Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems

School of Information Systems

5-2019

Detect rumors on Twitter by promoting information campaigns with generative adversarial learning

Jing MA

Wei GAO
Singapore Management University, weigao@smu.edu.sg

Kam-Fai WONG

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

 Part of the Databases and Information Systems Commons

Citation

MA, Jing; GAO, Wei; and WONG, Kam-Fai. Detect rumors on Twitter by promoting information campaigns with generative adversarial learning. (2019). *Proceedings of the Web Conference (WWW 2019)*. 3049-3055. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/4559

This Conference Proceeding Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email libIR@smu.edu.sg.

Detect Rumors on Twitter by Promoting Information Campaigns with Generative Adversarial Learning

Jing Ma
The Chinese University of Hong Kong
Hong Kong SAR
majing@se.cuhk.edu.hk

Wei Gao
Victoria University of Wellington
New Zealand
wei.gao@vuw.ac.nz

Kam-Fai Wong
The Chinese University of Hong Kong
& MoE Key Lab of High Confidence
Software Technologies, CUHK

ABSTRACT

Rumors can cause devastating consequences to individual and/or society. Analysis shows that widespread of rumors typically results from deliberately promoted information campaigns which aim to shape collective opinions on the concerned news events. In this paper, we attempt to fight such chaos with itself to make automatic rumor detection more robust and effective. Our idea is inspired by adversarial learning method originated from Generative Adversarial Networks (GAN). We propose a GAN-style approach, where a generator is designed to produce uncertain or conflicting voices, complicating the original conversational threads in order to pressurize the discriminator to learn stronger rumor indicative representations from the augmented, more challenging examples. Different from traditional data-driven approach to rumor detection, our method can capture low-frequency but stronger non-trivial patterns via such adversarial training. Extensive experiments on two Twitter benchmark datasets demonstrate that our rumor detection method achieves much better results than state-of-the-art methods.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; **Natural language processing**.

KEYWORDS

Information Campaigns; Rumor Detection; GAN

ACM Reference Format:
Jing Ma, Wei Gao, and Kam-Fai Wong. 2019. Detect Rumors on Twitter by Promoting Information Campaigns with Generative Adversarial Learning. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313741

1 INTRODUCTION

The proliferation of rumors is a rampant phenomenon on social media. Rumor producers can cause devastating effects by manipulating public events. Information campaigns are frequently carried out by rumor makers via social networks to promote controversial memes, fake news, etc. with high volume of misinformation that competes with genuine ones for dragging people's attention to bogus claims. For example, during the US 2016 presidential election, Russia reportedly had coordinated thousands of "social bots" like covert human agents and automated programs to spread false information by corroborating each other[1]. The widespread of rumors has triggered huge debates and has already unprecedentedly influenced US politics.

Social psychology literature defines a rumor as a story or a statement whose truth value is unverified or deliberately false [1]. Fact-checking websites such as snopes.com and politifact.com rely on manual effort to track and debunk rumors, which have obvious limitation on efficiency and coverage. Existing automated methods typically resort to supervised classifiers trained on a wide range of hand-crafted features based on message contents, user profiles and diffusion patterns [3, 8–10, 13]. To avoid feature engineering effort, data-driven models were exploited more recently and demonstrated state-of-the-art detection performances. For example, Ma et al. [12] employed recurrent neural networks (RNNs), specifically LSTM and GRU, to learn hidden features from text content of relevant posts regarding given claims for detecting rumors. Yu et al. [20] used convolutional neural networks (CNNs) for obtaining local-global features from the relevant posts.

Nevertheless, existing data-driven approaches typically rely on finding indicative responses such as skeptical and disagreeing opinions for detection. Rumor producers can take advantage of promoted campaigns to entangle public opinions or influence collective stances to get it widely disseminated and amplified. This poses a major technical challenge to data-driven methods as text patterns and other explicit features become hardly discriminative. Various conflicting and uncertain voices that co-exist can seriously disturb the learning (or extraction) of useful features. Figure 1 illustrates a case of promoted campaign for the rumor about "Saudi Arabia beheads first female robot citizen", which shows how the popular indicative patterns expressing skepticism and disagreement such as "fake news", "not sure", "no truth" are inundated by the promoted posts. Therefore, developing a more robust feature learner for rumor detection is urgently desirable.

