Claim Verification using a Multi-GAN based Model

Amartya Hatua
University of Houston
4800 Calhoun Rd, Houston, TX 77004
amartyahatua@gmail.com

Arjun Mukherjee
University of Houston
4800 Calhoun Rd, Houston, TX 77004
arjun@uh.edu

Rakesh M. Verma
University of Houston
4800 Calhoun Rd, Houston, TX 77004
rmverma@cs.uh.edu

Abstract

This article describes research on claim verification carried out using a multiple GAN-based model. The proposed model consists of three pairs of generators and discriminators. The generator and discriminator pairs are responsible for generating synthetic data for supported and refuted claims and claim labels. A theoretical discussion about the proposed model is provided to validate the equilibrium state of the model. The proposed model is applied to the FEVER dataset, and a pre-trained language model is used for the input text data. The synthetically generated data helps to gain information that improves classification performance over state of the art baselines. The respective F1 scores after applying the proposed method on FEVER 1.0 and FEVER 2.0 datasets are 0.65±0.018 and 0.65±0.051.

1 Introduction

Misleading claims and news are becoming pervasive in our lives. Sometimes these are extremely difficult to identify. As a result, they can cause serious problems. This makes the research on claim verification essential. Fake news can be broadly classified into three categories (Rubin et al., 2015): i) Serious fabrications (uncovered in mainstream or participant media, yellow press or tabloids); ii) Large-scale hoaxes; and iii) Humorous fakes (news satire, parody, game shows). To solve this problem, research on this subject has evolved from knowledge-base oriented methods to sophisticated deep learning-based techniques.

Related Work

In (Mihalcea and Strapparava, 2009), the authors used natural language processing (NLP) techniques to detect fake news. They used tokenization and stemming for preprocessing the data and applied Naıve Bayes and Support Vector Machine (SVM) algorithms for classification. In recent research, the linguistic style and text source are considered the most critical factors to decide the genuineness of a fact or claim (Rashkin et al., 2017), (Baly et al., 2018), (Pérez-Rosas et al., 2017).

Sometimes multiple sources of particular claims are used as external resources for claim verification. In (Rashkin et al., 2017), researchers compared the linguistic characteristics of real news with satire, hoaxes, and propaganda. They presented a case study based on the data collected by PolitiFact.com, where they used Glove for embedding, and Long Short Term Memory (LSTM) for prediction. To improve their result, they concatenate the Linguistic Inquiry and Word Count (LIWC) features (Pennebaker et al., 2001) with LSTM output vectors before the activation layer.

LIWC features have played a vital role in claim verification research. LIWC extracts essential words that are part of psycho-linguistic categories and help in content analysis according to (Krippendorff, 2018; Neuendorf and Kumar, 2015). Their research work was extended by Kashyap et al. (Popat et al., 2018), who proposed an end-to-end framework for credibility analysis. This framework is capable of aggregating information from external evidence articles, the language of these articles, and the trustworthiness of their sources. It also generates informative features for user-comprehensible explanations (Popat et al., 2018).

Using external information sources is an effective technique for claim verification, e.g., researchers in (Pochampally et al., 2014), (Pasternack and Roth, 2011), (Ge et al., 2013), (Li et al., 2014), and (Wan et al., 2016) used external sources for similar types of tasks. Ravali et al. proposed a novel method based on correlations between different sources of news in (Pochampally et al., 2014). To find the correlation between sources, joint precision and joint recall are used.

Proceedings of Recent Advances in Natural Language Processing, pages 494–503
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_056

Jeff Pasternack et al. introduced a generalized fact-finding framework in (Pasternack and Roth, 2011) to resolve conflicting claims. Similarly, (Ge et al., 2013), (Li et al., 2014), (Wan et al., 2016) also used potentially inconsistent sources and information to verify facts and claims. Liang Ge et al. (Ge et al., 2013) proposed a procedure that calculates the degree of information consistency, identifies the underlying reason(s) for any inconsistencies, and calculates a consistent score for each item. In (Li et al., 2014), researchers proposed an optimization framework in which truths and reliable sources are considered as two sets of unknown variables, and the framework aims to minimize the deviation between the truths and the multi-source observations. A generalized algorithm called TruthFinder is proposed in (Wan et al., 2016), which utilizes the information of different related websites to perform fact-checking.

In recent research on this topic, deep learning techniques are becoming popular. In (Choudhary and Arora, 2020), a sequential neural model is proposed, which helps to identify syntactic, grammatical, sentimental, and readability features for fake news detection. Yang et al. (Yang et al., 2018) proposed text and Image information based Convolution Neural Network (TI-CNN), which uses both text and images as evidence for fact-checking. In this model, CNN is used for feature extraction from both text and images.

