Claim Verification under Positive Unlabeled Learning

Fan Yang*
University of Houston
Houston, USA
fyang11@uh.edu

Eduard Dragut
Temple University
Philadelphia, USA
edragut@temple.edu

Arjun Mukherjee
University of Houston
Houston, USA
arjun@cs.uh.edu

***Abstract*—We extend evidence-aware claim verification to the context of positive-unlabeled (PU) learning. Existing works assume the truth and the falsity of the claims are known for training and form the task as a supervised learning problem. However, this assumption underestimates the difficulty of collecting false claims; we argue that claim verification is more challenging in the absence of negative labels. We consider a more practical setting, where only a comparatively small number of true claims are labeled and more claims remain unlabeled. Thus, we formulate the claim verification task as a PU learning problem. We decouple learning representation of claim-evidence pair from PU learning and adopt a pre-trained universal language model to encode claim-evidence pairs. We further propose to use the generative adversarial network (GAN) to capture the latent alignment between encoded claim-evidence pair and the truthfulness. We leverage the verification as part of the GAN by extending previous GAN based PU learning. We show that the proposed model achieves the best performance with a small amount of labeled data and is robust to the truthfulness prior estimation. We conduct a thorough analysis of the model selection. The proposed approach performs the best under two practical scenarios: (i) the unlabeled data is more than the labeled data; (ii) and the unlabeled positive data is more than the unlabeled negative data.**

***Index Terms*—claim verification, positive unlabeled learning, generative adversarial network**

I. Introduction

Claim verification aims to verify the credibility of a claim in a document. It has important applications to prevent misinformation diffusion – falsehood often diffuses faster and broader than the truth [1]. Existing works formulate the task as a classification problem [2]–[4], assume that true claims and false claims are presented, and study the language style [5], the source of the claim [6], and the external evidence [7], [8] to verify a claim.

We explore a more practical and challenging setting, where only limited true claims are labeled and more claims remain unlabeled. We argue this setting benefits the real-world application in the sense that a false claim is difficult to collect and may evolve over time. We investigate the task using positive-unlabeled (PU) learning [9], [10]. We assume there is an evidence retrieval system providing external evidence for each claim and all evidence are credible to reflect facts. Thus, we further consider the claim-evidence pair as the basic format to present a claim for verification, as illustrated in Table I. To verify if a claim is true or false, a system must indicate whether the provided evidence supports or refutes the claim.

TABLE I: Examples of claim verification. When only true claims are labeled, we investigate how to predict the truthfulness by leveraging both the labeled and unlabeled claim-evidence pairs.

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

Recent works report good results on the evidence-aware claim verification [11], by leveraging textual entailment [12]–[14] to model the support and the refute. These works conduct experiments on crowdsourcing data [8] and require both the true and the false claims. They cannot deal with the situation where only positive data is labeled, and therefore cannot be extended to the positive-unlabeled setting. In addition, classic PU learning deals with single document instances, whereas we consider at least two documents, one claim and one evidence, for each instance. To our knowledge, we are the first to study the interaction of multiple sentences under the PU learning.

We suggest decoupling representation learning of claim-evidence pairs from PU learning. We propose to use a universal language model to transform textual claim and evidence into numerical vectors. The language model learns the representation of a document without specifying downstream tasks. In this work, we utilize BERT [15], because it captures relationship among multiple sentences. Previous work also shows state-of-the-art performance when using BERT to learn representations for supervised claim verification. Thus, we feed the textual claim-evidence pair into the pretrained BERT to get the representation as the first step, then apply PU learning model to verify a claim.

The focus of the paper is to develop a PU learning model for claim verification. PU-learning has been explored to detect deceptive content [16], [17]. Generally, identifying useful

*The work was done prior to joining Amazon.

IEEE/ACM ASONAM 2020, December 7-10, 2020
978-1-7281-1056-1/20/$31.00 © 2020 IEEE

negative data from the unlabeled examples to form ordinary supervised learning [10], [18] and regarding the unlabeled data as the less weighted negative class [9], [19] are two major approaches for PU-learning. The latter is further improved to consider the unlabeled data as a weighted combination of positive and negative class [20]–[23]. Generative adversarial network [24] (GAN) is increasingly applied to PU-learning due to its capacity of approximating distributions. GenPU gives better performance than previous PU-learning methods; it generates pseudo negative samples and trains a binary classifier [25].

Although GenPU theoretically captures the distribution of negative samples, we identify two challenges when extending GenPU to claim verification. First, generating pseudo samples may ignore the truthfulness of textual statements and be fooled by superficial lexicons. For example, in Table I, two claims are about different topics and merely share some stop words. It is difficult to generate good samples capturing the truthfulness. Second, the two-step pipeline of GenPU is not efficient, in the sense that one needs to build the classifier after training the GAN. If claim-evidence pairs are not well generated, the classifier is not reliable. Thus, instead of decoupling the GAN and the classifier, we propose to jointly train the classifier and the generator to recognize the truthfulness, and take the claim verification as the validation criteria. Inspired by works that extend GAN to semi-supervised learning [26]–[29], we aim to model the claim-truthfulness alignment. Specifically, we consider two types of the generation: (1) generate claim representation conditioned on the truthfulness and (2) generate the label as classification. We leverage two discriminators to guide the generation procedures. Since the classifier is within our model, we implicitly ease the target for generating good claim representations. As a result, the proposed model maintains the capacity of GenPU and addresses the aforementioned concerns.

We make the following contributions: (i) To the best of our knowledge, this is the first work to consider positive-unlabeled (PU) evidence-aware claim verification. We contend that PU evidence-aware claim verification is better suited in practice because false claims are difficult to collect.; (ii) We decouple representation learning and PU learning. We propose to use the pretrained language model, BERT, to build the vector of claim-evidence pair, and extend GenPU to capture the latent alignment between a claim and its truthfulness.; (iii) We give an extensive empirical study of PU evidence-aware claim verification. Our experiment suggests that the ratio between the unlabeled positive class and the unlabeled negative class significantly affect the selection of models. Specifically, our model performs well when the ratio is high and is robust when the truthfulness prior is unknown.

