Detecting Deceptive Reviews using Generative Adversarial Networks

Hojjat Aghakhani[1], Aravind Machiry[1], Shirin Nilizadeh[2], Christopher Kruegel[1], and Giovanni Vigna[1]

[1]University of California, Santa Barbara
[1]{hojjat, machiry, chris, vigna}@cs.ucsb.edu
[2]Carnegie Mellon University Silicon Valley
[2]shirin.nilizadeh@sv.cmu.edu

***Abstract*—In the past few years, consumer review sites have become the main target of *deceptive opinion spam*, where fictitious opinions or reviews are deliberately written to sound authentic. Most of the existing work to detect the deceptive reviews focus on building supervised classifiers based on syntactic and lexical patterns of an opinion. With the successful use of Neural Networks on various classification applications, in this paper, we propose FakeGAN a system that for the first time augments and adopts Generative Adversarial Networks (GANs) for a text classification task, in particular, detecting deceptive reviews.**

Unlike standard GAN models which have a single Generator and Discriminator model, FakeGAN uses two discriminator models and one generative model. The generator is modeled as a stochastic policy agent in reinforcement learning (RL), and the discriminators use Monte Carlo search algorithm to estimate and pass the intermediate action-value as the RL reward to the generator. Providing the generator model with two discriminator models avoids the mod collapse issue by learning from both distributions of truthful and deceptive reviews. Indeed, our experiments show that using two discriminators provides FakeGAN high stability, which is a known issue for GAN architectures. While FakeGAN is built upon a *semi-supervised classifier*, known for less accuracy, our evaluation results on a dataset of TripAdvisor hotel reviews show the same performance in terms of accuracy as of the state-of-the-art approaches that apply supervised machine learning. These results indicate that GANs can be effective for text classification tasks. Specifically, FakeGAN is effective at detecting deceptive reviews.

I. Introduction

In the current world, we habitually turn to the wisdom of our peers, and often complete strangers, for advice, instead of merely taking the word of an advertiser or business owner. A 2015 study by marketing research company Mintel [1] found nearly 70 percent of Americans seek out others' opinions online before making a purchase. Many platforms such as Yelp.com and TripAdvisor.com have sprung up to facilitate this sharing of ideas amongst users. The heavy reliance on review information by the users has dramatic effects on business owners. It has been shown that an extra half-star rating on Yelp helps restaurants to sell out 19 percentage points more frequently [2].

This phenomenon has also lead to a market for various kinds of fraud. In simple cases, this could be a business rewarding its customers with a discount, or outright paying them, to write a favorable review. In more complex cases, this could involve astroturfing, opinion spamming [3] or *deceptive opinion spamming* [4], where fictitious reviews are deliberately written to sound authentic. Figure 1 shows an example of a truthful and deceptive review written for the same hotel. It is estimated that up to 25% of Yelp reviews are fraudulent [5], [6].

Detecting deceptive reviews is a text classification problem. In recent years, deep learning techniques based on natural language processing have been shown to be successful for text classification tasks. Recursive Neural Network (RecursiveNN) [7], [8], [9] has shown good performance classifying texts, while Recurrent Neural Network (RecurrentNN) [10] better captures the contextual information and is ideal for realizing semantics of long texts. However, RecurrentNN is a biased model, where later words in a text have more influence than earlier words [11]. This is not suitable for tasks such as detection of deceptive reviews that depend on an unbiased semantics of the entire document (review). Recently, techniques based on Convolutional Neural Network (CNN) [12], [13] were shown to be effective for text classification. However, the effectiveness of these techniques depends on careful selection of the window size [11], which controls the parameter space.

Moreover, in general, the main problem with applying classification methods for detecting deceptive reviews is the lack of substantial ground truth datasets required for most of the supervised machine learning techniques. This problem worsens for neural networks based methods, whose complexity requires much bigger dataset to reach a reasonable performance.

To address the limitations of the existing techniques, we propose FakeGAN, which is a technique based on Generative Adversarial Network (GAN) [14]. GANs are a class of artificial intelligence algorithms used in unsupervised machine learning, implemented by a system of two neural networks contesting with each other in a zero-sum game framework. GANs have been used mostly for image-based applications [14], [15], [16], [17]. In this paper, for the first time, we propose the use of GANs for a text classification task, i.e., detecting deceptive reviews. Moreover, the use of a semi-supervised learning method like GAN can eliminate the problem of ground truth scarcity that in general hinders the detection success [4], [18], [19].

