
	CNN	0.794	0.731	0.673	0.686
	BiLSTM	0.796	0.727	0.677	0.689
	BERT	0.844	0.834	0.835	0.835
	RDM	0.858	0.847	0.859	0.852
	CSRD	0.919	0.892	0.923	0.907
	LOSIRD*	0.925	0.922	0.924	0.923
	HAT-RD⊛	**0.937**	**0.932**	**0.936**	**0.934**

Table 2: The results of different methods on two datasets. The reported results are calculated from 5 runs with the same hyper-parameters except for the random seeds. *: the state-of-the-art model. ⊛: our model.

precision, recall, and F1 are all higher than 91% in the HAT-RD model. These results demonstrate the effectiveness of the hierarchical structure model and hierarchical adversarial training in rumor detection. However, the SVM-BOW result is poor because the traditional statistical machine learning method could not handle this complicated task. The results of the CNN, BiLSTM, Bert and RDM models are poorer than ours due to their insufficient information extraction capabilities. The models are based on post-processing information and cannot get a high-level representation from the hierarchy. Compared to other models, our HAT-RD model makes full use of both prior-knowledge information and current comments information to obtain more powerful representations for debunking rumors. Our model has a hierarchical structure and performs different levels of adversarial training. This enhances both post-level and event-level sample space and improves the robustness and generalization of the rumor detection model.

Ablation Analysis

To evaluate the effectiveness of each component of the proposed HAT-RD, we removed each one from the entire model for comparison. "ALL" denotes the entire model HAT-RD with all components, including post-level adversarial training (PA), event-level adversarial training (EA), the post-level auxiliary classifier (PC), and event-level primary classifier (EC). After removing each one of them, we obtained the sub-models "-PA", "-EA", "-PC" and "-EC", respectively. "-PA-PC" means that both the post-level adversarial training and auxiliary classifier were removed. "-PA-EA" denotes the reduced HAT-RD without both post-level adversarial training and event-level adversarial training. The results are shown in Figure 4.

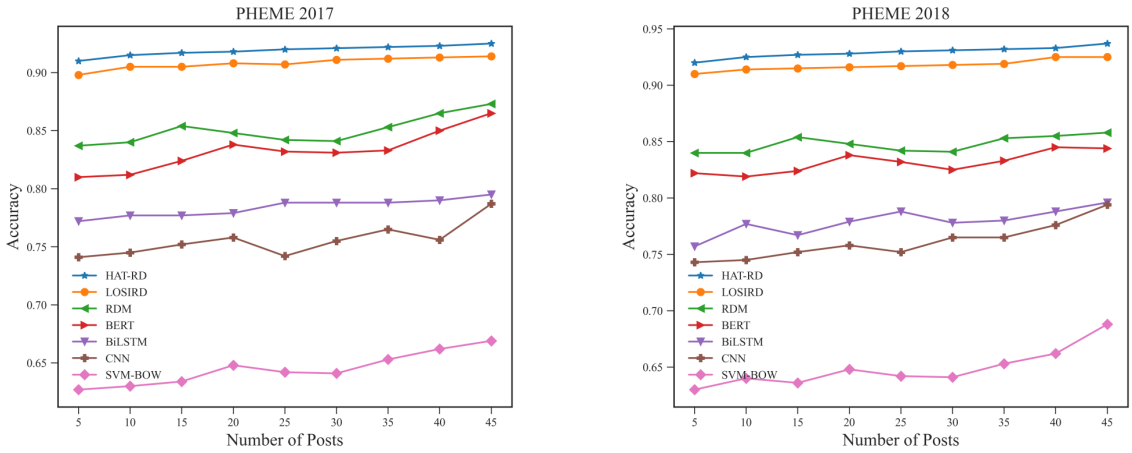


Figure 5: Early rumor detection accuracy at different numbers of posts.