In this paper, we propose a radically new rumor detection method by leveraging the mechanism of information campaign and promoting it in a controlled manner in order to achieve more robust and effective detection. Our seemingly counter-intuitive idea is inspired by the Generative Adversarial Networks or dubbed as GAN [6, 7], where a discriminative classifier learns to distinguish whether an instance is from real world, and a generative model is trained to confuse the discriminator by generating proximately realistic examples. The harder are the generated examples to be distinguished from real-world ones, the stronger is the discriminator that can be

[1] http://time.com/4783932/inside-russia-social-media-war-america/

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.
WWW '19, May 13–17, 2019, San Francisco, CA, USA
© 2019 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.
ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313741

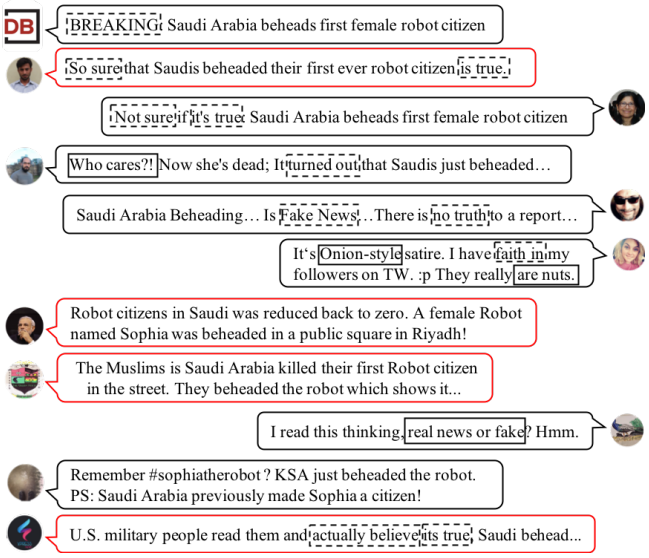


Figure 1: Sample responses to a rumor claim about "Saudi Arabia beheads its first female robot citizen" in a promoted campaign. Social bots activities are marked as red box. The supportive responses are listed at left side and denial (or uncertain) responses at right side. Patterns captured by existing methods are marked by dashed rectangles, and patterns that can be missing are marked by solid rectangles.

learned. We train our generator to output challenging examples by mimicking campaign promotions including misled grassroots conversations with uncertain and conflicting voices, so as to push our discriminator to strengthen feature learning from such difficult examples for capturing more discriminative patterns. Unlike typical GAN-style models such as in computer vision [6], which aim to learn a strong (image) generator, our goal however is to force our (rumor) *discriminator* to be more discriminative.

Intuitively, why can such a GAN-style method do better in feature learning? As shown in Figure 1, various users engagements can easily break the past data-driven methods that typically resort to repetitive patterns in responding posts. Due to the effect of generated campaign, high-frequency patterns such as "fake news" or "no truth" commonly occurring in the responses of rumors and "be true/sure" in those of non-rumors become less discriminative. As a result, the discriminator is adjusted adaptively to focus on capturing the relatively low-frequency patterns, such as "onion-style" and "be nuts", which are expected non-trivial as high-frequency ones while used to be ignored in existing feature learning methods. To retain the discriminative power of the original high-frequency patterns, we train the discriminator on an augmented training data with both generated campaign-like examples and the original examples.

The main contributions of our paper are four-fold:

- To the best of our knowledge, this is the first generative approach for rumor detection using a text-based GAN-style framework, where we make the text generator and discriminator adversarially strengthen each other for enhancing representation learning of rumor-indicative patterns.
- We model rumor dissemination as generative information campaigns for generating confusing training examples to challenge the discriminator of its detection capacity.
- Under the GAN-style framework, we reinforce our discriminator, which is trained on a set of more challenging examples replenished by the generator, to be focused on learning low-frequency yet discriminative patterns.
- We experimentally demonstrate that our model is more robust and effective than state-of-the-art baselines based on two public benchmark datasets for the tasks of rumor detection on Twitter.

2 PROBLEM STATEMENT

In general, rumor detection task can be defined as a binary classification problem, which aims to learn a classifier from training claims labeled as **rumor** or **non-rumor** for predicting the label of a test claim. A claim is a factual (rather than opinionated) assertion or statement that something is true, such as the example statement "Saudi Arabia beheads its first female robot citizen" in Figure 1.