We augment GAN models for our application in such a way that unlike standard GAN models which have a single Generator and Discriminator model, FakeGAN uses two discriminator

models D, D' and one generative model G. The discriminator model D tries to distinguish between truthful and deceptive reviews whereas D' tries to distinguish between reviews generated by the generative model G and samples from *deceptive* reviews distribution. The discriminator model D' helps G to generate reviews close to the deceptive reviews distribution, while D helps G to generate reviews which are classified by D as truthful.

Our intuition behind using two discriminators is to create a stronger generator model. If in the adversarial learning phase, the generator gets rewards only from D, the GAN may face the mod collapse issue [20], as it tries to learn two different distributions (truthful and deceptive reviews). The combination of D and D' trains G to generate better deceptive reviews which in turn train D to be a better discriminator.

Indeed, our evaluation using the TripAdvisor[1] hotel reviews dataset shows that the discriminator D generated by FakeGAN performs on par with the state-of-the-art methods that apply supervised machine learning, with an accuracy of 89.1%. These results indicate that GANs can be effective for text classification tasks, specifically, FakeGAN is effective at detecting deceptive reviews. To the best of our knowledge, FakeGAN is the first work that use GAN to generate better discriminator model (i.e., D) in contrast to the common GAN applications which aim to improve the generator model.

In summary, following are our contributions:

1) We propose FakeGAN, a deceptive review detection system based on a double discriminator GAN.
2) We believe that FakeGAN demonstrates a good first step towards using GANs for text classification tasks.
3) To the best of our knowledge, FakeGAN is the first system using semi-supervised neural network-based learning methods for detecting deceptive fraudulent reviews.
4) Our evaluation results demonstrate that FakeGAN is as effective as the state-of-the-art methods that apply supervised machine learning for detecting deceptive reviews.

II. Approach

Generative Adversarial Network (GAN) [14] is a promising framework for generating high-quality samples with the same distribution as the target dataset. FakeGAN leverages GAN to learn the distributions of truthful and deceptive reviews and to build a semi-supervised classifier using the corresponding distributions.

A GAN consists of two models: a generative model G which tries to capture the data distribution, and a discriminative model D that distinguishes between samples coming from the training data or the generator G. These two models are trained simultaneously, where G is trying to fool the discriminator D, while D is maximizing its probability estimation that whether a sample comes from the training data or is produced by the generator. In a nutshell, this framework corresponds to a minimax two-player game.

[1]Tripadvisor.com

The feedback or the gradient update from discriminator model plays a vital role in the effectiveness of a GAN. In the case of text generation, it is difficult to pass the gradient update because the generative model produces discrete tokens (words), but the discriminative model makes a decision for complete sequence or sentence. Inspired by SeqGAN [21] that uses GAN model for Chinese poem generation, in this work, we model the generator as a stochastic policy in reinforcement learning (RL), where the gradient update or RL reward signal is provided by the discriminator using Monte Carlo search. Monte Carlo is a heuristic search algorithm for identifying the most promising moves in a game. In summary, in each state of the game, it plays out the game to the very end for a fixed number of times according to a given policy. To find the most promising move, it must be provided by reward signals for a complete sequence of moves.

All the existing applications use GAN to create a strong generator, where the main issue is the convergence of generator model [22], [23], [20]. *Mode collapse* in particular is a known problem in GANs, where complexity and multimodality of the input distribution cause the generator to produce samples from a single mode. The generator may switch between modes during the learning phase, and this cat-and-mouse game may never end [24], [20]. Although no formal proof exists for convergence, in Section III we show that the FakeGAN's discriminator converges in practice.

Unlike the typical applications of GANs, where the ultimate goal is to have a strong generator, FakeGAN leverages GAN to create a well-trained discriminator, so that it can successfully distinguish truthful and deceptive reviews. However, to avoid the stability issues inherent to GANs we augment our network to have two discriminator models though we use only one of them as our intended classifier. Note that leveraging samples generated by the generator makes our classifier a *semi-supervised* classifier.