Recently, the FEVER dataset has gained a lot of traction (Thorne et al., 2018b), (Thorne and Vlachos, 2019), (Thorne et al., 2019). Hence, we use FEVER for claim verification. In earlier research with FEVER, most researchers followed a pipeline suggested by the baseline model (Thorne et al., 2018a), which consists of three sequential phases. The phases are: identifying relevant Wiki articles, extracting the appropriate supporting sentences, and determining the truthfulness of the claim. Earlier researchers implemented the Wiki article phase by Wikipedia API, token matching techniques and the AllenNLP framework (Gardner et al., 2017). For sentence selection, most earlier researchers have used TF-IDF, sequence matching neural network, and some ranking methods. The classification task is done using a TF-IDF approach in the base model, however later on neural network models, natural language inference models, and deep learning models were used.

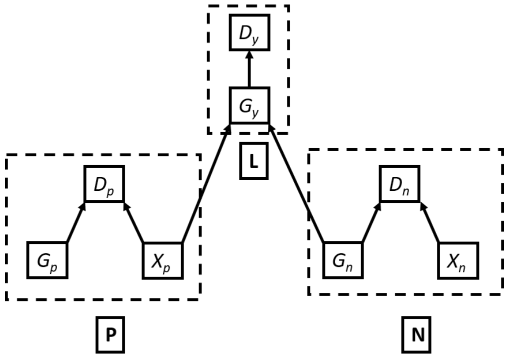


Figure 1: Schematic diagram of proposed model

Here, a GAN (Goodfellow et al., 2014) based method is proposed for claim verification. This model is inspired by two GAN based Positive Unlabeled (PU) learning models such as GenPU (Hou et al., 2017) and Yang et al. (Yang et al., 2020). Fig. 1 shows the proposed model. This model has three subunits P, N, and L. Each subunit consists of a generator (G_x) and discriminator (D_x) pair. Subunit P and N are responsible for generating positive and negative synthetic data; subunit L is responsible for binary class label generation of the synthetically generated data. Subunit P and N have positive (X_p) and negative (X_n) input data. The positive data consists of supported claims and respective evidence, while the negative data consists of refuted claims and respective evidence.

This model uses three generators (G_p, G_n, G_y) and three discriminators (D_p, D_n, D_y). G_p is responsible for generating positive claims and D_p discriminates between original and synthetically generated positive claims. G_n and D_n are responsible for similar functions for negative claims. Gy and Dy get the data generated by G_p and G_n and generate a class label (0/1) and D_y is the discriminator for G_y.

2 Proposed Methodology

As described above, three GAN units are used. These units are responsible for generating positive samples Equation 1, negative samples Equation 2 and class labels Equation 3. Algorithm 1 details the training of the generators and discriminators.

$$\min_{G_p} \max_{D_p} V(D, G) = \mathbb{E}_{x \sim p_p(x)} log(D_p(x)) + \mathbb{E}_{z \sim p_z(z)} log(1 - D_p(G_p(z))) \tag{1}$$

$$\min_{G_n} \max_{D_n} V(D,G) = \mathbb{E}_{x \sim p_n(x)}\, log(D_n(x)) + \mathbb{E}_{z \sim p_z(z)}\, log(1 - D_n(G_n(z))) \tag{2}$$

$$\min_{G_p, G_n, G_y} \max_{D_y} V(D,G) = \mathbb{E}_{x \sim p(x)}\, log(D_y(x)) + \pi_p\, \mathbb{E}_{z \sim p_z(z)}\, log(1 - D_y(G_y(G_p(z)))) + \pi_n\, \mathbb{E}_{z \sim p_z(z)}\, log(1 - D_y(G_y(G_n(z)))) \tag{3}$$

Algorithm 1 Training Algorithm

for training iterations **do**
  # **update** discriminator networks #
  sample mini-batch of noise examples $\{z^i\}_{i=1}^m$ from noise prior $p_z(z)$
  sample mini-batch of positive examples $\{x_p\}_{i=1}^m$ from noise prior $p_p(x)$
  sample mini-batch of negative examples $\{x_n\}_{i=1}^m$ from noise prior $p_n(x)$
  sample mini-batch of examples $\{x\}_{i=1}^m$ from noise prior $p(x)$
  update the positive discriminator D_p by ascending its stochastic gradient:$\nabla_{\theta_{D_p}} \frac{1}{m} \sum_{i=1}^m \pi_p [log(D_p(x_p^i)) + log(1 - D_p(G_p(z^i)))]$
  update the negative discriminator D_n by ascending its stochastic gradient:$\nabla_{\theta_{D_n}} \frac{1}{m} \sum_{i=1}^m \pi_n [log(D_n(x_n^i)) + log(1 - D_n(G_n(z^i)))]$
  update the discriminator D_y by ascending its stochastic gradient:$\nabla_{\theta_{D_y}} \frac{1}{m} \sum_{i=1}^m \pi_p [log(D_y(x^i)) + \pi_p log(1 - D_y(G_y(G_p(z))) + \pi_n log(1 - D_y(G_y(G_n(z)))]$
  # **update** generator networks #
  sample mini-batch of noise examples $\{z^i\}_{i=1}^m$from noise prior $p(z)$
  update the positive generator G_p by descending its stochastic gradient: $\nabla_{\theta_{G_p}} \frac{1}{m} \sum_{i=1}^m \pi_p [-log(D_p(G_p(z^i)) - log(D_y(G_p(z^i))]$
  update the negative generator G_p by descending its stochastic gradient: $\nabla_{\theta_{G_n}} \frac{1}{m} \sum_{i=1}^m \pi_n [-log(D_n(G_n(z^i)) - log(D_y(G_n(z^i))]$
  update the class label generator G_y by descending its stochastic gradient: $\nabla_{\theta_{G_y}} \frac{1}{m} \sum_{i=1}^m [-\pi_p log(D_y(G_p(z^i)) - \pi_n log(D_y(G_n(z^i))]$
end for
return G_y