Typically, a claim is short which contains very limited context. For reliable feature extraction, in Twitter rumor detection task, a claim is commonly represented by a set of posts (i.e., tweets) relevant to the claim which can be collected via Twitter's search function. Specifically, we represent a rumor dataset as $\{X\}$, where each $X = (y,\ x_1 x_2 \dots x_T)$ is a tuple representing a given claim: X consists of the ground-truth label $y \in \{N, R\}$ of the claim (i.e., Non-rumor or Rumor) and a sequence of relevant posts $x_1 x_2 \dots x_T$, where each x_t can represent a post or more generally a batch of posts in a time interval, and is indexed with a time step t. Thus, a claim can be considered as a time sequence of relevant posts. For clarity, we write an instance (claim) X as X_y, that is, X_R denotes a rumor and X_N a non-rumor.

3 GENERATIVE ADVERSARIAL LEARNING FOR RUMOR DETECTION

Information campaigns pose challenges to existing rumor detection models since frequent patterns indicative of veracity become distorted and misleading. Our basic idea is to strengthen representation learning of rumor indicative features, inspired by the mechanism of generative adversarial learning [7]. We propose a GAN-style model where a generator attempts to promote campaigns by generating hard examples and a discriminator aims to identify robust features to overcome the difficulty posed by the generator. Unlike the recent event adversarial model [18] for multi-modal fake news detection and the neural user response generator [16] for early detection, our idea and the adopted mechanisms are significantly different.

3.1 Controversial Example Generation

Our generative model aims to produce uncertain or conflicting voices regarding a given claim, making the differentiation of rumor and non-rumor harder which used to rely on repetitive patterns.

A straightforward way is to twist or complicate the opinions expressed in the original data examples via a handful of rule templates. For instance, we can 1) incorporate enquiry expressions such as "really?", "is it true?", "not sure", etc. into responding posts; 2) negate its stance of a post by adding a "not" after a "be" verb; and/or 3) apply antonym replacement at certain parts of the keywords, e.g.,

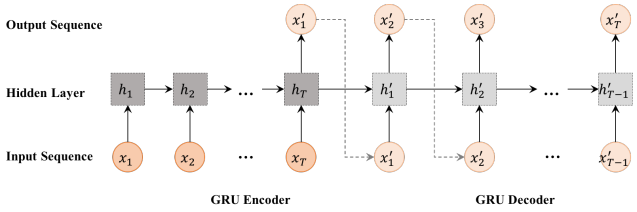


Figure 2: Framework of our generator with a neural sequence-to-sequence model. x'_t is the transformation of x_t, assuming that the length of output sequence equals to that of the input.

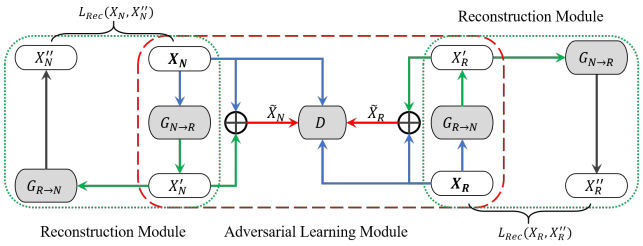


Figure 3: Overview of our GAN-style rumor detection model. $G_{N\to R}$ (or $G_{R\to N}$) is a generator. D denotes the discriminator.

by replacing "fake" with "true", "right" with "wrong", etc. However, it is difficult to generalize these rules for formally producing any kind of controversial voices. A general approach would be to cast our generator as trainable model that can cover a wide range of variations of expressions.

To this end, we design two generators, one for distorting non-rumor to make it look like a rumor, and the other for "whitewashing" rumor so that it looks like a non-rumor: 1) $G_{N\to R}$ generates skeptical or opposing voices against non-rumor claim; and 2) $G_{R\to N}$ generates supportive voices towards rumor claims. We define a function f_g to formulate our generative model:

$$X'_y = f_g(X_y) = \begin{cases} G_{N\to R}(X_y) & \text{if } y = N; \\ G_{R\to N}(X_y) & \text{if } y = R \end{cases} \tag{1}$$

where X_y is an original instance from training set which is either a rumor or non-rumor, and X'_y is the transformed instance with the generator while the label remains intact.

Considering the time sequence structure of posts in each instance, we use a sequence-to-sequence model [11, 17] for the generative transformation, which is illustrated in Figure 2. We encode an input sequence X_y into a hidden vector via an RNN encoder, and then generate the transformed sequence X'_y from it via an RNN decoder.