Original rumors	Rumors after refactoring	Rumor prediction confidence	
		Non-HAT	HAT-RD
Charlie Hebdo journalist Wandrille Lanos tells France 2 that men with kalashnikovs entered the office, calls it a "scene of carnage."	It a scene of carnage that men with kalashnikovs entered the office, according to media reports.	$0.73 \rightarrow 0.34$	$0.77 \rightarrow 0.63$
Reports police have now identified the hostage taker at #SydneySiege. Five hostages have escaped, 15 people are believed still being held.	The hostage taker now is identified at #SydneySiege. Five hostages escaped and 15 are still being held.	$0.87 \rightarrow 0.61$	$0.84 \rightarrow 0.74$
Hostage situation in Sydney is happening next door to @7NewsSydney's studio – this is a live stream of the coverage	BREAKING: hostage situation in sydney is happening next door to @7NewsSydney's studio.	$0.75 \rightarrow 0.35$	$0.74 \rightarrow 0.42$
Ferguson Police can afford machine guns but they can't afford dash cameras for their squad cars? interesting...	Ferguson Police can afford machine guns but they can't afford dash cameras for their squad cars? interesting...	$0.67 \rightarrow 0.44$	$0.71 \rightarrow 0.65$
MSNBC is literally reporting that Police Chief CLEARLY said it was about jay walking. This is a fucking clown show. #Ferguson	Police Chief CLEARLY said it was about jay walking. "Because he was walking down the middle of the street blocking traffic. That was it." This is a fucking clown show. #Ferguson	$0.72 \rightarrow 0.39$	$0.75 \rightarrow 0.53$

Table 3: The impact of hierarchical adversarial training on model robustness. Non-HAT means the model HAT-RD that removes the adversarial training method.

It can be observed that every component plays a significant role in improving the performance of HAT-RD. HAT-RD outperforms ALL-PA and ALL-EA, which shows that the post-level adversarial training and event-level adversarial training are indeed helpful in rumor detection. Both ALL-PA and ALL-EA are better than ALL-PA-EA, which shows that hierarchical adversarial training is more efficient than single-level adversarial training. The performance of ALL-PC is lower than that of HAT-RD, proving that the post-level auxiliary classifier contributes to the learning and convergence of the model.

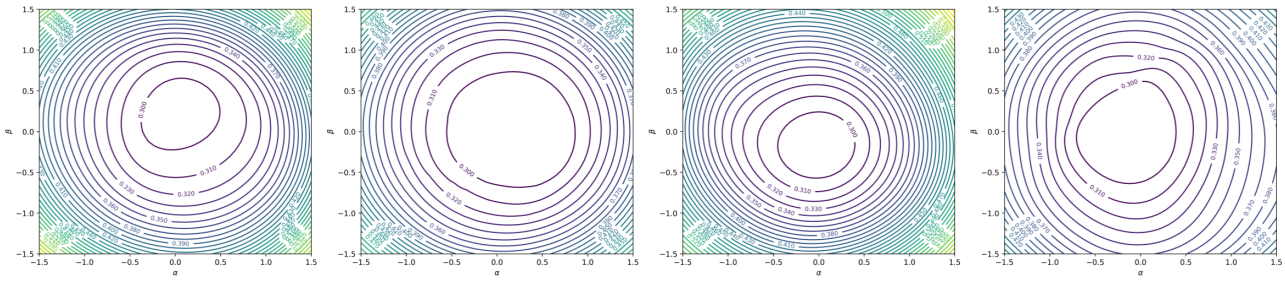
Early Rumor Detection

Our model's performance in early rumor detection was evaluated. To simulate the early stage rumor detection scenarios in the real world, 9 different size test sets from PHEME 2017 and PHEME 2018 were created. Each test set contained a certain number of posts, ranging from 5 to 45. We found that the HAT-RD model could detect rumors with an approximate 91% accuracy rate with only 5 posts as illustrated in Figure 5.

Additionally, the broken lines showed that the HAT-RD model's early rumor detection performance was significantly stable. Compared to the other models, our model uses hierarchical adversarial training and continuously generates optimal adversarial samples to join the training. It, therefore, has good generalization despite limited information.