The proposed model can handle only supported and refuted claims. D_y is trained with both supported and refuted claims, while D_p and D_n are trained with only supported and refuted claims separately. Hence, D_y is a more powerful discriminator compared to D_p and D_n. There is a possibility that D_p or D_n will assign some sentences generated by G_p and G_n wrongly. As D_y has the global view of both supported and refuted claims, it is better able to classify them. Consider a situation: G_p generates Y_p (a synthetic positive claim). In the next step, Y_p is the input to G_y, and G_y is generating 1 (positive class label). The output of G_y and input of G_p is the input to the discriminator state (D_y). If D_y classifies Y_p as real, then no penalty is incurred by G_y and G_p otherwise both G_p and G_y are penalized. Consider another situation, where G_y generates 0 (negative class label) for an input of Y_p and D_y also classifies the Y_p as fake, then a penalty will be added to G_p, not G_y. So D_y is acting as a global discriminator. Equation 4 is the loss function for the generator G_y, where π_p and π_n are the probabilities of positive and negative claims in the dataset.

$$L(y) = \pi_p[D_y(G_p(z)) log(D_y(G_y(G_p(z)))) + (1 - D_y(G_p(z))) log(1 - D_y(G_p(z)))] + \pi_n[D_y(G_n(z)) log(D_y(G_y(G_n(z)))) + (1 - D_y(G_n(z))) log(1 - D_y(G_n(z)))] \tag{4}$$

For a GAN, achieving equilibrium is very important. In the present context, to find the equilibrium condition, first, we need to find the optimal conditions for discriminators. Using the optimal conditions of the discriminators, the minimization conditions for the generator can be obtained. Considering the generators (G_p, G_n, G_y) are fixed, and π_p and π_n are the probabilities of positive and negative claims in the dataset, at the equilibrium condition the distribution of positive generated data ($p_{gp}(x)$) and negative generated data ($p_{gn}(x)$) will follow the Equations 5 and 6, where $p_p(x)$ and $p_n(x)$ are the positive and negative class probability distributions.

$$p_{gp}(x) = p_p(x) \tag{5}$$

$$p_{gn}(x) = p_n(x) \tag{6}$$

The optimal discriminator functions $D_p^*(x)$, $D_n^*(x)$, $D_y^*(x)$ can be derived by differentiating Equations 1, 2 and 3 (Hatua et al., 2021a).

$$D_P^*(x) = \frac{p_p(x)}{p_p(x) + p_{gp}(x)} \tag{7}$$

$$D_n^*(x) = \frac{p_n(x)}{p_n(x) + p_{gn}(x)} \tag{8}$$

$$\min_{G_p, G_n, G_y} \max_{D_y} V(D_y^*, G) =$$
$$log\left(\frac{p(x)}{p(x) + \pi_p p_{gp}(x) + \pi_n p_{gn}(x)}\right) +$$
$$\pi_p log\left(\frac{\pi_p p_{gp}(x) + \pi_n p_{gn}(x)}{p(x) + \pi_p p_{gp}(x) + \pi_n p_{gn}(x)}\right) + \quad (9)$$
$$\pi_n log\left(\frac{\pi_p p_{gp}(x) + \pi_n p_{gn}(x)}{p(x) + \pi_p p_{gp}(x) + \pi_n p_{gn}(x)}\right)$$

Using Jensen–Shannon divergence (JSD) (Fuglede and Topsoe, 2004), we can show that the argmin generators are achieved when the following conditions are satisfied:

$$p_p(x) = p_{gp}(x) \quad (10)$$

$$p_n(x) = p_{gn}(x) \quad (11)$$

$$p_y(x) = \pi_p p_{gp}(x) + \pi_n p_{gn}(x) \quad (12)$$

3 Data

FEVER is a publicly available dataset for claim verification with three types of claims: i) supported, ii) refuted, iii) Not Enough Information (NEI). For every supported and refuted claim, there is supporting/refuting evidence, while for the NEI class there is no evidence. All evidence provided in the FEVER dataset is collected from Wikipedia. In most cases, the first few lines of a particular Wikipedia page are taken in FEVER dataset as the evidence. Table 1 shows two examples of claim, evidence pairs and their class labels. For the experiments, we used only Supported and Refuted claims.

