Representation Learning for Trustworthy AI

by

Ahmadreza Mosallanezhad

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2023 by the
Graduate Supervisory Committee:

Huan Liu, Co-Chair
Michelle Mancenido, Co-Chair
Adam Doupé
Ross Maciejewski

ARIZONA STATE UNIVERSITY

May 2023

ABSTRACT

Artificial Intelligence (AI) systems have achieved outstanding performance and have been found to be better than humans at various tasks, such as sentiment analysis, and face recognition. However, the majority of these state-of-the-art AI systems use complex Deep Learning (DL) methods which present challenges for human experts to design and evaluate such models with respect to privacy, fairness, and robustness. Recent examination of DL models reveals that representations may include information that could lead to privacy violations, unfairness, and robustness issues. This results in AI systems that are potentially untrustworthy from a socio-technical standpoint.

Trustworthiness in AI is defined by a set of model properties such as non-discriminatory bias, protection of users' sensitive attributes, and lawful decision-making. The characteristics of trustworthy AI can be grouped into three categories: Reliability, Resiliency, and Responsibility. Past research has shown that the successful integration of an AI model depends on its trustworthiness. Thus it is crucial for organizations and researchers to build trustworthy AI systems to facilitate the seamless integration and adoption of intelligent technologies.

The main issue with existing AI systems is that they are primarily trained to improve technical measures such as accuracy on a specific task but are not considerate of socio-technical measures. The aim of this dissertation is to propose methods for improving the trustworthiness of AI systems through representation learning. DL models' representations contain information about a given input and can be used for tasks such as detecting fake news on social media or predicting the sentiment of a review. The findings of this dissertation significantly expand the scope of trustworthy AI research and establish a new paradigm for modifying data representations to balance between properties of trustworthy AI. Specifically, this research investigates multiple techniques such as reinforcement learning for understanding trustworthiness in users' privacy, fairness, and robustness in classification tasks like cyberbullying detection and fake news detection. Since most social measures in trustworthy AI cannot be used to fine-tune or train an AI model directly, the main contribution of this dissertation lies in using reinforcement learning to alter an AI system's behavior based on non-differentiable social measures.

DEDICATION

I dedicate my dissertation to my lovely wife, Marzie Bitaab, for supporting and encouraging me all the way!

I also dedicate this dissertation to everyone who helped, encouraged, and accompanied me. Without their help, this journey which would not be possible.

ACKNOWLEDGMENTS

This dissertation would be impossible without the help of my advisors, Dr. Huan Liu and Dr. Michelle Mancenido. I want to thank them for supporting me through my Ph.D. with excellent advising skills and guidance. I'm very fortunate to have Dr. Liu and Dr. Mancenido as my Ph.D. advisors, from whom I learned many abilities: how to write research papers and present ideas, how to find problems and approach them, and how to establish my future career and strive to improve.

I would like to thank my committee members, Dr. Adam Doupé, and Dr. Ross Maciejewski, for their helpful suggestions and insightful comments. I am grateful to the committee members for their suggestions and challenging questions throughout my comprehensive exam and proposal defense, which allow me to investigate more in my research area.

I was lucky to work as an intern at NVIDIA with amazing colleagues and mentors: Oleksii Kuchaiev, Jonathan Cohen, Sandeep Subramanian, and Abhinav Khattar.

During my Ph.D. study, my friends and colleagues provided me with consistent encouragement and support, and they deserve special thanks. I am thankful to my colleagues and friends at Data Minning and Machine Learning Lab: Kai Shu, Lu Cheng, Nayoung Kim, Kaize Ding, Raha Moraffah, Amrita Bhattacharjee, Faisal Alatawi, Mansooreh Karami, Kumarage Tharindu, Paras Sheth, Bohan Jiang, Ujun Jeong, and Zeyad Alghamdi. In particular, thank Kai Shu and Lu Cheng who taught me how to do research and come up with ideas. I am lucky to have you as my friends and colleagues.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Chapter 1

INTRODUCTION

Most AI systems today use Deep Learning which is complex and difficult for human experts to design and evaluate with respect to privacy, fairness, and robustness. The most important step in such systems is converting input data into a representation containing vital information for specific tasks. The representation is then used for inference. Existing state-of-the-art methods achieve outstanding performance when using technical measures such as accuracy, F-1 score, etc. However, recent studies have shown that these measures are not suitable for measuring the social aspects of an AI/ML model. A key factor that can make an AI/ML model not suitable for public use, is the presence of information in data representations that can cause privacy breaches or fairness concerns [20, 82]. Trustworthy AI focuses on improving the social aspect of an AI/ML model. It includes measurements of several interrelated properties such as reliability, privacy, fairness, and robustness. While accuracy is highly valued in machine learning, trustworthy AI involves balancing these properties and sometimes sacrificing accuracy for privacy. These properties can have different interpretations and formulations. For instance, fairness can mean demographic parity, equal odds, or individual fairness - some of which may conflict with each other. Figure 1 shows the main areas of study in trustworthy AI. Specifically, trustworthy AI aims to address three key aspects of an AI/ML model [12, 1]:

- Responsible: This property targets *fairness*, ethics, and sustainability of an AI/ML model.