II. Related Work

We give a brief overview of related work on claim verification and positive-unlabeled learning to better position our work in this paper.

Claim Verification has been studied from a number of perspectives. In the absence of evidence (either true or false), linguistic features are widely considered to analyze the credibility of claims [5]. For example, true claims are expressed in objective and unbiased language, so assertive verbs, mitigating words, and discourse markers can reveal the credibility of claims [30]. Other works consider psycholinguistic features, part-of-speech tags, and syntax to reveal misleading content [31]–[33]. Some works consider external information, such as user interaction [5], [30] and the meta-data of a claim source [6]. Since our confidence in a piece of evidence is critical for claim verification, a line of work aims to establish confidence in the truth from multiple, possibly conflicting, sources [34]–[38]. Given the degree of reliability of each source, the confidence in the truth of evidence can be obtained via some aggregating scheme, e.g., fusion [39] or weighted combination [40]. Traditionally, evidence is stored as structured knowledge, e.g., *<subject, predicate, object>* triples, can be used to verify claims [41]–[43]. However, structured knowledge require a non-trivial processing pipeline to be extracted from text [44], which may delay claim verification.

Much of the latest work focuses on unstructured textual data. They first retrieve evidence from textual candidates and then proceed with the verification of a claim. An example in this space is FEVER, which takes wiki pages as potential evidence and constructs claims by crowdsoucing [8]. It has a three-step pipeline: identify relevant wiki articles, extract the sentences (supporting evidence) relevant to a claim, and determine if the evidence supports the claim. Textual entailment [12], [14], [45], [46] is applied to the last step. Intuitively, textual entailment discovers the relationship between a pair of sentences and benefits the task. Such work builds upon methods from decomposable attention [14] and enhanced sequential inference [13], for sentence retrieval and textual entailment [2], [3], [11]. A few other works adopt pre-trained language models and graph-based methods [47]. These approaches assume the presence of true and false claims alike, whereas we argue that labeling claims require significant labor and unlabeled claims are not well leveraged. Thus, we tackle the problem from the perspective of positive unlabeled learning.

Positive-Unlabeled Learning traditionally has two major approaches: identifying possible negative examples [10], [18] and regarding the unlabeled data as the less weighted negative class [9], [19]. Finding negative examples depends on some heuristic in general and weighting unlabeled data is computationally expensive. Both approaches suffer from a systematic estimation bias [20], [23]. Elkan et al [22] propose to consider the unlabeled data as a weighted combination of positive and negative data. Based on this idea, they propose an unbiased PU classifier [20], [21] and a non-negative risk estimator [23]. More recently, a new paradigm solves the PU task by generating pseudo negative samples [25]. They leverage generative adversarial network [24] and give a thorough theoretical analysis of the model, showing that is capable of learning both positive and negative data distributions at equilibrium. We propose to extend the deep generative model

from two perspectives: (i) design the classifier to be part of the GAN; (ii) use BERT to vectorize claim-evidence pairs and take the intermediate representation as the target.

III. Model

The full pipeline of our model includes BERT [15], a pre-trained language encoder that encodes the claim-evidence pair, and an enhanced generative deep network to capture the latent alignment between the vectorized representation and the truthfulness.

A. Task Formulation

Evidence-aware claim verification assumes that a claim is paired with one or more pieces of evidence [8]. Let X denote a collection of claim-evidence pairs sampled from $p(\mathbf{x})$. We further define that each sample $\mathbf{x}$ contains a claim $\mathbf{C}$ with a sequence of L words, $[w_0^c, \ldots, w_L^c]$, and evidence $\mathbf{E}$ with a sequence K of words, $[w_0^e, \ldots, w_K^e]$. The task is to find a function such that $y = f(x), y \in \{0, 1\}$, where $y = 0$ means false claim and $y = 1$ means true claim. Thus, claim verification is a binary classification problem if both true claims X_P and false claims X_N are available. Under the positive-unlabeled setting, we only have true claims and more claims are unlabeled. We follow the definition in [22] to present unlabeled claim X_U as a collection of the true claims and the false claims with certain truthfulness prior probability. We denote the claim-evidence distribution in the equation: $p(\mathbf{x}) = \pi_p p(\mathbf{x}|y = 1) + \pi_n p(\mathbf{x}|y = 0)$, where π_p is the truthfulness prior that usually remains unknown and $\pi_p + \pi_n = 1, \pi_p > 0, \pi_n > 0$. PU claim verification trains on X_P and X_U, and predicts the truthfulness $y \in \{0, 1\}$.

B. BERT Encoder

Given a claim-evidence pair, we employ BERT [15] to obtain the vector representation. BERT is trained to predict the next sentence and masked words using extremely large datasets, so the semantic of sentences is well captured. We regard the hidden state of the [CLS] token as the presentation. We add the [SEP] token to separate the claim and the evidence. If a claim has multiple evidence, we concatenate them into a single sentence. Let $\mathbf{d}$ denotes the output of BERT and $[;]$ denotes concatenation. We consider three types of input: the claim $\mathbf{C}$, the evidence $\mathbf{E}$, and the claim-evidence pair $[\mathbf{C}; \mathbf{E}]$. The encoder transforms each component and gets encoded representation $\mathbf{d}^c, \mathbf{d}^e$ and $\mathbf{d}^{[c;e]}$. Following a practice that learns the interaction between a premise and a hypothesis [13], we further require the encoder to take subtraction and element-wise multiplication $\odot$ for the claim-evidence interaction. Thus, the final representation $\mathbf{x}$ of a claim-evidence pair is given in the equation: $\mathbf{x} = [\mathbf{d}^c; \mathbf{d}^e; \mathbf{d}^{[c;e]}; \mathbf{d}^c - \mathbf{d}^e; \mathbf{d}^c \odot \mathbf{d}^e]$. We store $\mathbf{x}$ for all claim-evidence pairs. When building GAN, we use $\mathbf{x}$ as the generating target, instead of the discrete lexicons.