FEVER training subset has 80,035 Supported claims, 29,775 Refuted claims, and 35,639 NEI claims. The FEVER 1.0 validation set and test set have 3,333 Supported claims, 3,333 Refuted claims, and 3,333 NEI claims respectively. FEVER 2.0 has 391 Supported claims, 396 Refuted claims, and 387 NEI claims respectively. For the experiments, we used only Supported and Refuted claims.

4 Experiments

The workflow of the experiment is given in Fig 2. In the first phase, data is preprocessed as described in Section 4.1. This preprocessed data is used as input to the proposed model for training. The Supported claim, evidence pairs are input to the positive synthetic data generator subunit, and the Refuted claim, evidence pairs are input to the negative synthetic data generator subunit. Once the proposed model is trained with the preprocessed data, the model is used for the testing phase using the test dataset. Finally, the model's performance is compared with the results of other standard methods and SOTA models. The steps of the experiments are detailed below.

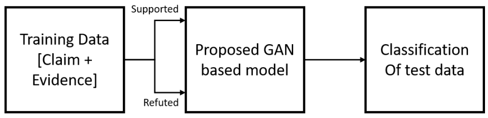


Figure 2: Workflow of the experiment

4.1 Data preprocessing

For this experiment, only 'Supported' and 'Refuted' claims are considered from the training dataset. In the training dataset, every claim has one or more statements (evidence). For a particular claim, its corresponding statements are concatenated separately. For example, suppose claim (C) evidence (E) and label $(L) are : [C; E :< e_1, e_2, e_3 >, L]$. The input data format for subsequent processes will be: $x = [< C; e_1, L >, < C; e_2, L >, < C; e_3, L >]$.

4.2 GAN Implementation

The implementation of GAN is the central part of this research. Two types of GANs are implemented: text generating GAN and binary class label generating GAN. The text generating GANs generate synthetic text data for supported and refuted claims. The binary class label generating GAN generates the binary class label for each generated claim. To implement text generating GAN, we use LaTextGAN (Donahue and Rumshisky, 2018). LaTextGAN follows two phases for the implementation. During the first phase, it creates an encoded space, and in the second phase, it follows the traditional GAN (Goodfellow et al., 2014) implementation steps and generates synthetic data in the encoded space. Finally, the synthetically generated data is decoded into normal text data. On the other hand, the implementation of binary labels generating GAN is similar to the implementation of the traditional GAN (Goodfellow et al., 2014). The evidence for the synthetically generated sentences are selected from the Wikipedia database (Thorne et al., 2018b) using cosine similarity (Huang et al., 2008).

Claim: Tetris has sold millions of physical copies. **Evidence:** It was announced that Tetris has sold more than 170 million copies, approximately 70 physical copies and ... **Label:** True
Claim: Andy Roddick lost 5 Master Series between 2002 and 2010. **Evidence:** Roddick was ranked in the top 10 for nine consecutive years between 2002 and 2010, and won five Masters Series in that period. **Label:** False

Table 1: Two claim, evidence pairs from FEVER

In this case we have selected one evidence for every synthetically generated sentence. The synthetically generated data and the evidence are concatenated and processes following the steps mentioned in Section 4.1.

4.3 New GenPU Based Baselines

These baselines are inspired from the GenPU. To explore further we have modified GenPU in two variants: Inverted GenPU and Symmetric GenPU. In case of Inverted GenPU the value functions for the positive and negative text generating GAN are exchanged. Hence the respective value functions become the equations mentioned in Equation 13, 14 and 15.

$$D_n^* = \operatorname*{argmax}_{D_n} \mathbb{E}_{x \sim p_p(x)} log(D_n(x)) + \mathbb{E}_{z \sim p_z(z)} log(1 - D_u(G_n(z))) \tag{13}$$

$$\min_{G_p} \max_{D_p} V(D,G) = -\mathbb{E}_{x \sim p_p(x)} log(D_n^*(x)) - \mathbb{E}_{z \sim p_z(z)} log(1 - D_n^*(G_n(z))) \tag{14}$$

$$\min_{G_n} \max_{D_n} V(D,G) = \mathbb{E}_{x \sim p_p(x)} log(D_p(x)) + \mathbb{E}_{z \sim p_z(z)} log(1 - D_p(G_p(z))) \tag{15}$$

In Symmetric GenPU the equations for both the value functions are same. The value functions for Symmetric GenPU are presented in Equation 16 and 17.