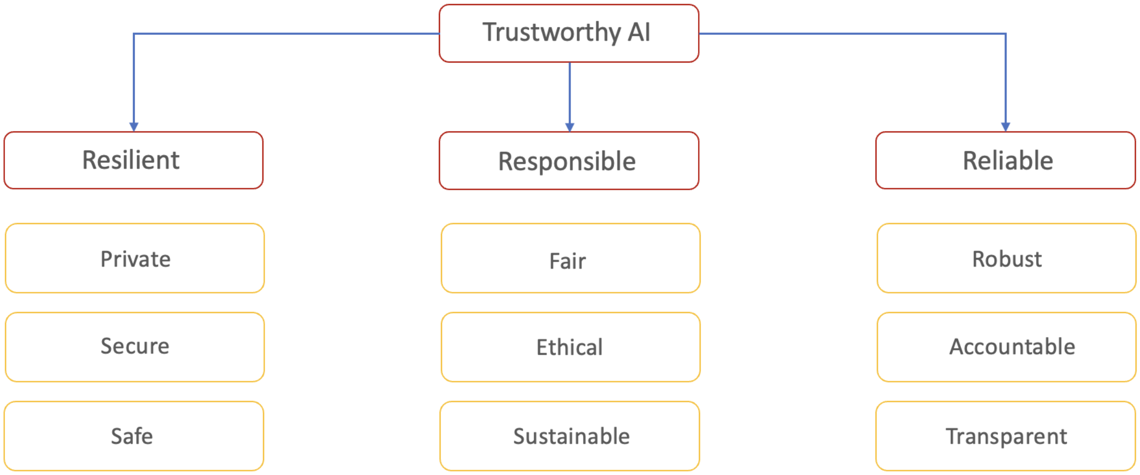


Figure 1. Main Aspects of a Trustworthy AI

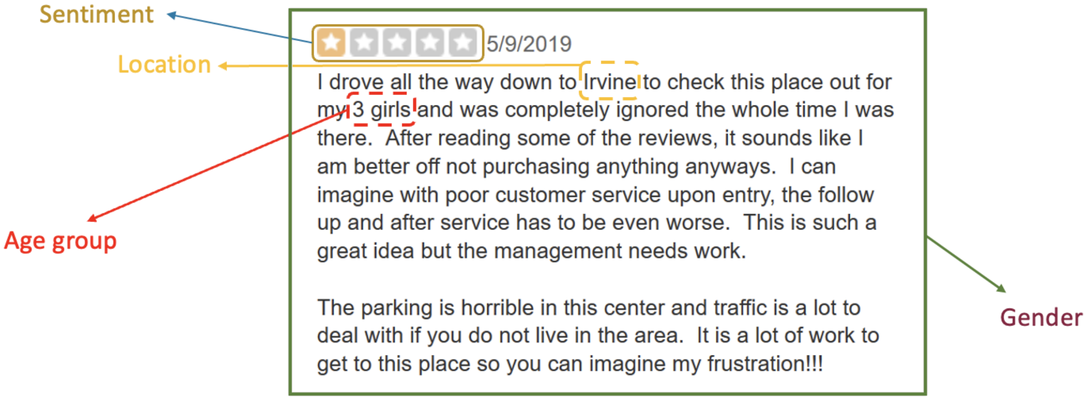
- Resilient: This property targets *privacy*, security, and safeness of an AI/ML model.
- Reliable: Targets *robustness*, accountability, and transparency.

These properties often intersect and improving one can enhance others. For instance, making a classifier fair by removing sensitive user information like gender, age or location can also increase its security. Similarly, an AI/ML model that is robust to input data changes (such as a domain adaptive model) is less vulnerable to evasion attacks. This dissertation concentrates on improving privacy, fairness, and robustness by altering representations in an AI/ML model. Each property is described in the following subsections:

1.1 Privacy-preserving Representation Learning

Machine learning models are being used in many applications on a daily basis. These models are usually trained on user-generated data. Social media users generate a tremendous amount of data such as profile information, network connections, and online reviews and posts. Online vendors use this data to understand users' preferences and further predict their future needs. However, user-generated data is rich in content and malicious attackers can infer users' sensitive information. AOL search data leak in 2006 is an example of privacy breaches that results in users' re-identification according to the published AOL search logs and queries [94]. Therefore, these privacy concerns mandate that data be anonymized before publishing. The problem of representation learning in both text and image data types is studied by using adversarial training and reinforcement learning. Adversarial learning is the state-of-the-art approach for creating privacy-preserving data representations [66, 25].In these methods, a model is trained to create an embedding, but does not control for the privacy-utility balance. The recent success of reinforcement learning (RL) [95, 129] shows a feasible alternative: By leveraging reinforcement learning, feedback from attackers and utility can be included in a reward function that allows for the control of the privacy-utility balance. Furthermore, parts of embedded data can be perturbed by an RL agent for preserving both utility and privacy, instead of retraining an embedding as in adversarial learning. In this dissertation, the focus is on preserving users' privacy through information removal on the following data types:

Text data type: Recent research has shown that textual data alone may contain sufficient information about users' private attributes that they do not want to disclose such as age, gender, location, political views, and sexual orientation [88, 138]. Little attention has been paid to protecting users' textual information [66, 145, 4, 110]. Figure 2 shows an example where user-shared reviews can include private information.
Image data type: Security and surveillance systems, such as those found in private industries (e.g., biometric access control systems) and public domains (e.g., face recognition systems at airports and traffic thruways), acquire images of people's faces



5/9/2019

I drove all the way down to Irvine to check this place out for my 3 girls and was completely ignored the whole time I was there. After reading some of the reviews, it sounds like I am better off not purchasing anything anyways. I can imagine with poor customer service upon entry, the follow up and after service has to be even worse. This is such a great idea but the management needs work.

The parking is horrible in this center and traffic is a lot to deal with if you do not live in the area. It is a lot of work to get to this place so you can imagine my frustration!!!

Figure 2. Example of Privacy Implications in Text.

for verification and identification tasks. The ease of collecting data on private citizens raises concerns about violating privacy-preserving contracts or expectations [56], because organizations have been known to exercise their prerogative to sell information on individuals or have been subject to malicious attacks that compromised users' privacy [83, 6]. For instance, in 2019, a malicious cyber-attack on a US Customs and Border Protection subcontractor exposed travelers' photos [75]. Due to such threats to individual liberties and privacy, one method that has been proposed to protect an individual's private information is to anonymize it before sharing. While some recent studies have shown that face images contain user-related private information such as gender or race [42, 76, 73], research on protecting face images from adversarial attacks has been limited [17].

1.2 Fair Representation Learning

Deep learning models are being used in many applications, especially in classifying text documents. Past research has shown that although the results of using deep learning models on text are promising, nevertheless, may come from a deeply biased model that captures, uses, and even amplifies the unintended biases embedded in social media data [144]. That is because humans are biased and human-generated language corpora can introduce human social prejudices into model training processes [14]. One important example of bias in sentiment analysis classifiers is unfairness in cyberbullying detection models[144, 30]. Davidson et al. reveal that tweets in African-American Vernacular English are more likely to be classified as abusive or offensive[30]. Similarly, a cyberbullying classifier may simply take advantage of sensitive triggers, e.g., demographic-identity information (e.g., "gay") and offensive terms ("stupid," "ni***r"), to make decisions.

For the problem of fairness in text classification methods, mitigating the unintended bias in cyberbullying detection is chosen. This task poses multi-faceted challenges that render recent model-agnostic research in fair text classification – especially, data manipulation methods [32, 128] – inapplicable.

1.3 Robust Representation Learning

A robust AI system operates reliably at expected performance levels even when faced with uncertainty or small changes in input data. Reliability comes from various aspects of an AI/ML model, including privacy, fairness, and data augmentation. In robustness, domain adaptation is explored. In real-world scenarios, large datasets for training or re-training deep learning models are often lacking. Data collection and annotation are time-consuming and costly. This challenge motivates the exploration of methods for adapting an already trained model from a source domain to another.

Two approaches are explored: steering language models for data augmentation and using domain adaptation methods to adopt a model's domain.

Synthetic data generation: Text generation is an important task for Natural Language Processing (NLP). With the rise of deep neural networks such as Recurrent Neural Networks (RNNs) and Long Shot Term Memory (LSTM) cells [38], there has been significant performance improvement in language modeling and text generation. Text generation has many different applications such as paraphrase generation and data augmentation. It is well-known that training data has a crucial role in the performance and quality of the AI/ML model [63]. Data augmentation helps us to diversify and improve the training set for an AI/ML model to improve its performance. In this research area, the problem of text generation for creating a robust AI/ML model is studied and a new problem in fake news detection is introduced.

Domain Adaptation: Domain adaptation problem arises when a well-performing model on a different (but related) target data distribution is aimed to be learned from a source data distribution. Domain adaptation is used when there is a lack of labeled data in the target domain. Re-training a model on the target domain would require a large amount of labeled data which may not be available or may be expensive to obtain. Domain adaptation allows us to leverage the knowledge learned from a related source domain where labeled data is available to improve the performance of the model on the target domain without requiring additionally labeled data [85]. This research area is focused on creating a domain-independent data representation for classification purposes using reinforcement learning.