GRU-RNN Encoder: We batch relevant posts into time intervals and treat each batch as a single unit in the time sequence, following the similar time segmentation described in [12]. Using RNN, we map each input unit $x_t \in X_y$ into a hidden vector h_t, for which we use GRU [4] to store hidden representation:

$$h_t = \text{GRU}(x_t, h_{t-1}; \theta_g) \tag{2}$$

where GRU$(\cdot)$ denote standard GRU transition equations, x_t is the input unit represented as a vector of tf*idf values of vocabulary words computed from the posts falling into the t-th time step, h_{t-1} refers to the previous hidden state, and θ_g represent all the parameters of GRU.

The output of the last time step h_T from the GRU-RNN encoder will be the hidden representation of X_y. Note that the sequence length T is not fixed which can vary with different instances.

GRU-RNN Decoder: We now describe the GRU-RNN decoder that transforms h_T into the generated sequence $X'_y = x'_1 x'_2 \dots x'_T$. Specifically, each unit is sequentially generated using GRU followed by a softmax output function. In each step t, a softmax output layer maps the hidden states h'_t which is obtained via GRU into the target representation x'_{t+1} of a batch of posts by computing a distribution over vocabulary words:

$$\begin{aligned} h'_t &= \text{GRU}(x'_t, h'_{t-1}; \theta'_g) \\ x'_{t+1} &= \text{softmax}(V_g h'_t + b_g) \end{aligned} \tag{3}$$

where h'_{t-1} is the previous hidden state of GRU decoder and θ'_g represents all the parameters inside the GRU[2], V_g and b_g are trainable parameters of the output layer.

3.2 GAN-Style Adversarial Learning Model

In this encoder-decoder framework, $f_g(X_y)$ defines the policy that generates the representation of controversial posts which are used to confuse the discriminator based on the input X_y. A crucial issue is how to control the generator to generate the needed. To this end, we use the performance of discriminator as a reward to guide the generator. Before we present the discriminator, let us introduce the architecture and control mechanism of our GAN-style model as shown in Figure 3, which consists of an adversarial learning module and two reconstruction modules (one for rumor and the other for non-rumor).

Adversarial learning module: In our model, the generators are encouraged to produce campaign-like instances to fool the discriminator so that discriminator can be focused on learning more discriminative features. Such a goal suggests a training objective resembling adversarial learning [7]. We formulate adversarial loss as the *negative* of discriminator loss based on the generator-augmented training data:

$$L_{Adv} = -L_D(\bar{y}, \hat{y}) \tag{4}$$

where $L_D(\cdot)$ is the loss between the ground-truth class probability distribution $\bar{y}$ and the class distribution $\hat{y}$ predicted by discriminator given an input instance (see Eq. 11 for the specific form of L_D).

We combine the generated examples and original ones to augment the training set by taking the union of them, i.e., $\{\{X_y\} \cup \{\tilde{X}_y\}\}$, where $\tilde{X}_y = X_y \oplus X'_y$ is the elementwise addition of the original and generated examples. Note that the elementwise addition has the effect to cancel out influential high-frequency patterns and promote the chance of important less frequent patterns for being selected. Meanwhile, we do not want to seriously weaken those useful features in the original example. Thus, the original example X_y is combined with $\tilde{X}_y$ for training as shown in Figure 3.

[2]Note that each generator ($G_{R\to N}$ or $G_{N\to R}$) corresponds to a unique sequence-to-sequence model. The encoding and decoding GRUs have similar yet different set of parameters: θ_g and θ'_g.

Reconstruction module: It is likely that the generator could distort the original example towards unexpected direction by changing some essential aspects of a story. For example, the theme of "Saudi beheads robot citizen" might get distorted as "Saudi digs canal for Qatar" which is irrelevant and not helpful. To avoid that, we introduce a reconstruction mechanism to make the generative process reversible. The idea is that the opinionated voices will be reversible through two generators of opposite direction so as to minimize the loss of fidelity of information. We define the reconstruction function as follows:

$$X''_y = f_r(X_y) = \begin{cases} G_{R\to N}(G_{N\to R}(X_y)) & \text{if } y = N; \\ G_{N\to R}(G_{R\to N}(X_y)) & \text{if } y = R \end{cases} \tag{5}$$

where X''_y is the reconstructed instance from an original instance X_y via two opposite generators. We formulate the difference between X''_y and X_y as a reconstruction loss:

$$L_{Rec} = \frac{1}{T}\sum_{t=1}^{T} ||x_t - x''_t||_2 \tag{6}$$

where x_t and x''_t are the t-th unit in the original and reconstructed sequences, respectively, T is the sequence length, and $||\cdot||_2$ is the L2-norm of a vector.