$$\min_{G_p} \max_{D_p} V(D,G) = \mathbb{E}_{x \sim p_p(x)} log(D_p(x)) + \mathbb{E}_{z \sim p_z(z)} log(1 - D_p(G_p(z))) \tag{16}$$

$$\min_{G_n} \max_{D_n} V(D,G) = \mathbb{E}_{x \sim p_p(x)} log(D_p(x)) + \mathbb{E}_{z \sim p_z(z)} log(1 - D_p(G_p(z))) \tag{17}$$

4.4 Other methods

The performance of the proposed method and new baselines is compared with other GAN based methods and classifiers. The GAN (LeakGAN (Guo et al., 2017) and LaTextGAN (Donahue and Rumshisky, 2018)) based models generate synthetic data and the synthetically generated data is added to the original dataset and it helps to create an extended feature space of the FEVER dataset and gives leverage to new features. This synthetically generated data is further classified using positive-unlabeled (PU) learning which considers supported facts as positive class and are added to the existing training dataset. Finally, this extended dataset is used for the training process. The synthetic data is generated using LeakGAN and LaTextGAN separately and two different sets of results are collected to compare the performance. The result of this method (Hatua et al., 2021b) for both the datasets is compared with the proposed method in Table 2, and Table 3. Other baselines include deep learning and machine learning based classification methods such as: BERT based classifier (Devlin et al., 2018), Graph Convolution Network (GCN) (Scarselli et al., 2008), Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), Convolution Neural Network (CNN) (Lawrence et al., 1997), Support Vector Machine (SVM) (Drucker et al., 1996), Naive Bayes (Lewis, 1998), Random forest (Pal, 2005), and Stochastic Gradient Descent (SGD) (Friedman, 2002).

To implement BERT based classifier Huggingface BERT (Devlin et al., 2018) pretrained transformer is used as tokenizer for the training, validation and testing dataset. The vocabulary size of the pretrained model is 30522 and the size of the hidden layer is 768. Later the pre-tuned model is fine tuned to classify the claims. In GCN, point wise mutual information between words is calculated to generate the graph. To implement the CNN five kernels of sizes 2, 3, 4, 5 and 6 are used. For LSTM the input data is encoded using GloVe (Pennington et al., 2014). The learning rate and batch size for GCN, CCN and LSTM are 0.001, 64 respectively. The Random forest is equipped with 1000 trees and entropy is used as supported criteria for the information gain. The SGD model utilizes hinge loss and L2 penalty. The deep learning models are implemented using PyTorch (Paszke et al., 2019), and the Scikit learn library (Pedregosa et al., 2011) is used for machine learning models.

5 Results

All models are trained with the FEVER training dataset and tested with FEVER 1.0 and FEVER 2.0 test dataset. In Tables 2, and 3 detailed results for each of the models are presented. Each experiment is repeated five times. The result for FEVER 1.0 is also compared with previous research work by Yang et al. (Yang et al., 2020).

Table 2: Result of FEVER 1.0

	FEVER 1.0 Dataset		
Classifiers	Precision	Recall	F1 Score
BERT	0.45 ± 0.011	0.44 ± 0.010	0.44 ± 0.009
Leak GAN	0.65 ± 0.003	0.64 ± 0.006	0.64 ± 0.003
LaTextGAN	0.41 ± 0.008	0.36 ± 0.016	0.38 ± 0.009
GCN	0.45 ± 0.015	0.44 ± 0.013	0.44 ± 0.013
SVM	0.53 ± 0.013	0.42 ± 0.013	0.46 ± 0.013
Naive Bayes	0.41 ± 0.016	0.34 ± 0.014	0.37 ± 0.015
RF	0.33 ± 0.011	0.33 ± 0.010	0.33 ± 0.011
SGD	0.31 ± 0.023	0.22 ± 0.022	0.25 ± 0.023
LSTM	0.45 ± 0.003	0.42 ± 0.004	0.43 ± 0.004
CNN	0.46 ± 0.012	0.44 ± 0.011	0.44 ± 0.012
Inverted GenPU	0.52 ± 0.013	0.71 ± 0.023	0.60 ± 0.018
Symmetric GenPU	0.33 ± 0.015	0.54 ± 0.02	0.40 ± 0.016
Proposed Method	0.50 ± 0.016	0.93 ± 0.018	0.65 ± 0.018
Yang et al. result	0.61	0.58	0.60

In Tables 2, and 3 we see that the F1 score for the proposed method is better than the new baselines and previous research.