In this dissertation, the focus is on studying the trustworthiness of AI/ML models. Experiments are first conducted to measure their most important social aspects in various applications such as cyberbullying detection. Then, a method is proposed to improve these social aspects and extensive experiments are conducted to demonstrate the effectiveness of the proposed approaches. Contributions to each social aspect are detailed below:

1. Privacy (chapter 3 and chapter 4):
    - The novel problem of joint privacy and utility-preserving image and text representation is studied.
    - A novel reinforcement learning-based framework that uses an adversary to improve the privacy of data representations is designed.
    - Extensive experiments are conducted to show the effectiveness of the methods in preserving users' privacy while maintaining the usefulness of data representation.
2. Fairness (chapter 5):
    - The novel problem of fairness in sequential data classification is studied.
    - An RL-based optimization framework for improving fairness in sequential data classification is proposed.
    - Experiments and case studies are performed to show how unfairness affects misclassifying sequential data and how the framework helps prevent that.
3. Robustness (chapter 6 and chapter 7):
    - The problem of data-efficient fake news detection is studied by exploring methods in data generation and domain adaptation.
    - An RL-based framework to enhance out-of-domain fake news detection methods by using non-differentiable measures is proposed.
    - Case studies, user studies, and experiments are conducted to indicate the effectiveness of the method.

This dissertation is organized as follows. Existing methods for trustworthy AI are first reviewed in chapter 2. The privacy aspect is then explored in chapter 3 and chapter 4. The fairness aspect in sequential data classification is reviewed in chapter 5. The challenges and methods for data-efficient fake news detection are studied in chapter 6 and chapter 7. Finally, the conclusion and future direction are provided in chapter 8.

Chapter 2

METHODS FOR TRUSTWORTHY AI

In this section, I go through methods that are used for creating trustworthy AI/ML models. Past research is categorized based on their goal into (1) privacy-preserving machine learning (resilient), (2) robust machine learning models (reliable), and (3) fair machine learning models (responsible):

- **Privacy-preserving machine learning models:**

  Differential Privacy (DP) is the general objective that is used for creating privacy-preserving models. DP is known to preserve privacy by minimizing the chance of individual record identification [105]. DP quantifies how much information is leaked during a particular mechanism and restricts the impact of each sample by introducing the degree of randomness. Recent methods focus on three aspects of privacy-preserving facial recognition tasks: modifying the training procedure [16, 77], inference procedure [148], creating a privacy-preserving image representation [86], and creating a privacy-preserving text representation [83].

  – Differentially Private Training: research in this area tries to create a privacy-preserving AI/ML model by introducing DP in the training process. Mao et al. propose a DP-based method for training a neural network that preserves users' privacy on models that are used on edge devices [77]. Using another approach than adding noise, Ren et al. propose to use a variant of adversarial training for privacy-preserving training of face recognition models [104]. This work uses an optimizer to alter the input so that an adversary cannot detect the true identity of an image. Another work by Tursynbek et al. proposes DP-SGD training method that adds random noise to the gradient during the stochastic gradient descent algorithm [134]. The added noise can be adjusted to make sure the utility of the trained ML model does not decrease by a large margin. In another similar work, Chamikara et al. propose PEEP that tries to add random noise to data for training a face recognition model. PEEP converts images to vectors using Eigenface and applies Laplacian noise over them [16].

- Differentially Private Inference: research in this area focus on making the inference part of the AI/ML model private. One common method that is used in this area is *model splitting*. In this approach, the ML model is split into two sub-models, a sub-model that extracts features and contains sensitive information and a sub-model for the inference that processes this information. In this method, the sensitive sub-model works on the clients' devices, while the inference model is shared publically [133]. Inspired by the split learning approach, Wen et al. propose S-Net, a method that trains several models on various parts of the face for the face recognition task [139]. As illustrated in Figure 3, this approach trains several models on sub-images. This prevents the attackers from extracting sensitive information about the actual input image. Another approach is to add randomization to the inference process [148]. Although this method can affect the utility of the AI/ML model, it guarantees the privacy of users' data and prevents attackers from performing model extraction attacks.

- Differentially Private Representations: research in this area focuses on creating a robust and privacy-preserving image representation that does

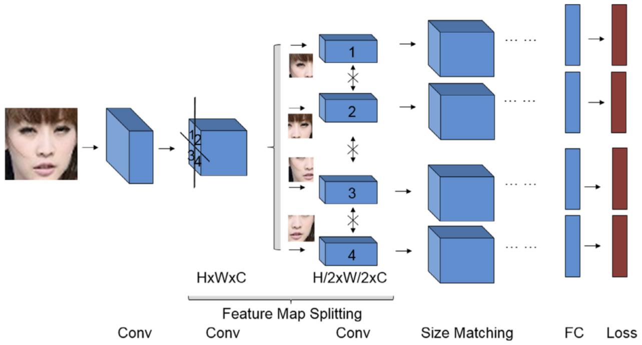


Figure 3. General Architecture of Split-net

not leak users' sensitive information such as age, gender, or race. AIA is one of the models that can be used to create a privacy-preserving image representation [86]. This model uses adversarial training to create an image representation that cannot be used to infer the gender of the users. This model assumes that an image representation created by an autoencoder will be used for one-to-one face matching task (using a siamese network):

$$L_{Total} = L_{AE} + \alpha L_{Siamese} - \beta L_{Adv} \tag{2.1}$$

where L_{AE} indicates the autoencoder's loss, $L_{Siamese}$ and L_{Adv} indicate the utility task (i.e., one-to-one face matching) and the adversary's (i.e., gender classifier) loss, respectively.

Another well-known model for creating privacy-preserving image representation is PPRL-VGAN which uses generative adversarial networks to create a safe image representation [18]. This model considers the utility task as facial expression recognition but can be extended to other tasks as well. This model can be used to change the private attributes of an image such as gender, race, or identity (Figure 4).

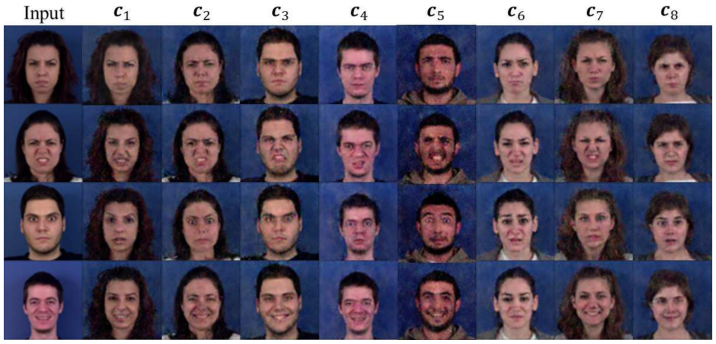


Figure 4. Vgan-based Model's Output Using Various Input Images

Finally, an interesting work by Shan et al. uses an optimization method to add noise to the images, making them private to pre-trained attribute inference models. This method, Fawkes, adds perturbation noise to an image in a way that maximizes the change in its representation, but keeps the noise to a minimum [114]. Although this method helps preserving the privacy of images, it does not guarantee the utility of the image. The goal of this method is to hide users' information from pre-trained image attribute classifiers that are used in the industry such as Microsoft's Azure face API [78], Face++ [35], and Amazon's Rekognition API [3].

- **Fair machine learning models:**

  Unfairness can occur in different applications of AI/ML methods. Past research has shown unfairness in language models [49, 89], natural machine translation models [126, 109], and cyberbullying detection methods [22, 20]. Similar methods are used for removing bias in such models.