Definitions

We start with defining certain symbols which will be used throughout this section to define various steps of our approach. The training dataset, $X = X_D \cup X_T$, consists of two parts, deceptive reviews X_D and truthful reviews X_T. We use χ to denote the vocabulary of all tokens (i.e., words) which are available in X.

Our generator model G_α parametrized by α produces each review $S_{1:L}$ as a sequence of tokens of length L where $S_{1:L} \in \chi^L$. We use Z_G to indicate all the reviews generated by our generator model G_α.

We use two discriminator models D and D'. The discriminator D distinguishes between truthful and deceptive reviews, as such $D(S_{1:L})$ is the probability that the sequence of tokens comes from X_T or $X_D \cup Z_G$. Similarly, D' distinguishes between deceptive samples in the dataset and samples generated by G_α consequently $D'(S_{1:L})$ is a probability indicating how likely the sequence of tokens comes from X_D or Z_G.

The discriminator D' guides the generator G_α to produce samples similar to X_D whereas D guides G_α to generate

"We loved the hotel. When I see other posts about it being shabby I can't for the life of me figure out what they are talking about. Rooms were large with TWO bathrooms, lobby was fabulous, pool was large with two hot tubs and huge gym, staff was courteous. For us, the location was great--across the street from Grant Park with a great view of Buckingham Fountain and close to all the museums and theatres. I'm sure others would rather be north of the river closer to the Magnificent Mile but we enjoyed the quieter and more scenic location. Got it for $105 on Hotwire. What a bargain for such a nice hotel."

(a) A truthful review provided by a high profile user on TripAdvisor

"My husband and I satayed for two nights at the Hilton Chicago,and enjoyed every minute of it! The bedrooms are immaculate,and the linnens are very soft. We also appreciated the free wifi,as we could stay in touch with friends while staying in Chicago. The bathroom was quite spacious,and I loved the smell of the shampoo they provided-not like most hotel shampoos. Their service was amazing, and we absolutely loved the beautiful indoor pool. I would recommend staying here to anyone."

(b) A deceptive review written by an Amazon Mechanical worker

Fig. 1: A truthful review versus a deceptive review, both written for the same hotel.

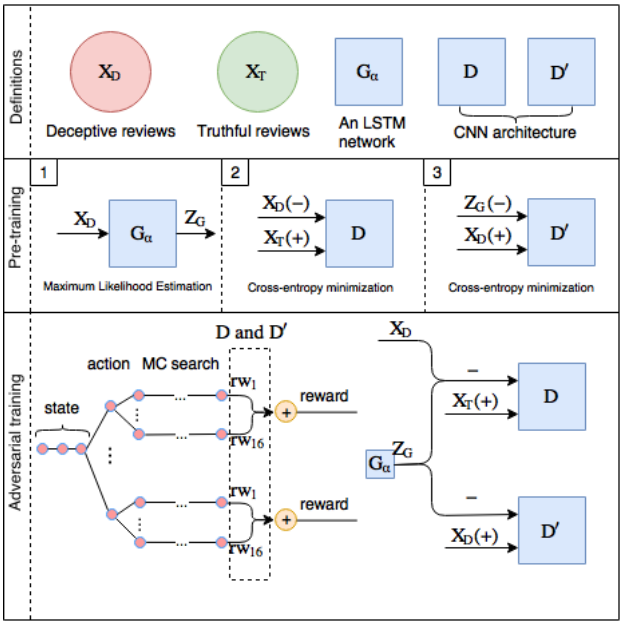


Fig. 2: The overview of FakeGAN. The symbols + and − indicates positive and negative samples respectively. Note that, these are different from truthful and deceptive reviews.

samples which seems truthful to D. So in each round of training, by using the feedback from D and D', the generator G_α tries to fool D' and D by generating reviews that seems deceptive (not generated by G_α) to D', and truthful (not generated by G_α or comes from X_D) to D.

Figure 2 shows an overview of FakeGAN. During pre-training, we use the Maximum Likelihood Estimation (MLE) to train the generator G_α on deceptive reviews X_D from the training dataset. We also use minimizing the cross-entropy technique to pre-train the discriminators.