Objective of optimization: The overall loss function of our GAN-style adversarial learning is defined as linear interpolation of L_{Adv} and L_{Rec}:

$$L = \alpha L_{Adv} + (1-\alpha)L_{Rec} \tag{7}$$

where α is the trade-off coefficient between adversarial and reconstruction losses. And the objective of adversarial learning takes a min-max form:

$$\max_{\Theta_D}\left(\min_{\Theta_G} L\right) \tag{8}$$

where $\Theta_G = \{\theta_g, \theta'_g, V_g, b_g\}$ are generators' parameters, and Θ_D are discriminator's parameters which will be detailed in the next section. In the min-max process, we first optimize Θ_G by minimizing adversarial loss L_{Adv} (i.e., maximizing discriminator loss L_D) and reconstruction loss L_{Rec} to generate confusing but reversible examples; we then optimize discriminator parameters Θ_D for classification by maximizing adversarial loss L_{Adv} (i.e., minimizing discriminator loss L_D), and note that L_{Rec} is independent of Θ_D.

3.3 Rumor Discriminator

With the training data augmented with the generative processing, the discriminator learns to capture more discriminative features, especially from low-frequency non-trivial patterns.

We build the discriminator based on a RNN rumor detection model [12]. Given an instance (either original or generated), the RNN model first maps relevant posts x_t at the t-th step into a hidden vector s_t using GRU:

$$s_t = \text{GRU}(x_t, s_{t-1}; \theta_d) \tag{9}$$

where s_{t-1} is the previous hidden vector and θ_d denotes all the GRU parameters in discriminator.

Following [12], we feed the hidden vector s_T at the last time step as the representation into a 2-class softmax function for classifying the instance:

$$\hat{y} = \text{softmax}(V_d s_T + b_d) \tag{10}$$

Algorithm 1: Generative adversarial training procedure.

Input : A set of training claims $\{X\}$, learning rate ϵ

1 Initialize Θ_G, and Θ_D with random weight values;
2 **for** *epoch from 1 to maxIter* **do**
3 **for** *each mini-batch* $\{\{X_N\},\{X_R\}\}$ **do**
4 Generate $\{\tilde{X}_N\}$: $\{X_N \oplus G_{N\to R}(X_N)\} \to \{\tilde{X}_N\}$;
5 Generate $\{\tilde{X}_R\}$: $\{X_R \oplus G_{R\to N}(X_R)\} \to \{\tilde{X}_R\}$;
6 Augment training set: $\big\{\{X_N\}\cup\{\tilde{X}_N\}, \{X_R\}\cup\{\tilde{X}_R\}\big\}$;
7 Compute loss L using Eq. 7;
  `/* Minimize L w.r.t. Θ_G */`
8 Compute gradient $\nabla(\Theta_G)$;
9 Update generators: $\Theta_G \leftarrow \Theta_G - \epsilon\nabla(\Theta_G)$;
  `/* Maximize L w.r.t. Θ_D */`
10 Compute gradient $\nabla'(\Theta_D)$;
11 Update discriminator: $\Theta_D \leftarrow \Theta_D - \epsilon\nabla'(\Theta_D)$;
12 **end for**
13 **end for**

where $\hat{y}$ is the vector of predicted probabilities over the two classes, V_d is the weight matrix of output layer and b_d is the trainable bias.

The loss of discriminator is defined as the square error between distributions of the predicted class and the ground-truth class:

$$L_D(\bar{y}, \hat{y}) = ||\bar{y} - \hat{y}||_2^2 + \lambda||\Theta_D||_2^2 \tag{11}$$

where $\bar{y}$ and $\hat{y}$ are respectively the ground-truth and predicted class probability distributions, $\Theta_D = \{\theta_d, V_d, b_d\}$ are discriminator parameters, and λ is the trade-off coefficient.

3.4 Generative Adversarial Training Algorithm

Algorithm 1 presents the iterative training process of the generators and discriminator in our GAN-style framework. Unlike original GAN [7] for obtaining better generators, our goal is to reinforce the discriminator to be more discriminative and generalizable.

The generators and discriminator are alternately trained using stochastic gradient decent with mini-batches [2]. In each epoch, controversial examples are generated and augmented into the original training data. We optimize the generator and discriminator against the augmented training examples with Eq. 8 that enforces a min-max game through steps 8-11 in Algorithm 1.

In training, we initialize model parameters with uniform distribution and update them by employing the derivative of the loss through back-propagation [5]; we iterate training until the maximum epoch number is met, which is set as 200; we fix the vocabulary size as 5,000, the size of hidden vector as 100, and tune the hyper parameters α, λ and ϵ using held-out dataset; post sequences take variable length dependent of specific instances by following [12].