  Computational methods can reinforce and even propagate social biases, with unintended biases in text classification tasks coming from datasets [32], distributed word embeddings [36, 9] (e.g., word2vec [79]), contextual word embeddings [60] (e.g., Bert [31]), machine learning algorithms [144, 20], and human annotators [39]. In pioneering work by [9], word embeddings trained on Google News articles were found to exhibit gender stereotypes to an alarming extent. Yet, only a handful of studies [144, 20, 37] have focused on mitigating these unintended biases in text classification, broadly, and toxicity detection, specifically.

One approach for mitigating bias in text classification–and mitigating demographic bias, in particular–is data augmentation [92, 32, 108]. This approach seeks to reduce data bias stemming from the lower weight and/or underrepresentation of minority (relative to a majority) groups by balancing the training data sets. Specifically, one can add external labeled data [32], swap gender-related terms [92], or assign different weights to instances from various groups [87]. The primary drawback of these data manipulation methods is their impracticality (e.g., the costliness of labeling data). Data augmentation can also result in meaningless sentences and may not be suitable for some types of demographic groups (e.g., race). Recent work by [144] sought to address these limitations. Based on the assumption that there are discriminative and non-discriminative data distributions, they sought to reconstruct the non-discriminative data distribution from discriminative ones by instance weighting. Critically, few works have considered the context in debiasing toxicity detection.

- **Robust machine learning models:**

  Machine learning robustness lies within the capabilities of a model that is robust to adversaries, changes in domain, and changes in the environment. Cross-domain modeling refers to a model capable of learning information from data in the source domain and being able to transfer it to a target domain [7]. In general, cross-domain models are categorized into *sample-level* and *feature-level* groups [149]. Sample-level domain adaptation methods focus on finding domain-independent samples by assigning weights to these instances [149, 124]. On the other hand, feature-level domain adaptation methods focus on weighting or extracting domain-independent features [59]. In addition to the aforementioned domain adaptation methods, Gong et al. combined both sample-level and feature-level domain adaptation [40] in BERT to create a domain-independent sentiment analysis model. Similarly, Vlad et al. used transfer learning on an enhanced BERT architecture to detect propaganda across domains [137]. Moreover, Zhuang et al. propose to use auto-encoders for learning unsupervised feature representations for domain adaptation [149]. The goal of this model is to leverage a small portion of the target domain data to train an auto-encoder for learning domain-independent feature representations. In this category, we focus on two works on using counterfactual data augmentation [84] and reinforced domain adaptation [85].

Chapter 3

PRIVACY: DEEP REINFORCEMENT LEARNING-BASED TEXT ANONYMIZATION AGAINST PRIVATE-ATTRIBUTE INFERENCE

3.1 Background

Social media users generate a tremendous amount of data such as profile information, network connections, and online reviews and posts. Online vendors use this data to understand users' preferences and further predict their future needs. However, user-generated data is rich in content and malicious attackers can infer users' sensitive information. AOL search data leak in 2006 is an example of privacy breaches that results in users' re-identification according to the published AOL search logs and queries [94]. Therefore, these privacy concerns mandate that data be anonymized before publishing. Recent research has shown that textual data alone may contain sufficient information about users' private attributes that they do not want to disclose such as age, gender, location, political views, and sexual orientation [88, 138]. Little attention has been paid to protecting users textual information [66, 145, 4, 110].

Anonymizing textual information comes at the cost of losing the utility of data for future applications. Some existing work shows the degraded quality of textual information [4, 145, 110]. Another related problem setting is when the latent representation of the user-generated texts is shared for different tasks. It is very common to use recurrent neural networks to create a representation of user-generated text to use for different machine learning tasks. Hitaj el al. show text representations can leak users' private information such as location [46]. This work aims to anonymize users' textual information against private-attribute inference attacks.

Adversarial learning is the state-of-the-art approach for creating a privacy-preserving text embedding [66, 25]. In these methods, a model is trained to create a text embedding, but we cannot control the privacy-utility balance. The recent success of reinforcement learning (RL) [95, 129] shows a feasible alternative: by leveraging reinforcement learning, we can include feedback of attackers and utility in a reward function that allows for the control of the privacy-utility balance. Furthermore, an RL agent can perturb parts of an embedded text for preserving both utility and privacy, instead of retraining an embedding as in adversarial learning. Therefore, I propose a novel Reinforcement Learning-based Text Anonymizer, namely, RLTA, composed of two main components: 1) an attention-based task-aware text representation learner to extract a latent embedding representation of the original text's content w.r.t. a given task and 2) a deep reinforcement learning based privacy and utility preserver to convert the problem of text anonymization to a one-player game in which the agent's goal is to learn the optimal strategy for text embedding manipulation to satisfy both privacy and utility. The Deep Q-Learning algorithm is then used to train the agent capable of changing the text embedding w.r.t. the received feedback from the privacy and utility sub-components.

We investigate the following challenges: 1) How could we extract the textual embedding w.r.t. a given task? 2) How could we perturb the extracted text embedding to ensure that user private-attribute information is obscured? and 3) How could we preserve the utility of text embedding during anonymization? Our main contributions are: (1) studying the problem of text anonymization by learning a reinforced task-aware text anonymizer, (2) incorporating a data-utility task-aware checker to ensure that the utility of textual embeddings is preserved w.r.t. a given task, and (3) conducting experiments on real-world data to demonstrate the effectiveness of RLTA in an important natural language processing task.

3.2 Problem Statement

Let $\mathcal{X} = \{x_1, x_2, ..., x_N\}$ denotes a set of N documents and each document x_i is composed of a sequence of words. We denote $\mathbf{v}_i \in \mathbb{R}^{d \times 1}$ as the embedded representation of the original document x_i. Let $\mathcal{P} = \{p_1, p_2, ..., p_m\}$ denote a set of m private attributes that users do not want to disclose such as age, gender, location, etc. The goal of reinforced task-aware text anonymizer is to learn an embedding representation of each document and then anonymize it such that 1) users' privacy is preserved by preventing any potential attacker to infer users' private-attribute information from the textual embedding data, and 2) utility of the text embedding is maintained for a given task $\mathcal{T}$ which incorporates such data, e.g., classification. Specifically, we study the following problem:

Problem 3.2.1 *Given a set of documents $\mathcal{X}$, set of private-attributes $\mathcal{P}$, and given task $\mathcal{T}$, learn an anonymizer f that can learn a private embedded representation $\mathbf{v}_i$ from the original document x_i so that, 1) the adversary cannot infer the targeted user's private-attributes $\mathcal{P}$ from the private text representation $\mathbf{v}_i$, and 2) the generated private representation $\mathbf{v}_i$ is good for the given task $\mathcal{T}$. The problem can be formally defined as:*

$$\mathbf{v}_i = f(x_i, \mathcal{P}, \mathcal{T}) \tag{3.1}$$

Due to the success of Reinforcement Learning [116, 95], I use RL to address the aforementioned problem. RL [131] formulates the problem within the framework of