The generator G_α is defined as a policy model in reinforcement learning. In timestep t, the state s is the sequence of produced tokens, and the action a is the next token. The policy model $G_\alpha(S_t|S_{1:t-1})$ is stochastic. Furthermore, the generator G_α is trained by using a policy gradient and Monte Carlo (MC) search on the expected end reward from the discriminative models D and D'. Similar to [21], we consider the estimated probability $D(S_{1:L}) + D'(S_{1:L})$ as the reward. Formally, the corresponding action-value function is:

$$A_{G_\alpha,D,D'}(a = S_L, s = S_{1:L-1}) = D(S_{1:L}) + D'(S_{1:L}) \quad (1)$$

As mentioned before, G_α produces a review token by token. However, the discriminators provide the reward for a complete sequence. Moreover, G_α should care about the long-term reward, similar to playing Chess where players sometimes prefer to give up immediate good moves for a long-term goal of victory [25]. Therefore, to estimate the action-value function in every timestep t, we apply the Monte Carlo search N times with a roll-out policy G'_γ to sample the undetermined last $L-t$ tokens. We define an N-time Monte Carlo search as

$$\{S^1_{1:L}, S^2_{1:L}, ..., S^N_{1:L}\} = MC_{G'_\gamma}(S_{1:t}, N) \quad (2)$$

where for $1 \leq i \leq N$

$$S^i_{1:t} = (S_1, ..., S_t) \quad (3)$$

and $S^i_{t+1:L}$ is sampled via roll-out policy G'_γ based on the current state $S^i_{1:t-1}$. The complexity of action-value estimation function mainly depends on the roll-out policy. While one might use a simple version (e.g., random sampling or sampling based on n-gram features) as the policy to train the GAN fast, to be more efficient, we use the same generative model ($G'_\gamma = G_\alpha$ at time t). Note that, a higher value of N results in less variance and more accurate evaluation of the action-value function. We can now define the action-value estimation function at t as

$$A_{G_\alpha,D,D'}(a = S_t, s = S_{1:t-1}) = \begin{cases} \frac{1}{N}\sum_{i=1}^{N}(D(S^i_{1:L}) + D'(S^i_{1:L})) & \text{if } t \leq L \\ D(S_{1:L}) + D'(S_{1:L}) & \text{if } t = L \end{cases} \quad (4)$$

where $S^i_{1:L}$s are created according to the Equation 2. As there is no intermediate reward for the generator, we define the the objective function for the generator G_α (based on [26]) to produce a sequence from the start state S_0 to maximize its final reward:

$$J(\alpha) = \sum_{S_1 \in \chi} G_\alpha(S_1|S_0) \; . \; A_{G_\alpha,D,D'}(a = S_1, s = S_0) \quad (5)$$

Conseqently, the gradient of the objective function $J(\alpha)$ is:

$$\nabla_\alpha J(\alpha) = \sum_{t=1}^{T} \mathbb{E}_{S_{1:t-1} \sim G_\alpha}[\sum_{S_t \in \chi} \nabla_\alpha G_\alpha(S_t|S_{1:t-1}) \; . \; A_{G_\alpha,D,D'}(a = S_t, s = S_{1:t-1})] \quad (6)$$

We update the generator's parameters (α) as:

$$\alpha \leftarrow \alpha + \lambda \nabla_\alpha J(\alpha) \tag{7}$$

where λ is the learning rate.

By dynamically updating the discriminative models, we can further improve the generator. So, after generating g samples, we will re-train the discriminative models D and D' for d steps using the following objective functions respectively:

$$min(-\mathbb{E}_{S\sim X_T}[\log D(S)] - \mathbb{E}_{S\sim X_D \vee G_\alpha}[1 - \log D(S)]) \tag{8}$$

$$min(-\mathbb{E}_{S\sim X_D}[\log D'(S)] - \mathbb{E}_{S\sim G_\alpha}[1 - \log D'(S)]) \tag{9}$$

In each of the d steps, we use G_α to generate the same number of samples as number of truthful reviews i.e., $|X_G| = |X_T|$. The updated discriminators will be used to update the generator, and this cycle continues until FakeGAN converges. Algorithm 1 formally defines all the above steps.