C. PU Claim-Truthfulness Alignment

The foundation of our model is the generative adversarial network (GAN) [24], which includes a generator G and a discriminator D. GAN estimates the distribution $p(x)$ via an adversarial competition between G and D. D is trained to distinguish the generated sample $\hat{\mathbf{x}}$ from the true sample $\mathbf{x}$, and G is trained to generate a better $\hat{\mathbf{x}}$ to fool D. We optimize D and G alternatively via a min-max game:

$$\min_G \max_D \mathcal{L}(G, D) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \log(D(\mathbf{x})) + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} \log(1 - D(G(\mathbf{z})))$$

where $p_z(\mathbf{z})$ denotes a simple distribution, such as $\mathcal{N}(0, 1)$.

For PU claim verification, we propose to model the joint claim-truthfulness distribution, so that predicting truthfulness can be achieved as an intermediate procedure for matching the joint distribution. We consider two types of generators: a conditional generator $p(\hat{\mathbf{x}}|y)$ that generates the claim representation $\hat{\mathbf{x}}$ based on the truthfulness and a truthfulness generator $p(\hat{y}|\mathbf{x})$ that imitates the role of a classifier to predict the label $\hat{y}$ given the claim representation.

1) Adversarial Alignment: Ideally, we prefer each generator to individually match the true joint distribution $p(\mathbf{x}, y)$, i.e. $p(\hat{\mathbf{x}}|y)p(y) \rightarrow p(\mathbf{x}, y)$ and $p(\hat{y}|\mathbf{x})p(\mathbf{x}) \rightarrow p(\mathbf{x}, y)$. However, it is not feasible for PU claim verification as we lack negative examples. Inspired by the work in [28] that learns joint distribution between the observation and a latent variable, we propose to match $p(\hat{\mathbf{x}}|y)p(y)$ and $p(\hat{y}|\mathbf{x})p(\mathbf{x})$ directly, which we mention as claim-truthfulness alignment.

Building a conditional generator is as follows. Assuming $\mathbf{z}$ is sampled from $\mathcal{N}(0, 1)$, we parametrize the generation as a multi-layer fully connected neural network (MLP) G_x, which concatenates y and $\mathbf{z}$ as the input and outputs pseudo claim representation $\hat{\mathbf{x}}$. Similarly, we parametrize another MLP G_y to generate $\hat{y}$ given $\mathbf{x}$.

We follow [28] to design the discriminator D_{xy} that is trained to distinguish $(\mathbf{x}, \hat{y}) \sim p(\mathbf{x}, \hat{y})$ from $(\hat{\mathbf{x}}, y) \sim p(\hat{\mathbf{x}}, y)$. We parametrize D_{xy} with another MLP and take $[\hat{\mathbf{x}}; y]$ and $[\mathbf{x}; \hat{y}]$ as the input. Training G_x, G_y and $D_{x,y}$ requires to optimize the min-max objective:

$$\min_{G_x, G_y} \max_{D_{x,y}} \mathcal{L}_{x,y} = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \log(D_{x,y}(\mathbf{x}, G_y(\mathbf{x}))) + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z}), y \sim p(y)} \log(1 - D_{x,y}(G_x(\mathbf{z}, y), y))$$

where $y \sim p(y)$ denotes the truthfulness prior distribution.

One problem of the above objectives is that y takes discrete value but $\hat{y} = G_y(\mathbf{x})$ gives continues probability. The discriminator could easily detect the difference. To solve the problem, we cast the probability to discrete value with a threshold: $\hat{y} = 1$ if $G_y(\mathbf{x}) > 0.5$ else 0. We use straight-through estimator [48] and back-propagate through the discrete $\hat{y}$ as the $G_y(\mathbf{x})$, i.e. $\frac{\partial \hat{y}}{\partial x} = \frac{\partial G_y(x)}{\partial x}$, so that we can update the network using stochastic gradient descent.

2) Alignment Enforcing: Because we do not have $p(\mathbf{x}, y)$, there is no guarantee to generate $\hat{y}$ that truly represents truthfulness. Thus, we enforce the proposed alignment with additional objectives. The first objective simulates the GenPU [25] by introducing an additional discriminator D_x. The purpose of D_x is to improve G_x. Intuitively, if $\hat{\mathbf{x}}$ is similar to $\mathbf{x}$, it is easier to align $p(\hat{y}, \mathbf{x})p(\mathbf{x})$ and $p(\hat{\mathbf{x}}, y)p(y)$. We parametrize

Algorithm 1 The Training Procedure

for e = 0 $\rightarrow$ total epoch **do**
  Sample a batch of $\mathbf{x}^+, \mathbf{x}, \mathbf{z}, y$
  Obtain $\hat{\mathbf{x}}$ via $G_x(\mathbf{z}, y)$, $\hat{y}$ via $G_y(\mathbf{x})$
  Update D_{xy} by maxmizing $\mathcal{L}_{x,y}$
  Update D_x by maximizing $\mathcal{L}_{x^+}$ + $\mathcal{L}_x$
  Update G_x by minimizing $\mathcal{L}_{x,y}$ + $\mathcal{L}_{x^+}$ + $\mathcal{L}_x$ + $\mathcal{L}_{Cy^+}$ + $\mathcal{L}_{Cy}$
  Update G_y by minimizing $\mathcal{L}_{x,y}$ + $\mathcal{L}_{Cy^+}$ + $\mathcal{L}_{Cy}$ + $\mathcal{L}_{Cu}$
end for
return G_y

D_x with another MLP. We adversarially train D_x and G_x to achieve two min-max games on true claims and on unlabeled claims, where $\mathbf{x}^+$ means positive samples.