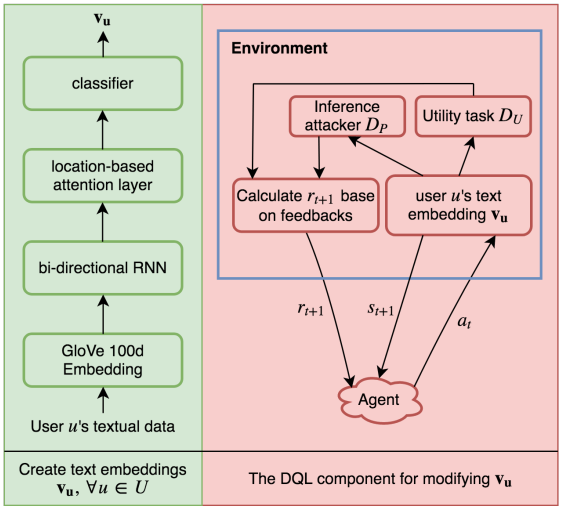

$\mathbf{v_u}$
classifier
location-based attention layer
bi-directional RNN
GloVe 100d Embedding
User u's textual data
Create text embeddings $\mathbf{v_u}$, $\forall u \in U$
Environment
Inference attacker D_P
Utility task D_U
Calculate r_{t+1} base on feedbacks
user u's text embedding $\mathbf{v_u}$
r_{t+1}
s_{t+1}
a_t
Agent
The DQL component for modifying $\mathbf{v_u}$

Figure 5. Architecture of Reinforcement Learning Text Anonymizer

Markov Decision Process (MDP), and learns an action-selection policy based on past observations of transition data. An MDP is defined by state space $S = \{s\}$, action space $A = \{a\}$, transition probability function $P : S \times A \times S \rightarrow [0, 1]$ and reward function $r : S \times A \times S \rightarrow \mathbb{R}$.

3.3 Proposed Method

We discuss the reinforced task-aware text anonymizer framework. The input of this private system is the user generated text, and the output is a privacy-preserving text representation. As in Figure. 5, this framework consists of two major components: 1) an attention based task-aware text representation learner, and 2) a deep RL based privacy and utility preserver. The text representation learner aims to extract the embedded representation of a document w.r.t. a given task by minimizing the task's loss function. Then, the deep RL preserver manipulates the embedded text representation by learning the optimal strategy so that both privacy and utility of the embedded representation are preserved. It includes two sub-components: 1) private-attribute inference attacker D_P, and 2) data-utility task-aware checker D_U. The former seeks to infer user private-attribute information based on their embedded text representation. The latter incorporates the given manipulated embedded text representation for a given task $\mathcal{T}$ and investigates the usefulness of the latent representation for $\mathcal{T}$.

The RL component then utilizes the feedback of the two sub-components to guide the data manipulation process by ensuring that the new text embedding does not leak user private-attributes by confusing the adversary in D_P and the changes made to the representation does not destroy the semantic meaning for $\mathcal{T}$.

3.3.1 Extracting Textual Embedding

Let $x = \{w_1, ..., w_m\}$ be a document with m words. Attention mechanism has shown to be effective in capturing embedding of textual information w.r.t. a given task [96, 136]. It allows the model to attend to different parts of the given original document at each step and then learns what to attend based on the input document and what it has produced as embedding representation so far, as shown in Figure. 5.

We use a bi-directional recurrent neural network (RNN) to encode the given document into an initial embedding representation. RNN has been shown to be effective for summarizing and learning semantic of unstructured noisy short texts [23, 115]. we use GloVe 100d [96] to exchange each word w_i with its corresponding word vector, note that different dimensionality can be used. This process produces a matrix of text $x' \in \mathcal{R}^{m*100}$.

We employ the gated recurrent unit (GRU) as the cell type to build the RNN, which is designed in a manner to have a more persisted memory [23]. The bi-directional GRU will read the text forward and backwards, then outputs two hidden states $\mathbf{h_t^{fw}}, \mathbf{h_t^{bw}}$ and an output $\mathbf{o_t}$. We then concatenate two hidden states as the initial encoded embedding of the given original document:

$$\mathbf{H_t} = Concat(\mathbf{h_t^{fw}}, \mathbf{h_t^{bw}}) \tag{3.2}$$

After calculating the initial context vector $\mathbf{H_t}$, we seek to pinpoint specific information within the $\mathbf{H_t}$, which helps the classifier to predict the labels with higher confidence [72] we use the location-based attention layer based on the work of [72]. The attention layer calculates a vector $\mathbf{a_t}$ including a weight for each element in the $\mathbf{H_t}$, showing the importance of that element. The context vector $\mathbf{v_t}$ is calculated:

$$\mathbf{v_t} = \sum_{i=1}^{m} a_{t,i}\mathbf{H_i} \tag{3.3}$$

The vector $\mathbf{v_t}$ is then fed to a neural network classifier for the given utility task. Classification is one of the common tasks for textual data. Based on the output of the classifier and loss function, we update the three networks so that the output of the attention layer is an useful context that can be used for a utility task [102].

3.3.2 Reinforced Task-Aware Text Anonymizer

Here, I discuss the details of the second component which seeks to preserve privacy and utility.

3.3.2.1 Protecting Private-Attributes

Textual information is rich in content and publishing textual embedding representation without proper anonymization leads to privacy breach and revealing the private-attributes of an individual such age, gender and location. It is thus essential to protect the textual information before publishing it. The goal of my model is to manipulated learned embedded representation such that any potential adversary cannot infer users' private-attribute information. However, a challenge is that the text anonymizer does not know the adversary's attack model. To address this challenge, I add a private-attribute inference attacker D_P sub-component to my text anonymizer. This sub-component learns a classifier that can accurately identify the private information of users from their embedded text representations $\mathbf{v_u}$. I incorporate this sub-component to understand how the textual embedded representation should be anonymized to obfuscate the private information.

Inspired by the success of RL [52, 81, 135], I model this problem using RL to automatically learn how to anonymize the text representations w.r.t. the private-attribute inference attacker. In my RL model, one agent is trained to change a randomly selected text embedding representation. Then, the agent keeps interacting with the environment and changes the text embedding accordingly based on its current state and received rewards so that the private-attribute inference attacker cannot correctly identify user's private-attribute information given his embedding. This part defines the main four parts of RL environment in my problem, i.e., environment, state, action and reward.

- **Environment:** Environment in my problem includes the private-attribute inference attackers D_P and the text embedding $\mathbf{v_u}$. Note that D_P is trained beforehand.
- **State:** State describes the current situation. Here, state is the current text embedding vector $\mathbf{v_{u,t}}$ which reflects the results of the agents' actions on $\mathbf{v_u}$ up to time t.

- **Actions:** Action is define as selecting one element such as $v_{u,k}$ in text embedding vector $\mathbf{v_u} = \{v_{u,1}, ..., v_{u,m}\}$ and changing it to a value near -1, 0 or 1. This results in $3.m$ actions where m is the size of the embedding vector.

  Changing value to near *1*: In this action, the agent changes the value of $v_{u,k}$ to a value between 0.9 to 1.0. As $v_{u,k}$ will be multiplied by a classifier's weight, the output will be the weight as is. In another word, the value $v_{u,k}$ will become important to the classifier.

  Changing value to near *0*: In this action the $v_{u,k}$ will be changed to a value between -0.01 to 0.01. This action makes $v_{u,k}$ seem neutral and unimportant to a classifier as it will result in a 0 when multiplied by a weight.

  Changing value to near *-1*: In this action, the agent changes $v_{u,k}$ to a value between -1.0 to -0.9. This action will make $v_{u,k}$ important to a classifier, but, in a negative way.