4 EXPERIMENTS AND RESULTS

4.1 Datasets

We resort to two public datasets TWITTER [12] and PHEME [22] for experimental evaluation[3] The two datasets were used for binary classification of rumor and non-rumor with respect to a claim via

[3]https://figshare.com/articles/PHEME_dataset_of_rumours_and_non-rumours/4010619.

Table 1: Statistics of the datasets

Statistic	TWITTER	PHEME
Users #	491,229	29,387
claims #	992	2,246
Non-Rumors #	498	1,123
Rumors #	494	1,123
Avg. time length / claim	1,582.6 Hours	19.9 Hours
Avg. # of posts / claim	1,111	26
Max # of posts / claim	62,827	289
Min # of posts / claim	10	12

its relevant tweets. In particular, PHEME were collected based on 5 breaking news, thus its claims overlap more than TWITTER which was collected based on the claims reported on snopes.com. Moreover, we filtered out claims with less than 10 tweets and balanced the number of instances of the two classes. Statistics of the resulting datasets are given in Table 1.

4.2 Experimental Setup

We made comparisons among the following models:

DT-Rank: A Decision-Tree-based Ranking method that identifies trending rumors [21] through searching for disputed claims.

DTC: A Decision Tree Classifier for modeling Twitter information credibility [3] using various handcrafted features.

SVM-TS: A linear SVM classification model that uses time series to model the chronological variation of social context features [13].

BOW: A naive baseline representing the texts using bag-of-words and building the rumor classifier with linear SVM.

GRU: A RNN-based rumor detection model [12] with GRU for representation learning of relevant posts over time.

CNN: A CNN-based model for misinformation identification [20] for learning rumor representations by framing the relevant posts as fixed-length sequence.

GAN-GRU, **GAN-CNN** and **GAN-BOW**: Our GAN-style learning models where the discriminator adopts the data-driven models **GRU**, **CNN** and **BOW** above, respectively. Since replacing GRU with CNN or BOW as discriminator is straightforward, we omit describing the structures of **GAN-CNN** and **GAN-BOW**.

We hold out 10% of the claims in each dataset for tuning the hyper parameters, and for the rest of the claims, we conduct 5-fold cross-validation and use accuracy, precision, recall and F-measure as evaluation metrics. We implemented these models under comparison and will release our source codes publicly[4].

4.3 Result and Analysis

Table 2–3 demonstrate the performance of all the compared models based on the two datasets. The results indicate that our **GAN-GRU** model outperforms all the baselines, which confirms the advantage of generative adversarial learning for rumor detection task.

It is observed that the 3 baselines based on hand-crafted features perform clearly worse than all the 6 purely data-driven models. **SVM-TS** is relatively better because it incorporate additional temporal information into the conventional models. The results of

[4]https://github.com/majingCUHK/Rumor_GAN

Table 2: Results of comparison on TWITTER dataset

Method	Class	Accu.	Prec.	Rec.	F_1
DT-Rank	R	0.674	0.652	0.814	0.724
	N		0.716	0.519	0.602
DTC	R	0.732	0.694	0.794	0.741
	N		0.778	0.675	0.723
SVM-TS	R	0.798	0.778	0.837	0.807
	N		0.823	0.760	0.790
BOW	R	0.775	0.746	0.833	0.787
	N		0.811	0.716	0.761
CNN	R	0.821	0.815	0.831	0.823
	N		0.829	0.810	0.819
GRU	R	0.835	0.821	0.858	0.839
	N		0.852	0.812	0.832
GAN-BOW	R	0.792	0.761	0.850	0.803
	N		0.830	0.733	0.779
GAN-CNN	R	0.852	**0.853**	0.850	0.851
	N		0.853	**0.852**	0.852
GAN-GRU	R	**0.863**	0.843	**0.892**	**0.866**
	N		**0.885**	0.833	**0.858**

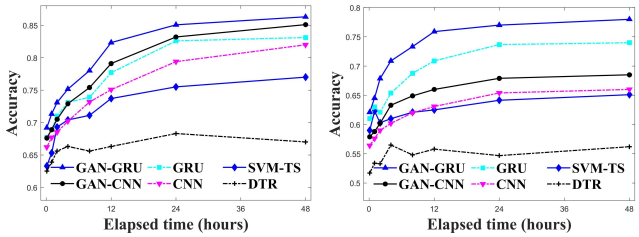
Table 3: Results of comparison on PHEME dataset