$$\min_{G_x}\max_{D_x}\mathcal{L}_{x^+} = \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}\log(1 - D_x(G_x(\mathbf{z}, y=1))) + \mathbb{E}_{\mathbf{x}^+\sim p(\mathbf{x}|y=1)}\log(D_x(\mathbf{x}^+))$$

$$\min_{G_x}\max_{D_x}\mathcal{L}_{x} = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\log(D_x(\mathbf{x})) + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z}), y\sim p(y)}\log(1 - D_x(G_x(\mathbf{z}, y)))$$

The second objective reconstructs a classification process and aims to improve $\hat{y}$ to be more similar as y. Specifically, we use G_y to classify $\mathbf{x}^+$ from $G_x(\mathbf{z}, y=0)$ and $G_x(\mathbf{z}, y=1)$ from $G_x(\mathbf{z}, y=0)$:

$$\min_{G_y, G_x}\mathcal{L}_{Cy^+} = -\mathbb{E}_{\mathbf{x}^+\sim p(\mathbf{x}|y=1)}\log(G_y(\mathbf{x}^+)) - \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}\log(1 - G_y(G_x(\mathbf{z}, y=0)))$$

$$\min_{G_y, G_x}\mathcal{L}_{Cy} = -\mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}\log(G_y(G_x(\mathbf{z}, y=1))) - \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}\log(1 - G_y(G_x(\mathbf{z}, y=0)))$$

We also use the above objective to constrain G_x and expect to force the generated sample separated.

3) Incorporate Truthfulness Prior: Finally, we follow a common practice to incorporate the class prior π_P into our model, which is critical for PU learning [20], [23]. Let N_U denote the number of unlabeled examples. We apply G_y on the unlabeled data. Then we select top $\pi_P \times N_U$ largest $G_y(\mathbf{x})$ as pseudo positive and leave the rest as pseudo negative. Intuitively, if G_y performs well, it should confidently predict the label, so we design the following objective to enforce the confidence:

$$\min_{G_y}\mathcal{L}_{Cu} = -\frac{1}{N_U}\sum_{i}^{N_U}\hat{y}_i^u\log(G_y(\mathbf{x}_i)) - (1-\hat{y}_i^u)\log(1 - G_y(\mathbf{x}_i)) \quad (1)$$

where $\hat{y}_i^u = 1$ if $\mathbf{x}_i$ is pseudo positive and $\hat{y}_i^u = 0$ if $\mathbf{x}_i$ is pseudo negative.

4) Training: The proposed model is differentiable with the straight-through estimator. We train the model using stochastic gradient descent. Theoretically, Equation 1 adds noise because G_y will make mistakes. But we find it works well in practice. The training procedure is described in Algorithm 1.

IV. Experimental Setup

The primary focus of the experiment is to evaluate our model and empirically understand PU evidence-aware claim verification. We compare the model with other PU-Learning methods. We explore the sensitivity of our model regarding the truthfulness prior. We vary the ratio between labeled data and unlabeled data and the ratio between positive data and negative data for model selection.

A. Dataset and Configuration

We evaluate the proposed model against the FEVER dataset [8]. FEVER utilizes annotators to generate claims based on random sampled wikipedia pages. Then another group of annotators label the claim as *Support* or *Refute* and provide relevant evidence, if they can find one among wiki pages. When there is no evidence found, the annotators label the claim as *NotEnoughInfo*. FEVER has 80,035 *Support* claims, 29,775 *Refute* claims, and 35,639 *NotEnoughInfo* claims for training. The FEVER validation set and test set have 3,333 *Support* claims, 3,333 *Refute* claims, and 3,333 *NotEnoughInfo* claims, respectively. In our experiment, we define the *Support* class as positive and the *Refute* class as negative. We neglect the *NotEnoughInfo* claims as they have no evidence. Unlike binary classification where the hold-out test set is evaluated, PU learning must evaluate the unlabeled data, also. If we can correctly pick negative examples from the unlabeled data, we can change PU learning to supervised learning. Thus, we treat the unlabeled data as an additional test set and name it the unlabeled test set. Under the PU learning setting, the labeled data will always be the positive data. We randomly split the *Support* class into 16 folds so that each fold has roughly 5,000 claims. We consider four experiments for analysis. Each experiment may take a different number of folds to construct the labeled set and unlabeled set. Here are the scenarios that we consider:

GC: In our first empirical study, we take one fold as the labeled positive data. We treat the remaining 15 folds of *Support* claims and the whole *Refute* claims as unlabeled. We report the comparison of different methods with this training data. We name this scenario **GC**, short for **G**eneral **C**omparison.

STP: We name the second scenario **STP**, which explores the **S**ensitivity of models on the estimated **T**ruthfulness **P**rior. We use one fold of the *Support* class as the labeled positive data and keep the remaining as unlabeled. We vary the truthfulness prior whereas the previous experiment takes a fixed prior (Total *Support* claims divided by total claims).

SAL: We name the third scenario **SAL**, which explores the **S**ensitivity of models on the **A**mount of the **L**abeled data. We randomly sample six folds of the *Support* class to be unlabeled. Then we vary the number of folds with labeled positive data from one to ten.

STR: We name the fourth scenario **STR**, which explores the **S**ensitivity of models on the actual **T**ruthfulness

TABLE II: General comparison results.

	BoW	BERT	GEAR	UN-SVM	Wei-Un	UN-NN	uPU	nnPU	GenPU	Our work
FEVER Test Acc	68.62	81.66	88.97	66.26	66.17	67.92	63.36	68.10	53.50	**68.26**
Unlabeled Test Pre	NA	NA	NA	43.32	47.63	53.35	54.18	58.10	43.69	**61.74**
Unlabeled Test Rec	NA	NA	NA	**80.97**	71.42	66.94	63.10	61.36	17.11	58.95
Unlabeled Test F1	NA	NA	NA	56.43	57.15	59.18	58.10	59.68	24.59	**60.28**
Unlabeled Test Acc	NA	NA	NA	64.46	67.73	73.75	74.03	75.75	70.18	**77.92**

Ratio. The truthfulness ratio is defined as the number of unlabeled *Support* claims divided by total unlabeled claims. We randomly sample one fold of the *Support* class to be labeled positive. For the remaining 15 folds, we vary the number of folds with unlabeled data from one to fifteen, and combine them with instances from the *Refute* class.