Table 3: Result of FEVER 2.0

	FEVER 2.0 Dataset		
Classifiers	Precision	Recall	F1 Score
BERT	0.46 ± 0.013	0.44 ± 0.014	0.44 ± 0.013
Leak GAN	0.52 ± 0.023	0.51 ± 0.019	0.51 ± 0.021
LaTextGAN	0.42 ± 0.02	0.39 ± 0.019	0.40 ± 0.019
GCN	0.43 ± 0.023	0.39 ± 0.013	0.40 ± 0.016
SVM	0.40 ± 0.019	0.37 ± 0.022	0.38 ± 0.019
Naive Bayes	0.33 ± 0.030	0.22 ± 0.023	0.26 ± 0.025
Random forest	0.33 ± 0.014	0.26 ± 0.017	0.29 ± 0.015
SGD	0.30 ± 0.025	0.22 ± 0.029	0.25 ± 0.027
LSTM	0.43 ± 0.028	0.40 ± 0.039	0.41 ± 0.032
CNN	0.41 ± 0.021	0.38 ± 0.011	0.39 ± 0.018
Inverted GenPU	0.58 ± 0.024	0.71 ± 0.022	0.63 ± 0.012
Symmetric GenPU	0.41 ± 0.016	0.55 ± 0.011	0.46 ± 0.013
Proposed method	0.49 ± 0.061	0.97 ± 0.041	0.65 ± 0.051

The gradual change of precision, recall, and F1 score for the FEVER 1.0 and FEVER 2.0 is presented in Fig. 3, and Fig. 4. Moreover, to visualize the distribution of original and synthetic data, t-SNE plots of the positive and negative generated data are shown in Figures 5, and 6. The perplexity of the t-SNE plot is 30, and the learning rate is 120. It can be observed that the distribution of synthetically

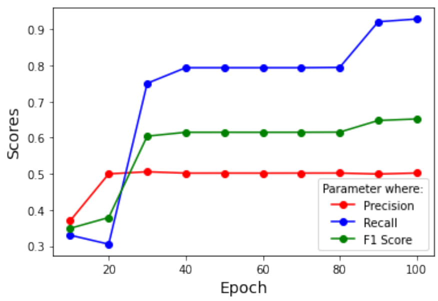


Figure 3: Precision, Recall and F1 Score for FEVER 1.0 Dataset

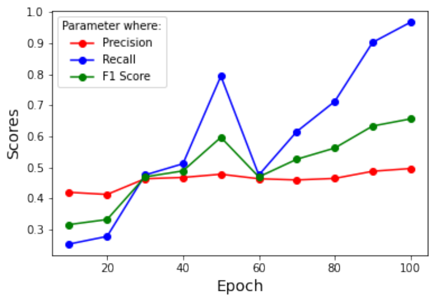
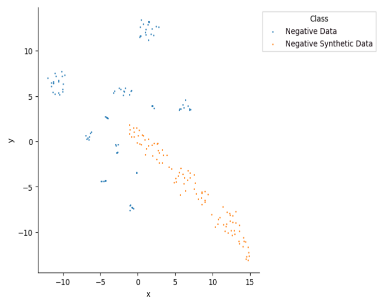
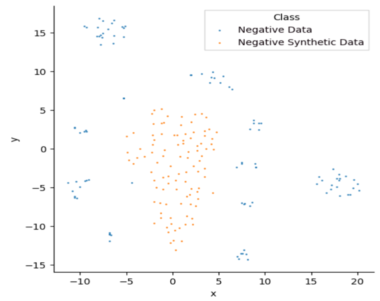


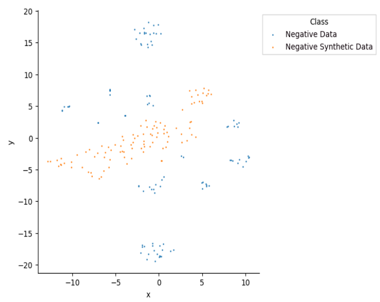
Figure 4: Precision, Recall and F1 Score for FEVER 2.0 Dataset



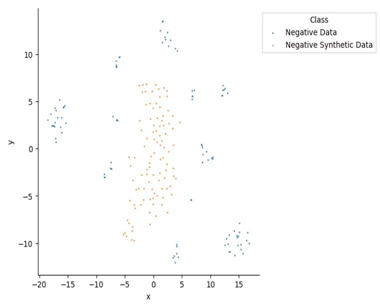
(a) Epoch = 25



(b) Epoch = 50

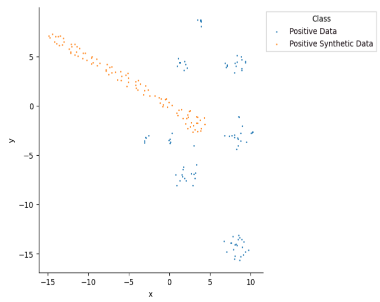


(c) Epoch = 75

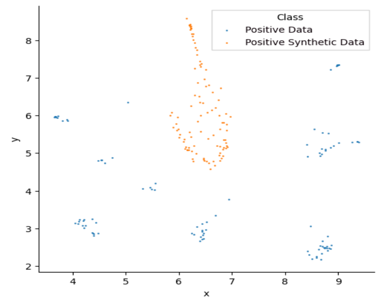


(d) Epoch = 100

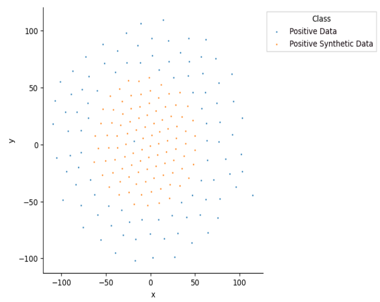
Figure 5: t-SNE Plot of original and synthetic data for negative class