Algorithm 1 FakeGAN

Require: discriminators D and D', generator G_α, roll-out policy G_γ, dataset X
Initialize α with random weight.
Load word2vec vector embeddings into G_α, D and D' models
Pre-train G_α using MLE on X_D
Pre-train D by minimizing the cross entropy
Generate negative examples by G_α for training D'
Pre-train D' by minimizing the cross entropy
$\gamma \leftarrow \alpha$
repeat
  for g-steps **do**
    Generate a sequence of tokens $S_{1:L} = (S_1, ..., S_L) \sim G_\alpha$
    for t in $1 : L$ **do**
      Compute $A_{G_\alpha, D_\beta, D'_\theta}(a = S_t, s = S_{1:t-1})$ by Eq. 4
    end for
    Update α via policy gradient Eq. 7
  end for
  for d-steps **do**
    Use G_α to generate X_G.
    Train discriminator D by Eq. 8
    Train discriminator D' by Eq. 9
  end for
  $\gamma \leftarrow \alpha$
until D reaches a stable accuracy.

The Generative Model

We use RecurrentNNs (RNNs) to construct the generator. An RNN maps the input embedding representations $s_1, ..., s_L$ of the input sequence of tokens $S_1, ..., S_L$ into hidden states $h_1, ..., h_L$ by using the following recursive function.

$$h_t = g(h_{t-1}, s_t) \tag{10}$$

Finally, a softmax output layer z with bias vector c and weight matrix V maps the hidden layer neurons into the output token distribution as

$$p(s|s_1, ..., s_t) = z(h_t) = \text{softmax}(c + V.h_t) \tag{11}$$

To deal with the common vanishing and exploding gradient problem [27] of the backpropagation through time, we exploit the Long Short-Term Memory (LSTM) cells [28].

The Discriminator Model

For the discriminators, we select the CNN because of their effectiveness for text classification tasks [29]. First, we construct the matrix of the sequence by concatenating the input embedding representations of the sequence of tokens $s_1, ..., s_L$ as:

$$\zeta_{1:L} = s_1 \oplus ... \oplus s_L \tag{12}$$

Then a kernel w computes a convolutional operation to a window size of l by using a non-linear function π, which results in a feature map:

$$f_i = \pi(w \otimes \zeta_{i:i+l-1} + b) \tag{13}$$

Where $\otimes$ is the inner product of two vectors, and b is a bias term. Usually, various numbers of kernels with different window sizes are used in CNN. We hyper-tune size of kernels by trying kernels which have been successfully used in text classification tasks by community [13], [30], [11]. Then we apply a max-over-time pooling operation over the feature maps to allow us to combine the outputs of different kernels. Based on [31] we add the highway architecture to improve the performance. In the end, a fully connected layer with sigmoid activation functions is used to output the class probability of the input sequence.

III. Evaluation

We implemented FakeGAN using the TensorFlow [32] framework. We chose the dataset from [4] which has 800 reviews of 20 Chicago hotels with positive sentiment. The dataset consists of 400 truthful reviews provided by high profile users on TripAdvisor and 400 deceptive reviews written by Amazon Mechanical Workers. To the best of our knowledge, this is the biggest available dataset of labeled reviews and has been used by many related works [4], [18], [33]. Similar to SeqGAN [21], the generator in FakeGAN only creates fixed length sentences. Since the majority of reviews in this dataset has a length less than 200 words, we set the sequence length of FakeGAN (L) to 200. For sentences whose length is less than 200, we pad them with a fixed token <END> to reach the size of 200 resulting in 332 truthful and 353 deceptive reviews. Note that, having a larger dataset results in a less training time. Although larger dataset makes each adversarial step slower, it provides G a richer distribution of samples, thus reduces the number of adversarial steps resulting in less training time.

We used the k-fold cross-validation with k=5 to evaluate FakeGAN. We leveraged GloVe vectors[2] for word representation [34]. Similar to SeqGAN [21], the convergence

[2]Check "glove.6B.200d.txt" from https://nlp.stanford.edu/projects/glove/

of FakeGAN varies with the training parameters g and d of generator and discriminative models respectively. After experimenting with different values, we observed that following values $g = 1$ and $d = 6$ are optimal. For pre-training phase, we trained the generator and the discriminators until convergence, which took 120 and 50 steps respectively. The adversarial learning starts after the pre-training phase. All our experiments were run on a 40-core machine, where the pre-training took $\sim$one hour, and the adversarial training took $\sim$11 hours with a total of $\sim$12 hours.