B. Overall Comparison

We report accuracy for the **GC** scenario. We also report precision, recall, and F1 score on the *Refute* class for the unlabeled set, because it is imbalanced. We acknowledge other works on the FEVER dataset (e.g., [3], [11]), but their results are not comparable with ours. Their methods require to retrieve evidence first whereas we focus on PU learning and leverage gold evidence. We select the following PU methods for comparison: **BoW** is the method that presents the claim-evidence pairs as TF-IDF and uses SVM to verify the claim. We train the model on fully labeled dataset. **BERT** also leverages gold labels for training, but uses the encoded BERT representations. We include this method and **BoW** to demonstrate that a good representation of claim-evidence pair improves the ability to make inference across sentences. **GEAR** [47] is the state-of-the-art method on the FEVER. They propose a graph-based network to reason evidence. We neglect the evidence retrieval step by providing gold evidence and train GEAR on fully labeled dataset. **UN-SVM** is the SVM baseline that views unlabeled examples as instances from the *Refute* class. We find that the linear kernel and the "balanced" class weight give the best performance. We search soft-marge penalty "C" in the range $[10^{-4}, 10^{-1}]$. **Wei-Un** [22] represents weighted unlabeled examples for PU-learning. It consists of two steps: it finds the probability whether an instance is about to be labeled (labeled means positive); if not, the method takes the instance as negative and weights it by the probability obtained in the first step. Each step involves a classifier, and we follow the details given in [22] to use SVM. **UN-NN** is the neural network baseline taking unlabeled examples as the *Refute* class. We weight unlabeled examples by $\frac{N_P}{N_U}$, which gives the best performance. **uPU** [20] represents the unbiased PU-learning. The main idea is to optimize the following loss function: $\mathcal{L} = \pi_p\mathcal{L}^+(\mathbf{x}_p)+\mathcal{L}^-(\mathbf{x}_u)-\pi_p\mathcal{L}^-(\mathbf{x}_p)$ **nnPU** [23] represents the non-negative PU-learning. It extends uPU by preventing the loss $\mathcal{L}$ to be less than zero: $\mathcal{L} = \pi_p\mathcal{L}^+(\mathbf{x}_p) + \max(0, \mathcal{L}^-(\mathbf{x}_u) - \pi_p\mathcal{L}^-(\mathbf{x}_p))$ **GenPU** [25] constructs two generators to generate positive examples and negative examples, as well as three discriminators to discriminate real positive examples and synthetic positive examples, real positive examples and synthetic negative examples, and real unlabeled examples and all synthetic examples.

We repeat the **GC** scenario ten times and report the average result in Table II. For each iteration, we randomly sample one fold without replacement as labeled positive and merge the remaining 15 folds and the *Refute* class as unlabeled. We conduct the t-test for our method and the second-best method, *nnPU*. On the unlabeled test set, our method achieves best results as measured by the accuracy score ($p = 0.008$) and f-1 score ($p = 0.077$). The accuracy score is not significantly better than that of nnPU ($p > 0.1$) on the FEVER test.

We now report an analysis under the binary classification setting. GEAR gives the best accuracy on the FEVER test set. BERT outperforms all other PU learning methods by a large margin. This suggests that PU evidence-aware claim verification is a challenging task, requiring more future work Notably, we fix the BERT model during training as other PU learning methods do, so one may expect a better accuracy of BERT under the binary classification, if the BERT model is fine-tuned. However, the BoW baseline under the binary classification slightly outperforms other PU learning methods. This indicates that modeling the inference between claim and evidence is critical to verify a claim, and BERT appears capable to accomplish this.

We observe that GenPU does not perform well on this task, even though it is the state-of-the-art model in PU learning. Claim-evidence pairs contain descriptive information, such as topics and domains, besides that related to truthfulness, and BERT encodes all the information without separating them. Thus, it appears that the issue is related to the generators in GenPU: they cannot disentangle the truthfulness from the surrounded noise (e.g., topics and domains). It thus generates samples that are sub-optimal to train a verification model. We contend that GenPU's accuracy on this task may improve if we can better represent truthfulness from the claim-evidence pair. Our extended version of the GenPU model does not suffer such an issue. In our design, we do not train the classifier on the generated samples, but jointly optimize it with the generator and the discriminator.

We also observe that our model outperforms other methods on the unlabeled test set more significantly than on the FEVER test set, in term of the accuracy. Since the unlabeled test set contains more *Support* examples than the *Refute* class, our model seems to be more confident on the *Support* class. When looking at the precision and recall scores in Table II, we observe that the increase of the precision leads to a decrease of the recall on the *Refute* class. UN-SVM reports high recall and low precision, which is reasonable because UN-SVM treats unlabeled data as the *Refute* class. Wei-Un, uPU, and nnPU aim to provide a balanced estimation, which improves the precision. Our model balances precision and recall better than

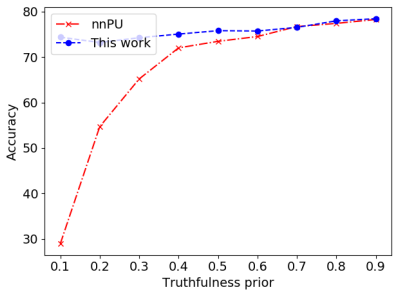

Fig. 1: The **STP** scenario. We report accuracy w.r.t truthfulness priors.