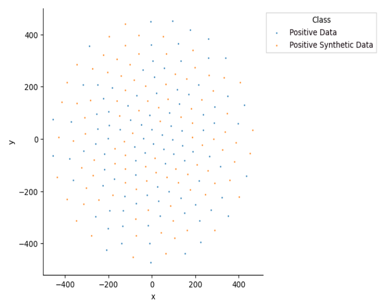
(a) Epoch = 25



(b) Epoch = 50



(c) Epoch = 75



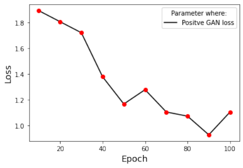
(d) Epoch = 100

Figure 6: t-SNE Plot of original and synthetic data for positive class

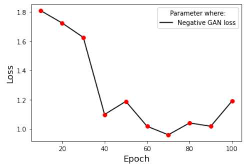
generated positive data is very similar to that of original positive text data, while the distribution of the negative synthetic data is similar to the original negative text data. The positive synthetic data is much more similar to the positive text data compared to the similarity between negative synthetic data and negative text data.

The proposed GAN based model starts with some random values and tries to generate synthetic data, which helps to achieve a better F1 score. In the training process, after every epoch, we have calculated the F1 score for both the test datasets and observed a gradual improvement of the F1 score.

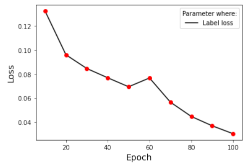
Fig. 7a, 7b, and 7c depicting the positive loss, negative loss and label generating loss. We can see the three losses are decreasing over epochs gradually,



(a) Positive

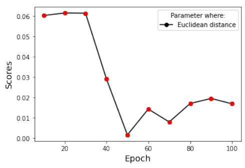


(b) Negative

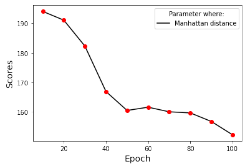


(c) Label

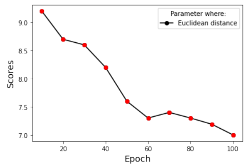
Figure 7: Different losses



(a) Cosine

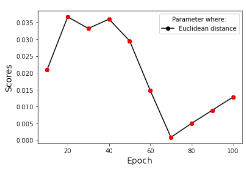


(b) Manhattan

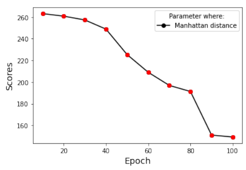


(c) Euclidean

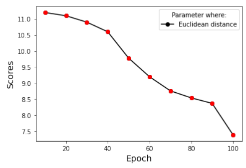
Figure 8: Similarity scores for positive data



(a) Cosine



(b) Manhattan



(c) Euclidean

Figure 9: Similarity scores for negative data

which also suggests that all the generator discriminator pairs are training to achieve the equilibrium state. To test the gradual progression of the synthetically generated data, we also measure the similarity scores between original (positive and negative) data and synthetic data (positive and negative) while training the model. It has been observed that for the generated data, the similarity score gradually improves over epochs, as shown in Fig. 8, and 9. To measure the similarity 20,000 synthetically generated data are randomly selected and Cosine similarity (Singhal et al., 2001), Manhattan distance (Sinwar and Kaushik, 2014), Euclidean distance (Aggarwal et al., 2001) are calculated.

6 Conclusion

We propose a multiple GAN-based model that employs the GAN's synthetic data generation capability to solve claim verification problems. The model generates synthetic data for supported, refuted claims and their class labels using three separate generator discriminator pairs. The synthetic data eventually helps in the fact-checking task for FEVER 1.0 and FEVER 2.0 test datasets. The results have shown that the proposed model starts with random data generation, and as the training progresses, it generates synthetic data similar to the original data.

Different statistical and analytical similarity metrics confirm that the similarity between original data and synthetically generated data increases as the training progresses. This gradual improvement of data quality shows the effectiveness of the model. The proposed model produces an F1 score of 0.65 $\pm$ 0.018 and 0.65 $\pm$ 0.051 for FEVER 1.0 and FEVER 2.0, respectively.

Dataset quality is a subtle issue, e.g., see (Verma et al., 2019; Verma and Marchette, 2019). In the future, this model can be extended to a multi-class classifier, and a similar set of experiments can be carried out on other publicly available standard datasets to test this proposed model's effectiveness across different datasets.