A. Accuracy of Discriminator D

As mentioned before, the goal of FakeGAN is to generate a highly accurate discriminator model, D, that can distinguish deceptive and truthful reviews. Figure 3a shows the accuracy trend for this model; for simplicity, the trend is shown only for the first iteration of k-fold cross-validation. During the pre-training phase, the accuracy of D stabilized at 50^{th} step. We set the adversarial learning to begin at step 51. After a little decrease in accuracy at the beginning, the accuracy increases and converges to 89.2%, which is on-par with the accuracy of state-of-the-art approach [4] that applied supervised machine learning on the same dataset ($\sim 89.8\%$). The accuracy, precision and recall for k-fold cross-validation are 89.1%, 98% and 81% all with a standard deviation of 0.5. This supports our hypothesis that adversarial training can be used for detecting deceptive reviews. Interestingly even though FakeGAN relies on semi-supervised learning, it yields similar performance as of a fully-supervised classification algorithm.

B. Accuracy of Discriminator D'

Figure 3b shows the accuracy trend for the discriminator D'. Similar to D, D' converges after 450 steps with an accuracy of $\sim 99\%$ accuracy. It means that at this point, the generator G will not be able to make any progress trying to fool D', and the output distribution of G will stay almost same. Thus, continuing adversarial learning does not result in any improvement of the accuracy of our main discriminator, D.

C. Comparing FakeGAN with the original GAN approach

To justify the use of two discriminators in FakeGAN, we tried using just one discriminator (only D) in two different settings. In the first case, the generator G is pre-trained to learn only *truthful reviews* distribution. Here the discriminator D reached 83% accuracy in pre-training, and the accuracy of adversarial learning, i.e., the classifier, reduces to about 65%. In the second case, the generator G is pre-trained to learn only *deceptive reviews* distribution. Unlike the first case, adversarial learning improved the performance of D by converging at 84%, however, still, the performance is lower than that of FakeGAN.

These results demonstrate that using two discriminators is necessary to improve the accuracy of FakeGAN.

D. Scalability Discussion

We argue that the time complexity of our proposed augmented GAN with two discriminators is the same as of original GANs because their bottleneck is the MC search, where using the rollout policy (which is G until the time) generates 16 complete sequences, to help the generator G for just outputting the most promising token as its current action. This happens for every token of a sequence which is generated by G. However, compared to MC search, discriminators D and D' are efficient and not time-consuming.

E. Stability Discussion

As we discussed in Section II, the *stability* of GANs is a known issue. We observed that the parameters g and d have a large effect on the convergence and performance of FakeGAN as illustrated in the Figure 4, when d and g are both equal to one. We believe that the stability of GAN makes hyper-tuning of FakeGAN a challenging task thus prevents it from outperforming the state-of-the-art methods based on supervised machine learning. However, with the following values $d = 6$ and $g = 1$, FakeGAN converges and performs on par with the state-of-the-art approach.

IV. Related work

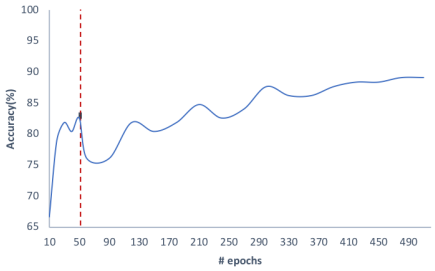
Text classification has been used extensively in email spam [35] detection and link spam detection in web pages [36], [37], [38]. Over the last decade, researchers have been working on *deceptive opinion spam*.

Jindal et al. [3] first introduced *deceptive opinion spam* problem as a widespread phenomenon and showed that it is different from other traditional spam activities. They built their ground truth dataset by considering the duplicate reviews as spam reviews and the rest as nonspam reviews. They extracted features related to review, product and reviewer, and trained a Logistic Regression model on these features to find fraudulent reviews on Amazon. Wu et al. [39] claimed that deleting dishonest reviews will distort the popularity significantly. They leveraged this idea to detect deceptive opinion spam in the absence of ground truth data. Both of these heuristic evaluation approaches are not necessarily true and thorough.