- **Reward:** Reward is defined based on how successfully the agent obfuscated the private-attribute information against the attacker so far. In particular, I defined the reward function at state s_{t+1} according to the confidence of private-attribute inference attacker C_{p_k} for private-attribute p_k given the resultant text embedding at state s_{t+1}, i.e., $\mathbf{v_{t+1}}$. Considering the classifier's input data as $\mathbf{v_u}$ and its correct label as i, I define the confidence for a multi-class classifier as the difference between the probability of actual value of the private-attribute and the minimum probability of other values of the private-attribute:

  $$C_{p_k} = Pr(l = i|\mathbf{v_u}) - \max_{j \neq i} Pr(l = j|\mathbf{v_u}) \tag{3.4}$$

  Where l indicates label. For each private-attribute attacker p_k, the confidence score C_{p_k} is within the range $[-1, 1]$. Positive value demonstrates that the attacker has predicted private-attribute accurately, and negative value indicates that the attacker was not able to infer user's private-attribute. According to this definition, the reward will be positive if action a_t has caused information hiding, and will be negative if the action a_t was not able to hide sensitive information. Having confidence of private-attribute inference attackers, reward function at state s_{t+1} is defined as:

$$r_{t+1}(s_{t+1}) = - \sum_{p_k \in D_P} C_{p_k}(s_{t+1}) \tag{3.5}$$

The reward r_t is calculated according to the state s_{t+1} which associated with the transition of agent from state s_t after applying action a_t. Note that the goal of agent is to maximize the amount of received rewards so that the mean of rewards r over time $t \in [0, T]$ (T is the terminal time) will be positive and above 0.

3.3.2.2 Preserving Utility of Text Embedding

Thus far, I have discussed how to 1) learn textual embeddings from the given original document w.r.t. the given task, and 2) prevent leakage of private-attribute information by developing a reinforcement learning environment which incorporates a private-attribute inference attacker and manipulates the initial given text embedding accordingly to fool the attacker. However, data obfuscation comes at the cost of data utility loss. Utility is defined as the quality of the given data for a given task. Neglecting the utility of the text embedding while manipulating it, may destroy the semantic meaning of the text data for the given task. Classification is one of the common tasks. In order to preserve the utility of data, we need to ensure that preserving privacy of data does not destroy the semantic meaning of the text embedding representation w.r.t. the given task. I approach this challenge by changing the agent's reward function w.r.t. the data utility. I add a utility sub-component, i.e., classifier D_U, to the reinforcement learning environment which its goal is to assess the quality of resultant embedding representation. I use the confidence of the classifier for the given task to measure the utility of embedding representation using the text embedding vector $\mathbf{v_u}$ the its correct label i.

$$C = Pr(l = i|\mathbf{v_u}) - \min_j Pr(l = j|\mathbf{v_u}) \tag{3.6}$$

The agent can then use the feedback from the utility classifier to make decision when taking actions. I thus modify the reward function in order to incorporate the confidence of utility sub-component. Reward function at state s_{t+1} can be defined as:

$$\begin{aligned} r_{t+1}(s_{t+1}) =& \alpha C_{D_U}(s_{t+1}) - \\ & -(1-\alpha) \sum_{p_k \in D_P} C_{p_k}(s_{t+1}) - \mathcal{B} \end{aligned} \tag{3.7}$$

where C_{D_U} and C_{p_k} represent the confidence of utility sub-component and private-attribute inference attacker, respectively. Moreover, $\mathcal{B}$ demonstrates a baseline reward which forces the agent to reach a minimum reward value. The coefficient α also control the amount of contribution from both private-attribute inference and utility sub-components in the Eq. 3.7.

3.3.3 Optimization Algorithm

Given the formulation of states and actions, the agent aims to learn the optimal strategy via manipulating text representations w.r.t. the private-attribute attackers and utility sub-component feedbacks. It manipulate the text embeddings by repeatedly choosing an action a_t given current state s_t, and then applying actions on current state to transit to the new one s_{t+1}. The agent then receives reward r_{t+1} as a consequence of interacting with the environment. The goal of agent is to manipulate text embedding $v_{u,k}$ in a way that maximizes its reward according to Eq. 3.7. Moreover, the agent updates its action selection policy $\pi(s)$ so that it can achieve the maximum reward over time.

RLTA uses Deep Q-Learning which is a variant of Q-Learning. In this algorithm the goal is to find the following function:

$$Q^*(s_t, a_t) = \mathbb{E}_{s_{t+1}}[r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a')] \tag{3.8}$$

where $Q(s, a)$ corresponds to the Q-function for extracting actions and it is defined as the expected return based on state s and action a. Moreover, $Q^*(s, a)$ denotes the optimal action-value Q-function which has the maximum expected return using the optimal policy $\pi(s)$. Rewards are also discounted by a factor of γ per time step. The agent keeps interacting with the environment till it reaches the terminal time T.

Since it is not feasible to estimate $Q^*(s, a)$ in Eq.3.8, we use a function approximator to estimate the state-action value function $Q^*(s, a) \approx Q(s, a; \theta)$. Given neural networks as excellent function approximators [28], we lverage a deep neural network function approximator with parameters θ, or a Deep Q-Network (DQN) [81] by minimizing the following:

$$L(\theta) = \mathbb{E}_{s_t, a_t, r_{t+1}, s_{t+1}}[(y - Q(s, a; \theta))^2] \tag{3.9}$$

in which y is the target for the current iteration:

$$y = \mathbb{E}_{s_{t+1}}[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^p)] \tag{3.10}$$

θ^p is the parameters from the previous iteration.

We update the DQN according to the derivation of Eq. 3.9 with respect to the parameter θ:

$$\begin{aligned}\nabla_\theta L(\theta) = &\mathbb{E}_{s_t,a_t,r_{t+1},s_{t+1}}[(r \\ &+\gamma \max_{a'} Q(s_{t+1}, a'; \theta^p) \\ &-Q(s_t, a_t; \theta))\nabla_\theta Q(s_t, a_t; \theta)]\end{aligned} \tag{3.11}$$

3.4 Experiments

Experiments are designed to answer the following questions: **Q1**(*Privacy*): How well RLTA can obscure users' private-attribute information? **Q2**(*Utility*): How well RLTA can preserve utility of the textual data w.r.t. the given task? **Q3**(*Privacy-Utility Relation*): How does improving user privacy affects loss of utility?

To answer the first question (**Q1**), I investigate the robustness of resultant text embedding against private-attribute inference attacks, considering two private-attribute information: location and gender. To answer the second question (**Q2**), I report experimental results w.r.t. a well-known task, sentiment analysis. Sentiment analysis has many applications in user-behavioral modeling and Web [142]. In particular, RLTA predicts sentiment of the given textual embedding. To answer the final question (**Q3**), I examine the privacy improvement against utility loss.

3.4.1 Dataset

We use a real-world dataset from Trustpilot [48]. This dataset includes user reviews along with users private-attribute information such as location and gender. I remove non-English reviews based on LANGID.py[1] [71] and only keep reviews classified as English. Then, I consider English reviews associated with location of US and UK and create a subset of data with $10k$ users. Each review is associated with a rating score. I consider the review's sentiment as positive if its rating score is $\{4, 5\}$ and consider it as negative if rating is $\{1, 2, 3\}$

3.4.2 Implementation Details

For extracting the initial textual embedding, I use a bi-directional RNNs which their hidden sizes are set to 25. This makes the size of the final hidden vector $\mathbf{H_t}$ as 50. I also use a logistic regression with a linear network as the classifier in the attention mechanism. We use a 3-layer network for the Deep Q-network, i.e., input, hidden and output layers. Dimensions of the input and hidden layers are set to 50 and 700, respectively. Dimension of the last layer, i.e., output, is also set as 150. This layer outputs the state-action values which I execute the action with the best value.