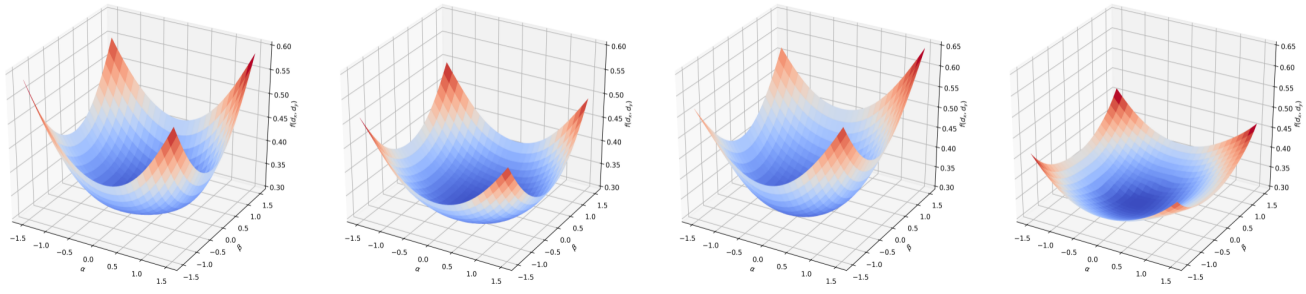
Case Study

To evaluate the robustness of our model against reconstructed and disguised rumors, we randomly selected five rumor texts from the data set and manually reconstructed them



(a) (a*) (b) (b*)

Figure 6: 2D visualization of the minima of the Loss function selected by standard training (a, b) and hierarchical adversarial training (a*, b*) on PHEME 17 (a, a*) and PHEME 18 (b, b*) dataset.



(a) (a*) (b) (b*)

Figure 7: 3D visualization of the minima of the Loss function selected by standard training (a, b) and hierarchical adversarial training (a*, b*) on PHEME 17 (a, a*) and PHEME 18 (b, b*) dataset.

while keeping the main meaning and labels of the rumors unchanged. The experimental results are shown in Table 3. The model "Non-HAT" has obvious changes in its prediction confidence for reconstructed rumors, while the model "HAT-RD" has significantly more robust prediction confidence and is generally not deceived by disguised rumors. The experimental results show that hierarchical adversarial training can enhance the model's ability to resist changes in the spread of rumors so they cannot escape detection.

The Impact of Hierarchical Adversarial Training on Loss Landscape

To further visually analyze the effectiveness of the hierarchical adversarial training method, we drew the high-dimensional non-convex loss function with a visualization method proposed by (Li et al. 2018). We visualize the loss landscapes around the minima of the empirical risk selected by standard and hierarchical adversarial training with the same model structure. The 2D views are plotted in Figure 6 and the 3D views in Figure 7. We defined two direction vectors, d_x and d_y with the same dimensions as θ, drawn from a Gaussian distribution with zero mean and a scale of the same order of magnitude as the variance of layer weights. We then chose a center point θ^* and added a linear combination of α and β to obtain a loss that is a function of the contribution of the two random direction vectors.

$$f(d_x, d_y) = L(\theta^* + \alpha d_x + \beta d_y) \tag{23}$$

The results show that the hierarchical adversarial training method indeed selects flatter loss landscapes by dynamically generating post-level perturbation and event-level perturbation. Having a flatter Loss function indicates that the model is more robust in input features and can prevent the model from overfitting.

Conclusion and Future Work

Herein, we proposed a new hierarchical rumor detection model that considers the camouflages and variability of rumors from an adversarial perspective. Dynamically generating perturbations on the post-level and event-level embedding vectors enhanced the model's robustness and prevented deception by disguised and reconstructed rumors. Moreover, Visual experiments prove that the hierarchical adversarial training method we proposed can optimize the model for a flatter loss landscape. The evaluations of two real-world rumor datasets show that our model can outperform state-of-the-art baselines.

Robustness and generalization are the focus of rumor detection. In the future, we can integrate features such as text and images for multi-modal adversarial training to further enhance the model.