Method	Class	Accu.	Prec.	Rec.	F_1
DT-Rank	R	0.562	0.588	0.421	0.491
	N		0.549	0.704	0.617
DTC	R	0.581	0.582	0.573	0.578
	N		0.579	0.588	0.584
SVM-TS	R	0.651	0.663	0.617	0.639
	N		0.642	0.686	0.663
BOW	R	0.704	0.724	0.675	0.699
	N		0.687	0.734	0.710
CNN	R	0.665	0.671	0.652	0.661
	N		0.661	0.679	0.669
GRU	R	0.742	0.737	0.753	0.745
	N		0.754	0.730	0.739
GAN-BOW	R	0.736	0.755	0.701	0.727
	N		0.721	**0.772**	0.745
GAN-CNN	R	0.688	0.683	0.698	0.690
	N		0.695	0.678	0.685
GAN-GRU	R	**0.781**	**0.773**	**0.796**	**0.784**
	N		**0.791**	0.766	**0.778**

DT-Rank are poor due to the low coverage of the patterns using the regular expressions it defined.

BOW performs surprisingly well which is comparable to or even outperforms using hand-crafted features which confirms the advantage of using the simplest data-driven approach. **GRU** performs the best among all the baselines, which is not surprising since it takes advantage of deep neural nets to capture complex hidden features indicative of rumors beyond explicit and shallow patterns.

We conjectured that **CNN** should be comparable to **GRU** because both can learn deep latent features from data. This turned out to be incorrect on PHEME where CNN performs much worse. The reason is that RNN can inherently deal with variable-length sequence while CNN is essentially not a sequential model. The relevant posts per



(a) TWITTER dataset (b) PHEME dataset

Figure 4: Results of rumor early detection

claim in PHEME is significantly fewer than TWITTER, rendering lots of zero-valued input units to CNN that can worsen convolution operations, but RNN can easily get rid of zero input units by shortening sequence length. This also explains the overall better performance of all models on TWITTER.

Clearly, our three generative adversarial models outperform their counterparts in baselines that are not generative adversarial. The improvements of our models over these baselines range from 2.6% to 5.3% on the two datasets, indicating the adversarial learning with the generative-discriminative process is generally helpful and effective. However, two reasons may prevent further improvements on the two datasets: 1) The post content associated with PHEME claims overlap heavily since they come from only 5 breaking news, rendering high-frequency patterns more sensitive to topical categories than veracity categories. The claims in TWITTER dataset are however easier to classify as per veracity since each of them is an independent news topic, making high-frequency patterns well correlated to veracity rather than topic[5]. The potential of GAN-style learning by promoting the chance of low-frequency patterns is thus somewhat limited on TWITTER. This can be implied by the relatively lower improvement obtained on TWITTER than PHEME. 2) In addition to overlapping topics, PHEME is harder to classify also because there are only 26 posts per claim in average, thus useful information available is relatively limited.

Furthermore, rumor detection task emphasizes the identification performance on the rumor category. GAN-GRU achieves the highest recall on rumor category, indicating that more rumors can be found. From a balanced view of precision and recall, we also observe that the F1 scores of rumor category achieved by GAN-GRU are higher than non-rumor category.

4.4 Early Rumor Detection

Early alerts of rumors can prevent further spreading of rumorous content. By setting a detection delay time, only tweets posted no later than the delay can be used for evaluating early detection performance. We compare the accuracies of different baselines and our models as shown in Figure 4.

The accuracies of all methods increase with elapsed time, and our models demonstrate clear advantages at early stage. Particularly, **GAN-GRU** using less than 12-hour data, has already outperformed **GRU** (the best baseline) using all time data, indicating the superior early detection performance of our model. This is due to the generation component that can enrich training data at early stage when the volume of actual posts is generally low.

[5]This can be confirmed by the relatively higher performance of all models on TWITTER.