For each of the private-attribute attackers and utility sub-components, I use feed-forward network with a single hidden layer with dimension of 100 which gets the textual embedding as input and uses the Softmax function as output.

We first train both private-attribute inference attacker D_P and utility sub-component D_U on the training set. These sub-components do not change after that. Then, I train an agent on each selected data for 5000 episodes. The reward discount for agents is $\gamma = 0.99$ and batch size $b = 32$. I also set the terminal time $T = 25$. I run RLTA for 5 times and select the best agent based on the cumulative

[1] https://github.com/saffsd/langid.py

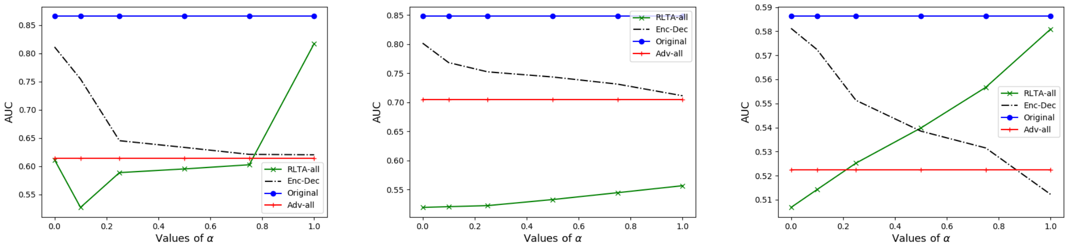


Figure 6. AUC Scores for Private-attribute (Gender and Location Inference from Left to Right) and Sentiment Prediction Tasks

reward. We also vary α as $\alpha = \{0, 0.25, 0.5, 0.75, 1\}$. The higher values of α indicate more utility contribution in RLTA.

3.4.3 Experimental Design

We use 10-fold cross validation of RLTA for evaluating both private-attribute inference attacker and an utility task with the following baselines:

- **Original:** This baseline is a variant of proposed RLTA which does not change the original user text embeddings $\mathbf{v_u}$ and publishes it as is.
- **Adv-all:** This adversarial method has two main components, i.e., generator and discriminator, and creates a text representation that has high quality for a given task, but has poor quality for inferring private-attributes [66].
- **Enc-Dec:** Using an auto-encoder is one of the effective methods to create a text embedding [90]. I modify this simple method to create a privacy-preserving text embedding. This method gets the original text x and outputs a re-constructed text $\bar{x}$. The following loss function is used to train the model. After training, encoder's output is used as the text representation $\mathbf{v_u}$ [23].

$$loss = -\sum_{x \in X} \log Pr(\bar{x}|x) + \alpha((\sum_{p_k} C_{p_k}) - C_{D_U}) \qquad (3.12)$$

In which α is the privacy budget.

To examine the privacy of final text embedding, we apply the trained private-attribute attacker sub-component D_P to the output of each method to evaluate the users' privacy. We consider two private attributes, i.e., location and gender. Then the attacker's AUC is calculated. Lower attacker's AUC indicates that textual embeddings have higher privacy after anonymization against the private-attribute inference attacker. I also report experimental results w.r.t. the utility. In particular, sentiment (positive and negative) of the given textual embedding is predicted by applying trained utility sub-component D_U to the resultant text embedding from test set for each method. I then compute AUC score for sentiment prediction task. Higher values of AUC demonstrate that the utility of textual embedding has been preserved.

3.4.4 Experimental Results

We answer the three question **Q1**, **Q2** and **Q3** to evaluate my proposed method RLTA.We use a natural language processing task, sentiment prediction, using a three layer neural network.

Privacy (Q1). Figure. 6 (a-b) demonstrates the results of private-attribute inference attack w.r.t. gender and location attributes. The lower the value of AUC is, the more privacy user has in terms of obscuring private attributes. I also report the performance of RLTA for different values of α.

We observe that ORIGINAL is not robust against private-attribute inference attack for both gender and location attributes. This confirms leakage of users private information from their textual data. Moreover, RLTA has significantly lower AUC score for both gender and location attributes in comparison to other methods. This demonstrates the effectiveness of RL for obfuscating private attributes. In RLTA, the AUC score for private-attribute inference attack increases for both attributes with the increase of α which shows the degradation in user privacy. The reason is because of the fact that agent pays less attention to privacy by increasing the value of α.

In the ENC-DEC method, as the value of α increases, the encoder tries to generate a text representation that is prune to inference attacks but it does not lose its utility w.r.t. the given task D_U. The results show that as α increases, the AUC of inference attackers will decrease.

Utility (Q2). To answer the second question, I investigate the utility of embeddings w.r.t. sentiment prediction. Results for different values of α are demonstrated in Figure. 6(c). The higher the value of the AUC is, the higher utility is preserved.

The ORIGINAL approach has the highest AUC score which shows the utility of the text embeddings before any anonymization. I observe that the results for RLTA is comparable to the ORIGINAL approach which shows that RLTA preserves the utility of text embedding. Moreover, RLTA outperforms ADV-ALL which confirms the effectiveness of reinforced task-aware text anonymization approach in preserving utility of the textual embeddings. I also observe that the AUC of RLTA w.r.t. sentiment prediction task increases with the increase of value of α. This is because with the increase of α, the agent pays more attention to the feedbacks of utility sub-component.

We also observe a small utility loss after applying RLTA when $\alpha = 1$. This is because the agent keeps changing the text embedding until it reaches the terminal time. These changes result in loss of utility even when the $\alpha = 1$.

Finally, in the ENC-DEC method, as both utility and attackers have the same

Table 1. Impact of Different Private-attribute Inference Attackers

Method	Location	Gender	Utility
Original	84.77	86.54	58.57
Enc-Dec	71.55	58.35	53.78
Adv-all	70.37	57.15	52.15
RLTA	53.34	56.41	54.83
RLTA-gen	56.64	**55.02**	**56.67**
RLTA-loc	**52.04**	56.64	54.13

importance, trying to preserving privacy would result in huge utility loss as I increase the value of α.

Privacy-Utility Relation (Q3). Results show that the Original achieves the highest AUC score for both utility task and private-attribute inference attack. This shows that Original has the highest utility which comes at the cost of significant user privacy loss. However, comparing results of privacy and utility for $\alpha = 0.5$, I observe RLTA has achieved the lowest AUC score for attribute inference attacks in comparison to other baselines, thus has the highest privacy. It also reaches the higher utility level in comparison to the Adv-all. RLTA also has comparable utility results to the Original approach. I also observe that increasing the α reduces the performance of RLTA in terms of privacy but increases its performance for utility. However, with $\alpha = 1$, RLTA preserves both user privacy and utility in comparison to Original, Enc-Dec, and Adv-all.

3.4.4.1 Impact of Different Components

Here, I investigate the impact of different private-attribute inference attackers. I define two variants of my proposed model, RLTA-Gen and RLTA-Loc. In each of these variants, I train the agent in RLTA w.r.t. the one of private-attribute attackers, e.g., RLTA-Gen is trained to solely hide *gender* attribute. For this experiment I set $\alpha = 0.5$ as in this case privacy and utility sub-components contribute equally during training phase (Eq. 3.7). Results are shown in Table 1.

RLTA-Loc and RLTA-Gen have the best performance amongst all methods in obfuscating location and gender private-attributes, respectively. Results show that using RLTA-Loc could also help improve privacy on gender and likewise for (RLTA-Gen) in comparison to other approaches.