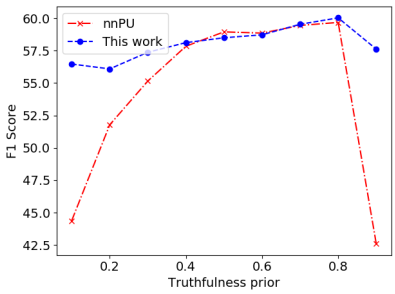

Fig. 2: The **STP** scenario. We report F1 w.r.t truthfulness priors.

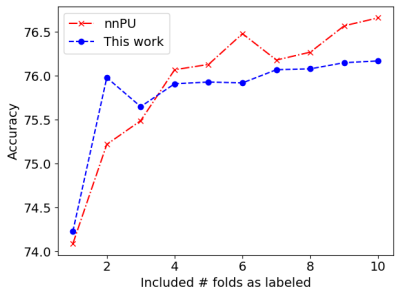

Fig. 3: The **SAL** scenario. We report accuracy w.r.t amount of labeled data.

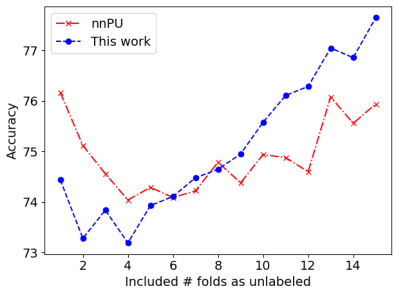

Fig. 4: The **STR** scenario. We report accuracy w.r.t truthfulness ratios.

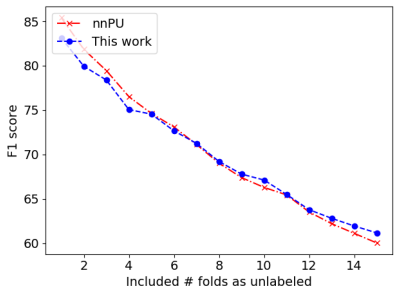

Fig. 5: The **STR** scenario. We report F1 w.r.t truthfulness ratios.

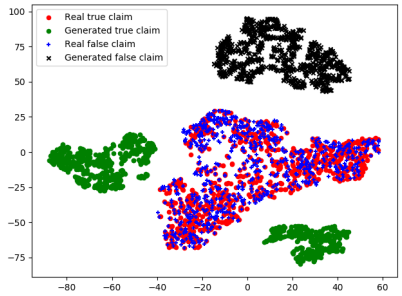

Fig. 6: The T-SNE visualization on encoded claim-evidence pairs.

these baselines, and thus achieves the best F1 score on the unlabeled test set.

C. Sensitivity on the Estimated Truthfulness Prior

The above experiment provides the exact truthfulness ratio (0.716). Nevertheless, it is difficult to estimate the truthfulness prior in real-life applications. For example, political claims may be more problematic than other domains, and the unlabeled data collected may be biased. Thus, we explore the **STP** scenario and compare our model with the second-best method, nnPU.

We vary the estimated prior from 0.1 to 0.9 and take 0.1 as the basic unit. The prior denotes the percentage of the *Support* claim that we believe to exist in the unlabeled data. We repeat the experiment as in the previous section and report the accuracy and the F1 score in Figure 1 and Figure 2, respectively. Figure 1 shows that our model outperforms nnPU in accuracy when truthfulness prior is biased towards the *Refute* class. When the prior is far less than the actual truthfulness ratio (between 0.1 and 0.4), our model shows a strong advantage over nnPU. The decline of truthfulness prior significantly hurts the accuracy of nnPU, but only slightly affects our model. When the prior is close to the actual truthfulness ratio, both methods exhibit comparative results.

Figure 2 shows that our model outperforms nnPU on the F1 score when the truthfulness prior is over-represented (more than 0.8) or under-represented (less than 0.4). There is no significant difference between the two models when the truthfulness prior is in the range of 0.4 to 0.8. Compared to Figure 1, we observe that F1 of nnPU drops significantly at 0.9, whereas accuracy is not affected, because a large prior π_p is likely to cause nnPU overfitting on the positive class. The nnPU method applies the truthfulness prior to penalize biasing unlabeled data as negative. A skewed prior estimation may lead nnPU to bias either class (an under-represented prior biases the negative class, whereas a over-represented prior biases the positive class). Thus, it is not surprising that nnPU is sensitive to the prior. On the contrary, our model splits the unlabeled data and reinforces the confident predictions. Therefore, our model maintains a good prediction when the prior is ill-estimated. We conclude that our model is robust to the truthfulness prior. This is particularly important when estimating the truthfulness prior is not feasible.

D. Sensitivity on the Amount of the Labeled Data

We examine the **SAL** scenario and compare our model with the second best method, nnPU. We first sample six folds of the *Support* class and combine them with the *Refute* class as unlabeled. For the remaining ten folds, we start with one fold as the labeled data and ignore the other nine folds. Then, we add one fold of labeled data at a time and keep the unlabeled data fixed. Our goal is to vary the amount of labeled data by controlling the number of folds.

We report the accuracy score in Figure 3. We first observe that increasing the amount of labeled data benefits both our model and nnPU. When we include two folds as labeled, our model has a clear benefit. In addition, as we introduce more folds of the *Support* class as labeled, nnPU starts to outperform our model. We suspect the ratio between labeled data and unlabeled data may affect the performance. However, no other work reports a similar observation. We leave this study for future work.

E. Sensitivity on the Actual Truthfulness Ratio

We examine the **STR** scenario and again compare our model with nnPU. We first sample one fold of the *Support* class as labeled. For the remaining 15 folds, we start the experiment by combining one fold *Support* examples with the *Refute* class as unlabeled. Then we increment one fold at a time to be unlabeled. By controlling the number of folds, we vary the truthfulness ratio of the unlabeled data. We report accuracy and F1 score in Figure 4 and Figure 5, respectively.