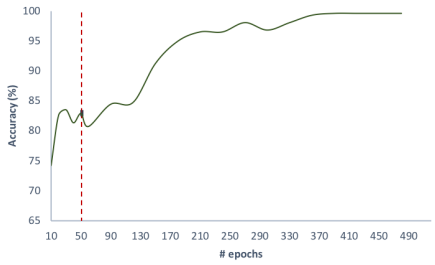
Yoo et al. [19] instructed a group of tourism marketing students to write a hotel review from the perspective of a hotel manager. They gathered 40 truthful and 42 deceptive hotel reviews and found that truthful and deceptive reviews have different lexical complexity. Ott et al. [4] created a much larger dataset of 800 opinions by crowdsourcing[3] the job of writing fraudulent reviews for existing businesses. They combined work from psychology and computational linguistics to develop and compare three[4] approaches for detecting deceptive opinion spam. On a similar dataset, Feng et al. [33] trained Support Vector Machine model based on syntactic stylometry features for deception detection. Li et al. [18] also combined ground truth dataset created by Ott et al. [4] with their employee (domain-expert) generated deceptive reviews to build a feature-based additive model for exploring the general rule for deceptive opinion spam detection. Rahman

[3]They used Amazon Mechanical Turk

[4]Genre identification, psycholinguistic deception detection, and text categorization.

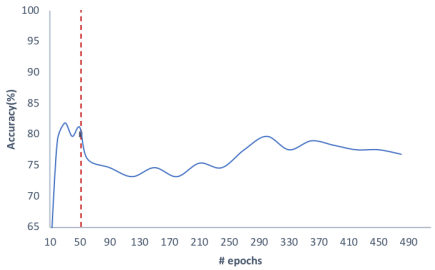


(a) Accuracy of FakeGAN (Discriminator D) at each step by feeding the testing dataset to D. While minimizing cross entropy method for pre-training D converges and reaches accuracy at $\sim 82\%$, adversarial training phase boosts the accuracy to $\sim 89\%$.

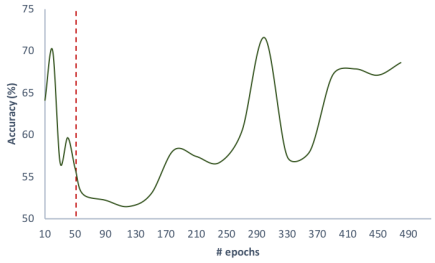


(b) Accuracy of D' at each step by feeding the testing dataset and generated samples by G to D'. Similar to figure 3a, this plot shows that D' converged after 450 steps resulting in the convergence of FakeGAN.

Fig. 3: The accuracy of D and D' on the test dataset over epochs. The vertical dashed line shows the beginning of adversarial training.



(a) The accuracy of D fluctuates around 77% in constrast to the stabilization at 89.1% in Figure 3a (with values g=1 and d=6)



(b) Accuracy of D'. Unlike in Figure 3b, this plot shows that D' is not stable.

Fig. 4: The accuracy of D and D' on the test dataset over epochs while both g and d are one.

et al. [40] developed a system to detect venues that are targets of deceptive opinions. Although, this easies the identification of deceptive reviews considerable effort is still involved in identifying the actual deceptive reviews. In almost all these works, the size of the dataset limits the proposed model to reach its real capacity.

To alleviate these issues with the ground truth, we use a Generative adversarial network, which is more an unsupervised learning method rather than supervised. We start with an existing dataset and use the generator model to create necessary reviews to strengthen the classifier (discriminator).

V. Future work

Contrary to the popular belief that supervised learning techniques are superior to unsupervised techniques, the accuracy of FakeGAN, a semi-supervised learning technique is comparable to the state-of-the-art supervised techniques on the same dataset. We believe that this is a preliminary step which we plan to extend by trying different architectures like Conditional GAN [41] and better hyper-tuning.