Acknowledgments

Research was supported in part by grants NSF 1838147, NSF 1433817, ARO W911NF-20-1-0254 and ONR N00014-19-S-F009. Verma is the founder of Everest Cyber Security and Analytics, Inc. The views and conclusions contained in this document are those of the authors and not of the sponsors. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding copyright notation(s) herein.

References

Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. 2001. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer.

Ramy Baly, Georgi Karadzhov, Dimitar Alexandrov, James Glass, and Preslav Nakov. 2018. Predicting factuality of reporting and bias of news media sources. *arXiv preprint arXiv:1810.01765*.

Anshika Choudhary and Anuja Arora. 2020. Linguistic feature based learning model for fake news detection and classification. *Expert Systems with Applications*, page 114171.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

David Donahue and Anna Rumshisky. 2018. Adversarial text generation without reinforcement learning. *arXiv preprint arXiv:1810.06640*.

Harris Drucker, Christopher J Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. 1996. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161.

Jerome H Friedman. 2002. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378.

Bent Fuglede and Flemming Topsoe. 2004. Jensen-shannon divergence and hilbert space embedding. In *International Symposium on Information Theory, 2004. ISIT 2004. Proceedings.*, page 31. IEEE.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform. *arXiv:1803.07640*.

Liang Ge, Jing Gao, Xiaoyi Li, and Aidong Zhang. 2013. Multi-source deep learning for information trustworthiness estimation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 766–774.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27:2672–2680.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2017. Long text generation via adversarial training with leaked information. *arXiv preprint arXiv:1709.08624*.

Amartya Hatua, Arjun Mukherjee, and Rakesh M Verma. 2021a. Claim verification using a multi-gan based model. *arXiv preprint arXiv:2103.08001*.

Amartya Hatua, Arjun Mukherjee, and Rakesh M Verma. 2021b. On the feasibility of using GANs for claim verification - experiments and analysis. In *ROMCIR*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ming Hou, Brahim Chaib-Draa, Chao Li, and Qibin Zhao. 2017. Generative adversarial positive-unlabelled learning. *arXiv preprint arXiv:1711.08054*.

Anna Huang et al. 2008. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, volume 4, pages 9–56.

Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. 1997. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113.

David D Lewis. 1998. Naive (bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning*, pages 4–15. Springer.

Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. 2014. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1187–1198.

Rada Mihalcea and Carlo Strapparava. 2009. The lie detector: Explorations in the automatic recognition of deceptive language. *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 309–312.

Kimberly A Neuendorf and Anup Kumar. 2015. Content analysis. *The international encyclopedia of political communication*, pages 1–10.

Mahesh Pal. 2005. Random forest classifier for remote sensing classification. *International journal of remote sensing*, 26(1):217–222.

Jeff Pasternack and Dan Roth. 2011. Making better informed trust decisions with generalized fact-finding. In *Twenty-Second International Joint Conference on Artificial Intelligence*. Citeseer.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch:

An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.

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

Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2017. Automatic detection of fake news. *arXiv preprint arXiv:1708.07104*.

Ravali Pochampally, Anish Das Sarma, Xin Luna Dong, Alexandra Meliou, and Divesh Srivastava. 2014. Fusing data with correlations. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 433–444.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2018. Declare: Debunking fake news and false claims using evidence-aware deep learning. *arXiv preprint arXiv:1809.06416*.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2931–2937.

Victoria L Rubin, Yimin Chen, and Nadia K Conroy. 2015. Deception detection for news: three types of fakes. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Amit Singhal et al. 2001. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43.

Deepak Sinwar and Rahul Kaushik. 2014. Study of euclidean and manhattan distance metrics using simple k-means clustering. *Int. J. Res. Appl. Sci. Eng. Technol*, 2(5):270–274.

James Thorne and Andreas Vlachos. 2019. Adversarial attacks against fact extraction and verification. *arXiv preprint arXiv:1903.05543*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018a. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018b. The fact extraction and verification (fever) shared task. *arXiv preprint arXiv:1811.10971*.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019. The fever2. 0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 1–6.

Rakesh M Verma and David J Marchette. 2019. *Cybersecurity Analytics*. Chapman and Hall/CRC.

Rakesh M. Verma, Victor Zeng, and Houtan Faridi. 2019. Data quality for security challenges: Case studies of phishing, malware and intrusion detection datasets. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 2605–2607, New York, NY, USA. Association for Computing Machinery.

Mengting Wan, Xiangyu Chen, Lance M. Kaplan, Jiawei Han, Jing Gao, and Bo Zhao. 2016. From truth discovery to trustworthy opinion discovery: An uncertainty-aware quantitative modeling approach. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1885–1894.

Fan Yang, Eduard Dragut, and Arjun Mukherjee. 2020. Claim verification under positive unlabeled learning. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*.

Yang Yang, Lei Zheng, Jiawei Zhang, Qingcai Cui, Zhoujun Li, and Philip S Yu. 2018. Ti-cnn: Convolutional neural networks for fake news detection. *arXiv preprint arXiv:1806.00749*.