In terms of the accuracy, nnPU performs better than our model when a small number of folds (less than six) of the positive examples are included as the unlabeled data. When unlabeled data contains more negative data than positive data, regarding unlabeled as the negative class brings less bias. Besides, we would expect UN-SVM, UN-NN, Wei-Un, and uPU all perform well when negative data dominates the unlabeled data. However, when unlabeled data contains more positive class, our model outperforms nnPU. There is a similar trend in F1 score: nnPU outperforms our model in the presence of more negative data as unlabeled and our model outperforms nnPU in the presence of positive data as unlabeled, though the difference is smaller than that of accuracy. We conclude that our model is better for those cases where unlabeled data contains more instances from the positive.

F. Visualization

We adopt t-sne [49] to visualize the real representation and the generated representation of claim-evidence pairs. We report the visualization in Figure 6, where green is the generated *Support* claim-evidence pair, black is the generated *Refute* claim-evidence pair, red is the real *Support* claim-evidence pair, and blue is the real *Refute* claim-evidence pair. We choose the model with highest validation accuracy. Our first observation is that claim-evidence pairs overlap closely. This is reasonable, because truthfulness is not clearly represented in the encoded BERT representation. BERT also embeds other dominant or easily expressed information. For example, given the same evidence, *Andy Roddick lost 5 Master Series between 2002 and 2010*, and *Andy Roddict won 5 Master Series between 2002 and 2010* would give opposite truthfulness value, but the two claims differ by only one word. We would expect the two representations to be similar in the embedding space. The truthfulness value is not visible in the plot since the embeddings of the two classes overlap.

Furthermore, GenPU does not appear to generate samples that capture well the truthfulness, as it is fooled by other overwhelming information. We further observe that the generated claim-evidence representation is separated from the real claim-evidence representation. This may benefit classification, as the decision boundary is forced to be placed between green and blue samples. Besides, a recent study argues that a good semi-supervised learning requires a bad generator [50]. We extend their theory and suggest a bad generator may also benefit PU learning. However, since we model the BERT representation in this work, it is not clear what is the meaning of the generated samples. We will investigate this aspect in our future work.

V. Conclusion

In this work, we examine evidence-aware claim verification under positive-unlabeled learning. We study the scenario where we have some claim-evidence pairs for which we know the evidence supports the claim (i.e. labeled) and some claim-evidence pairs that are unlabeled. We hypothesize that generating pseudo negative samples to train a binary classifier may not be feasible, because claim-evidence pairs contain an overwhelming amount of noise and the truthfulness of a claim is "hidden" in the noise. We extend GenPU [25] from two perspectives: we use BERT [15] to encode claim-evidence pairs and leverage the classifier as part of the GAN, forcing it to focus on the classification instead of the generation. We conduct extensive experiments to analyze our proposed model, and compare it with a number of baselines, including nnPU [23]. We show that the proposed model is robust when the truthfulness prior varies and has a clear benefit over the baselines when the estimation of the prior is not accurate. The empirical study shows that our model favors the cases where we have (1) more unlabeled data than labeled data or (2) more unlabeled positive data than unlabeled negative data.

The proposed model has a few limitations that we aim to tackle in the near future. First, both our model and nnPU are sensitive to the amount of labeled data and the ratio between unlabeled positive data and unlabeled negative data. Our approach, extend GenPU and leveraging verification into GAN, may not fully address this issue. Ideally, one desires to conjecture that the generated samples represent truthfulness of claim-evidence pairs and therefore help us in finding the decision boundary between the true claims and the false claims. However, such a study is non trivial. One solution is to seek to construct a representation learning that can delineate the truthfulness from other pieces of information. Then we can compare the generated samples with the truthfulness embeddings. In addition, since representation learning has achieved much progress in the field of (unsuper)supervised learning, extending representation learning under the PU learning is a promising future research direction.

VI. Acknowledgement

Research was supported in part by grants NSF 1838147, NSF 1838145, ARO W911NF-20-1-0254. 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 any copyright notation herein.

References

[1] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[2] A. Hanselowski, H. Zhang, Z. Li, D. Sorokin, B. Schiller, C. Schulz, and I. Gurevych, "Multi-sentence textual entailment for claim verification," in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 2018, pp. 103–108.

[3] T. Yoneda, J. Mitchell, J. Welbl, P. Stenetorp, and S. Riedel, "Ucl machine reading group: Four factor framework for fact finding (hexaf)," in *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, 2018, pp. 97–102.

[4] S. Zhi, Y. Sun, J. Liu, C. Zhang, and J. Han, "Claimverif: a real-time claim verification system using the web and fact databases," in *Proceedings of the 2017 ACM on CIKM*. ACM, 2017, pp. 2555–2558.

[5] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi, "Truth of varying shades: Analyzing language in fake news and political fact-checking," in *Proceedings of the 2017 Conference on EMNLP*, 2017, pp. 2931–2937.

[6] W. Y. Wang, ""liar, liar pants on fire": A new benchmark dataset for fake news detection," in *Proceedings of the 55th ACL*. Association for Computational Linguistics, 2017, pp. 422–426.

[7] K. Popat, S. Mukherjee, A. Yates, and G. Weikum, "Declare: Debunking fake news and false claims using evidence-aware deep learning," in *Proceedings of the 2018 Conference on EMNLP*, 2018, pp. 22–32.

[8] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "Fever: a large-scale dataset for fact extraction and verification," in *Proceedings of the 2018 Conference of NAACL*, vol. 1, 2018, pp. 809–819.

[9] W. S. Lee and B. Liu, "Learning with positive and unlabeled examples using weighted logistic regression," in *ICML*, vol. 3, 2003, pp. 448–455.

[10] X. Li and B. Liu, "Learning to classify texts using positive and unlabeled data," in *IJCAI*, vol. 3, 2003, pp. 587–592.

[11] Y. Nie, H. Chen, and M. Bansal, "Combining fact extraction and verification with neural semantic matching networks," *arXiv preprint arXiv:1811.07039*, 2018.