References

Ajao, O.; Bhowmik, D.; and Zargari, S. 2018. Fake news identification on twitter with hybrid cnn and rnn models. In *Proceedings of the 9th international conference on social media and society*, 226–230.

Augenstein, I.; Rocktäschel, T.; Vlachos, A.; and Bontcheva, K. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 876–885. ACL.

Chen, T.; Li, X.; Yin, H.; and Zhang, J. 2018. Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In *Pacific-Asia conference on knowledge discovery and data mining*, 40–52. Springer.

Chen, Y.-C.; Liu, Z.-Y.; and Kao, H.-Y. 2017. Ikm at semeval-2017 task 8: Convolutional neural networks for stance detection and rumor verification. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, 465–469.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186.

Gong, Z.; Wang, W.; Li, B.; Song, D.; and Ku, W.-S. 2018. Adversarial texts with gradient methods. *arXiv preprint arXiv:1801.07175*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *3nd International Conference on Learning Representations, ICLR 2015*.

Guo, H.; Cao, J.; Zhang, Y.; Guo, J.; and Li, J. 2018. Rumor detection with hierarchical social attention network. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 943–951.

Jia, R.; and Liang, P. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2021–2031.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kochkina, E.; Liakata, M.; and Zubiaga, A. 2018. Pheme dataset for rumour detection and veracity classification. *figshare, Jun*.

Li, H.; Xu, Z.; Taylor, G.; Studer, C.; and Goldstein, T. 2018. Visualizing the loss landscape of neural nets. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 6391–6401.

Li, J.; Ni, S.; and Kao, H.-Y. 2020. Birds of a Feather Rumor Together? Exploring Homogeneity and Conversation Structure in Social Media for Rumor Detection. *IEEE Access*, 8: 212865–212875.

Li, J.; Ni, S.; and Kao, H.-Y. 2021. Meet The Truth: Leverage Objective Facts and Subjective Views for Interpretable Rumor Detection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Li, J.; Sujana, Y.; and Kao, H.-Y. 2020. Exploiting microblog conversation structures to detect rumors. In *Proceedings of the 28th International Conference on Computational Linguistics*, 5420–5429.

MA, J.; GAO, W.; MITRA, P.; KWON, S.; JANSEN, B. J.; WONG, K. F.; and CHA, M. 2016. Detecting rumors from microblogs with recurrent neural networks.(2016). In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, 3818–3824.

Ma, J.; Gao, W.; and Wong, K.-F. 2018. Rumor detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics.

Ni, S.; Li, J.; and Kao, H.-Y. 2021. MVAN: Multi-View Attention Networks for Fake News Detection on Social Media. *IEEE Access*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Ruchansky, N.; Seo, S.; and Liu, Y. 2017. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 797–806.

Shu, K.; Cui, L.; Wang, S.; Lee, D.; and Liu, H. 2019. defend: Explainable fake news detection. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 395–405.

Sujana, Y.; Li, J.; and Kao, H.-Y. 2020. Rumor Detection on Twitter Using Multiloss Hierarchical BiLSTM with an Attenuation Factor. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 18–26.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*.

Tasnim, S.; Hossain, M. M.; and Mazumder, H. 2020. Impact of rumors and misinformation on COVID-19 in social media. *Journal of preventive medicine and public health*, 53(3): 171–174.

Yang, Y.; Zheng, L.; Zhang, J.; Cui, Q.; Li, Z.; and Yu, P. S. 2018. TI-CNN: Convolutional neural networks for fake news detection. *arXiv preprint arXiv:1806.00749*.

Yu, F.; Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2019. Attention-based convolutional approach for misinformation identification from massive and noisy microblog posts. *Computers & Security*, 83: 106–121.

Zhao, Z.; Dua, D.; and Singh, S. 2018. Generating Natural Adversarial Examples. In *International Conference on Learning Representations*.

Zhou, K.; Shu, C.; Li, B.; and Lau, J. H. 2019. Early rumour detection. 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)*, 1614–1623.