VI. Conclusion

In this paper, we propose FakeGAN, a technique to detect deceptive reviews using Generative Adversarial Networks (GAN). To the best of our knowledge, this is the first work to leverage GANs and semi-supervised learning methods to identify deceptive reviews. Our evaluation using a dataset of 800 reviews from 20 Chicago hotels of TripAdvisor shows that FakeGAN with an accuracy of 89.1% performed on par with the state-of-the-art models. We believe that FakeGAN demonstrates a good first step towards using GAN for text classification tasks, specifically those requiring very large ground truth datasets.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. This material is based on research sponsored by the Office of Naval Research under grant numbers N00014-15-1-2948, N00014-17-1-2011 and by DARPA under agreement number FA8750-15-2-0084. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. This work is also sponsored by a gift from Google's Anti-Abuse group. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

REFERENCES

[1] M. marketing research company, "Seven in 10 americans seek out opinion before making purchases," http://www.mintel.com/press-centre/social-and-lifestyle/seven-in-10-americans-seek-out-opinions-before-making-purchases, 2015.

[2] M. Anderson and J. Magruder, "Learning from the crowd: Regression discontinuity estimates of the effects of an online review database," *The Economic Journal*, vol. 122, no. 563, pp. 957–989, 2012.

[3] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proceedings of the 2008 International Conference on Web Search and Data Mining*, ser. WSDM '08. New York, NY, USA: ACM, 2008, pp. 219–230. [Online]. Available: http://doi.acm.org/10.1145/1341531.1341560

[4] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 309–319.

[5] B. Technology, "Yelp admits a quarter of submitted reviews could be fake," September 2013, http://www.bbc.com/news/technology-24299742.

[6] M. Luca and G. Zervas, "Fake it till you make it: Reputation, competition, and yelp review fraud," *Management Science*, 2016.

[7] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2011, pp. 151–161.

[8] R. Socher, E. H. Huang, J. Pennin, C. D. Manning, and A. Y. Ng, "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection," in *Advances in neural information processing systems*, 2011, pp. 801–809.

[9] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts *et al.*, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, vol. 1631, 2013, p. 1642.

[10] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[11] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification." in *AAAI*, vol. 333, 2015, pp. 2267–2273.

[12] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[13] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[15] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[16] K. Ehsani, R. Mottaghi, and A. Farhadi, "Segan: Segmenting and generating the invisible," *arXiv preprint arXiv:1703.10239*, 2017.

[17] E. L. Denton, S. Chintala, R. Fergus *et al.*, "Deep generative image models using a laplacian pyramid of adversarial networks," in *Advances in neural information processing systems*, 2015, pp. 1486–1494.

[18] J. Li, M. Ott, C. Cardie, and E. H. Hovy, "Towards a general rule for identifying deceptive opinion spam." in *ACL (1)*. Citeseer, 2014, pp. 1566–1576.

[19] K.-H. Yoo and U. Gretzel, "Comparison of deceptive and truthful travel reviews," *Information and communication technologies in tourism 2009*, pp. 37–47, 2009.

[20] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, "Unrolled generative adversarial networks," *arXiv preprint arXiv:1611.02163*, 2016.

[21] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: sequence generative adversarial nets with policy gradient," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[22] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[23] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.

[24] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[25] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[26] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] X. Zhang and Y. LeCun, "Text understanding from scratch," *arXiv preprint arXiv:1502.01710*, 2015.

[30] W. Y. Wang, ""liar, liar pants on fire": A new benchmark dataset for fake news detection," *arXiv preprint arXiv:1705.00648*, 2017.

[31] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[32] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[33] S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, 2012, pp. 171–175.

[34] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[35] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural networks*, vol. 10, no. 5, pp. 1048–1054, 1999.

[36] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen, "Combating web spam with trustrank," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004, pp. 576–587.

[37] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, "Detecting spam web pages through content analysis," in *Proceedings of the 15th international conference on World Wide Web*. ACM, 2006, pp. 83–92.

[38] Z. Gyongyi and H. Garcia-Molina, "Web spam taxonomy," in *First international workshop on adversarial information retrieval on the web (AIRWeb 2005)*, 2005.

[39] G. Wu, D. Greene, B. Smyth, and P. Cunningham, "Distortion as a validation criterion in the identification of suspicious reviews," in *Proceedings of the First Workshop on Social Media Analytics*. ACM, 2010, pp. 10–13.

[40] M. Rahman, B. Carbunar, J. Ballesteros, G. Burri, D. Horng *et al.*, "Turning the tide: Curbing deceptive yelp behaviors." in *SDM*. SIAM, 2014, pp. 244–252.

[41] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR*, vol. abs/1411.1784, 2014. [Online]. Available: http://arxiv.org/abs/1411.1784