Table 4: Examples of original and generated posts

claim	**[Non-rumor:] Augusta County School Close over Arabic Assignment**
Real Posts	Believe that is scary ? Augusta County schools closed Friday ... Schools are shut down for a homework assignment on Arabic Augusta County schools closed Friday... Oh, the stupid -it burns. My god rt : Augusta County schools closed Friday ... Clear violation of church and state. The teacher was clearly wrong ...
$G_{N \to R}$	The assignment in school not well with great wrong , I disagree ... Why close school? seriously? I not believe . not sure but wrong :d I think the assignment will be a mess , do harm to... so disappointed . Those teacher have fault, that was crazy, not a truth , totally a rumor I am confused of not enough report about us. Will be a joke
claim	**[Rumor:] Dearborn Implement Sharia Law**
Real Posts	A satire website produces fake news for entertainment. An oninion-style ... No movement implement Sharia Law Dearborn just sustain Shawarma Law Link me your source, even better, public voting record doesn't exist I can assure you that no one is trying to implement Sharia Law. Is this true? the city council of Dearborn, Michigan became the first
$G_{R \to N}$	A Claim that Law probably offense and out of control :p #Wakeupamerica, A Claim expose that is well great via fully report Omg, a claim earn simultaneously smile and pleasant , Seriously? Report that breaking for nyc, I mean it Awesome! no rumor just fact I discover it a total fact , but claim is illegal, will appear to be wrong .

We also conduct experiments to explain why the generator can boost the discriminator in an adversarial manner. We sample a rumor and a non-rumor claim from TWITTER, and list some generated contents in Table 4. We observe that 1) the generations although seemed non-grammatical are relevant to the input claim; 2) the generations can distort the real-world posts by using conflicting or uncertain expressions; 3) incorporating the generations counterbalances the discriminative power of high-frequency patterns (in yellow), implying a higher chance of lower-frequency features (in blue) being captured by the discriminator.

Given the limited number of claims in the datasets, our models perform reasonably well. This is because there are a good number of relevant tweets per claim, where many indicative patterns exist, such as those highlighted keywords and phrases in Table 4. Therefore, the feature space is not sparse, which is generally advantageous for generators to generate diverse campaign-style texts and for the discriminator to capture discriminative features.

5 CONCLUSION AND FUTURE WORK

We propose a novel GAN-style model that can generate and exploit the effect of information campaigns for better rumor detection. Our neural-network-based generators create training examples to confuse rumor discriminator so that the discriminator are forced to learn more powerful features from the augmented training data. Experimental results confirm that our method is effective and robust based on two public benchmark datasets for rumor detection on Twitter. In our future work, we plan to use GAN to generate structured data such as rumor propagation trees to boost rumor detection performance and compare with structured models [14, 15, 19].

ACKNOWLEDGMENT

This work is partly supported by Innovation and Technology Fund (ITP/004/16LP), TBF (TBF18ENG002) and General Research Fund of Hong Kong (14232816, 14209416, 14204118).

REFERENCES

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

[2] Antoine Bordes, Léon Bottou, and Patrick Gallinari. 2009. Sgd-qn: Careful quasi-newton stochastic gradient descent. *Journal of Machine Learning Research* 10, Jul (2009), 1737–1754.

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

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

[5] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, Aug (2011), 2493–2537.

[6] Emily L Denton, Soumith Chintala, Rob Fergus, et al. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*. 1486–1494.

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*. 2672–2680.

[8] Sejeong Kwon, Meeyoung Cha, and Kyomin Jung. 2017. Rumor Detection over Varying Time Windows. *PLOS ONE* 12, 1 (2017), e0168344.

[9] Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent features of rumor propagation in online social media. In *2013 IEEE 13th International Conference on Data Mining*. 1103–1108.

[10] 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*. 1867–1870.

[11] Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 1412–1421.

[12] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. 3818–3824.

[13] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. 2015. Detect Rumors Using Time Series of Social Context Information on Microblogging Websites. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1751–1754.

[14] Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect Rumors in Microblog Posts Using Propagation Structure via Kernel Learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 708–717.

[15] Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor Detection on Twitter with Tree-structured Recursive Neural Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1980–1989.

[16] Feng Qian, Chengyue Gong, Karishma Sharma, and Yan Liu. 2018. Neural user response generator: fake news detection with collective user intelligence. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 3834–3840.

[17] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. 3104–3112.

[18] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 849–857.

[19] Ke Wu, Song Yang, and Kenny Q Zhu. 2015. False rumors detection on sina weibo by propagation structures. In *Proceedings of 2015 IEEE 31st International Conference on Data Engineering*. 651–662.

[20] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2017. A convolutional approach for misinformation identification. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 3901–3907.

[21] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Enquiring Minds: Early Detection of Rumors in Social Media from Enquiry Posts. In *Proceedings of the 24th International Conference on World Wide Web*. 1395–1405.

[22] Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2017. Exploiting context for rumour detection in social media. In *International Conference on Social Informatics*. Springer, 109–123.