[12] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on EMNLP*, 2015, pp. 632–642.

[13] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced lstm for natural language inference," in *Proceedings of the 55th ACL*, vol. 1, 2017, pp. 1657–1668.

[14] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Proceedings of the 2016 Conference on EMNLP*, 2016, pp. 2249–2255.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[16] H. Li, Z. Chen, B. Liu, X. Wei, and J. Shao, "Spotting fake reviews via collective positive-unlabeled learning," in *2014 IEEE International Conference on Data Mining*. IEEE, 2014, pp. 899–904.

[17] Y. Ren, D. Ji, and H. Zhang, "Positive unlabeled learning for deceptive reviews detection." in *EMNLP*, 2014, pp. 488–498.

[18] B. Liu, W. S. Lee, P. S. Yu, and X. Li, "Partially supervised classification of text documents," in *ICML*, vol. 2. Citeseer, 2002, pp. 387–394.

[19] B. Liu, Y. Dai, X. Li, W. Lee, and P. Yu, "Building text classifiers using positive and unlabeled examples," in *Third IEEE International Conference on Data Mining*. IEEE, 2003, pp. 179–186.

[20] M. Du Plessis, G. Niu, and M. Sugiyama, "Convex formulation for learning from positive and unlabeled data," in *International Conference on Machine Learning*, 2015, pp. 1386–1394.

[21] M. C. Du Plessis, G. Niu, and M. Sugiyama, "Analysis of learning from positive and unlabeled data," in *NeurIPS*, 2014, pp. 703–711.

[22] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 213–220.

[23] R. Kiryo, G. Niu, M. C. du Plessis, and M. Sugiyama, "Positive-unlabeled learning with non-negative risk estimator," in *NeurIPS*, 2017, pp. 1675–1685.

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

[25] M. Hou, B. Chaib-Draa, C. Li, and Q. Zhao, "Generative adversarial positive-unlabeled learning," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 2018, pp. 2255–2261.

[26] L. Chongxuan, T. Xu, J. Zhu, and B. Zhang, "Triple generative adversarial nets," in *NeurIPS*, 2017, pp. 4088–4098.

[27] Z. Deng, H. Zhang, X. Liang, L. Yang, S. Xu, J. Zhu, and E. P. Xing, "Structured generative adversarial networks," in *NeurIPS*, 2017, pp. 3899–3909.

[28] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville, "Adversarially learned inference," *arXiv preprint arXiv:1606.00704*, 2016.

[29] A. Odena, "Semi-supervised learning with generative adversarial networks," *arXiv preprint arXiv:1606.01583*, 2016.

[30] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum, "Where the truth lies: Explaining the credibility of emerging claims on the web and social media," in *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 2017, pp. 1003–1012.

[31] N. Hassan, F. Arslan, C. Li, and M. Tremayne, "Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 1803–1812.

[32] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," in *COLING 2018*. Association for Computational Linguistics, 2018, pp. 3391–3401.

[33] V. Vydiswaran, C. Zhai, and D. Roth, "Content-driven trust propagation framework," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 974–982.

[34] L. Ge, J. Gao, X. Li, and A. Zhang, "Multi-source deep learning for information trustworthiness estimation," in *Proceedings of the 19th ACM SIGKDD*. ACM, 2013, pp. 766–774.

[35] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "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*. ACM, 2014, pp. 1187–1198.

[36] X. Li, W. Meng, and Y. Clement, "Verification of fact statements with multiple truthful alternatives," in *12th International Conference on Web Information Systems and Technologies*, 2016.

[37] M. Wan, X. Chen, L. Kaplan, J. Han, J. Gao, and B. Zhao, "From truth discovery to trustworthy opinion discovery: An uncertainty-aware quantitative modeling approach," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1885–1894.

[38] X. Yin, J. Han, and S. Y. Philip, "Truth discovery with multiple conflicting information providers on the web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 6, pp. 796–808, 2008.

[39] R. Pochampally, A. Das Sarma, X. L. Dong, A. Meliou, and D. Srivastava, "Fusing data with correlations," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 2014, pp. 433–444.

[40] J. Pasternack and D. Roth, "Making better informed trust decisions with generalized fact-finding," in *IJCAI*, vol. 11, 2011, pp. 2324–2329.

[41] X. L. Dong, E. Gabrilovich, K. Murphy, V. Dang, W. Horn, C. Lugaresi, S. Sun, and W. Zhang, "Knowledge-based trust: Estimating the trustworthiness of web sources," *Proceedings of the VLDB Endowment*, vol. 8, no. 9, pp. 938–949, 2015.

[42] B. Shi and T. Weninger, "Discriminative predicate path mining for fact checking in knowledge graphs," *Knowledge-Based Systems*, vol. 104, pp. 123–133, 2016.

[43] P. Shiralkar, A. Flammini, F. Menczer, and G. L. Ciampaglia, "Finding streams in knowledge graphs to support fact checking," in *Data Mining (ICDM), 2017*. IEEE, 2017, pp. 859–864.

[44] J. Thorne and A. Vlachos, "Automated fact checking: Task formulations, methods and future directions," in *COLING 2018*. Association for Computational Linguistics, 2018, pp. 3346–3359.

[45] I. Dagan, O. Glickman, and B. Magnini, "The pascal recognising textual entailment challenge," in *Machine Learning Challenges Workshop*. Springer, 2005, pp. 177–190.

[46] A. Williams, N. Nangia, and S. R. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," *arXiv preprint arXiv:1704.05426*, 2017.

[47] J. Zhou, X. Han, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Gear: Graph-based evidence aggregating and reasoning for fact verification," in *Proceedings of the 57th ACL*, 2019, pp. 892–901.

[48] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[49] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[50] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov, "Good semi-supervised learning that requires a bad gan," in *NeurIPS*, 2017, pp. 6510–6520.