RLTA-Gen performs better in terms of utility, in comparison to RLTA which incorporates both gender and location attackers. Moreover, results show that both RLTA-Gen and RLTA-Loc have better utility than other baselines.

To sum up, these results indicate that although using one private-attribute attacker in the training process can help in preserving more utility, it can compromise obscuring other private attributes.

Parameter Analysis: Our proposed method RLTA has an important parameter α to change the level of privacy and utility. I illustrate the effect of this parameter by changing it as $\alpha \in \{0.0, 0.1, 0.25, 0.5, 0.75, 1.0\}$. According to the Figure 6, when the α parameter increases, the privacy loss will decrease, but, the utility loss will increase. This shows the utility and the privacy have an association with each other. Hence, the more privacy loss decreases the more utility loss increases. Choosing the right value for α depends on the application and usage of this method. According to the results, choosing $\alpha = 0.5$ would result in a balanced privacy-utility. In some applications where the privacy of users are important and critical, I can set the α parameter above 0.5. On the other hand, if the user's privacy is not a top priority, this parameter can be set to a lower value than 0.5 which although it does not protect users' private attribute as well as when $\alpha >= 0.5$, it does protect users' private attribute at a reasonable level.

3.5 Conclusion

In this work, I propose a deep reinforcement learning-based text anonymization, RLTA, which creates a text embedding that does not leak the user's private-attribute information while preserving its utility w.r.t. a given task. RLTA has two main components: (1) an attention-based task-aware text representation learner, and (2) a deep RL-based privacy and utility preserver. Our results illustrate the effectiveness of RLTA in preserving privacy and utility. One future direction is to generate privacy-preserving text rather than embeddings.

Chapter 4

PRIVACY: TOWARD PRIVACY AND UTILITY PRESERVING IMAGE REPRESENTATION

4.1 Background

Security and surveillance systems, such as those found in private industries (e.g., biometric access control systems) and public domains (e.g., face recognition systems at airports and traffic thruways), acquire images of people's faces for verification and identification tasks. The ease of collecting data on private citizens raises concerns about violating privacy-preserving contracts or expectations [56], because organizations have been known to exercise their prerogative to sell information on individuals or have been subject to malicious attacks that compromised users' privacy [83, 6]. For instance, in 2019, a malicious cyber-attack on a US Customs and Border Protection subcontractor exposed travelers' photos [75]. Due to such threats to individual liberties and privacy, one method that has been proposed to protect an individual's private information is to anonymize it before sharing. While some recent studies have shown that face images contain user-related private information such as gender or race [42, 76, 73], research on protecting face images from adversarial attacks has been limited [17].

Two general approaches have been proposed for preserving the privacy of image data. The first method, known as visual privacy, perturbs images so that a human cannot infer a user's private attributes. Text representations have many hidden attributes which can be used for different sentiment analysis tasks [44, 45, 43]. Similar to the text representations, image representations can also have both useful and sensitive attributes. Thus, in order to protect images, the second method creates a representation from the image data [17], which then replaces the original image in image-based applications. An advantage of visual privacy is that the perturbed images can easily be used in various image-based tasks. This approach, however, does not provide the same level of privacy-preserving effectiveness as the second method [17]. Furthermore, current methods do not guarantee that the perturbed image is still useful in a specific utility task.

In this approach, I address the challenge of creating a privacy-preserving image representation while simultaneously preserving the image's utility for a given image-based task. To address this challenge, I propose the AIA (Adversarial Image Anonymizer) framework composed of three main modules: (1) a component to encode images for representation learning; (2) a component for privacy-preserving representation learning; and (3) a component to preserve the utility of the learned representations. The main contributions of this dissertation are (1) the study of the novel problem of joint privacy and utility-preserving image representation; (2) a principled framework (AIA) that integrates adversarial learning and a generative model to create privacy-preserving image representations which can be used with a given utility task; and (3) extensive experiments on a publicly available data set to demonstrate the effectiveness of AIA for creating both privacy-preserving image representations for a specific utility task.

4.2 Problem Statement

Let $\mathcal{X} = \{x_1, x_2, ..., x_N\}$ denote a set of N grey-scaled images where each image x_i is composed of a matrix $x_i \in IR^{1 \times n \times m}$. Let p denote a private attribute that users do not want to disclose such as gender. I address the following problem:

Given a set of images $\mathcal{X}$ and private attribute p, learn an anonymizer f that can learn an image representation $\mathbf{g} \in IR^{1 \times k}$ such that: (1) [Privacy preservation] an adversary cannot infer the targeted user's private attribute p, and (2) [Utility preservation] the image representation $\mathbf{g}$ can be effectively used in a given task $\mathcal{T}$ such as 1-to-1 face matching. The problem can be expressed as:

$$\mathbf{g_i} = f(x_i, p, \mathcal{T}) \tag{4.1}$$

Due to the success of auto-encoders in learning image representations [74], I use auto-encoders with adversarial training to create both utility and privacy-preserving image representations.

4.3 Proposed Method

The main components of the Adversarial Image Anonymizer (AIA) framework appear in Figure 7. The input to the system is a grey-scale image, while the output is a privacy-preserved vector $\mathbf{g} \in IR^{1 \times k}$. The model has 3 main components: (1) an auto-encoder composed of an encoder that learns the input image representation and a decoder that can reconstruct the input image given its representation, (2) an adversary which tries to infer the user's private attribute using the image representation, and (3) a task $\mathcal{T}$ that is used to preserve the utility of the image representation. In this

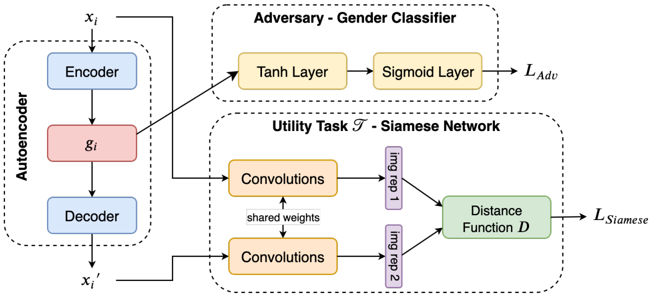


Figure 7. Architecture of Adversarial Image Anonymizer

framework, I first train each component individually and, then, use an adversarial loss function to enhance the overall model. I present the details of my proposed method next.

4.3.1 Learning Image Representations

The goal of this component is to generate an image representation that can be used in different tasks. The image representations in my model are created using an auto-encoder. This is a generative model that is trained in an unsupervised way to learn latent representations of the input images. A key feature of auto-encoders is their dimensionality reduction ability which was an important reason to integrate them instead of using GANs. Another reason for using an auto-encoder instead of a GAN is to prevent some of the issues in GAN-based models such as stability problems and time-consuming training [99]. The auto-encoder consists of the following components:

Encoder. The encoder learns a latent representation of the input image and aims at reducing its dimensionality. x_i is a grey-scaled image with dimensions $n \times m$. To create a representation of the image, Convolutional Neural Networks (CNN) are used to learn filters that can identify the important parts of an input image. In the model, a convolution layer $Conv$ is first used to create feature maps from the input image. Then, an activation function F is applied on the gathered features and a pooling layer P is used. The pooling layer helps to select the parts of the features with strong correlation to the input image. The output of these three layers is referred to as a block L_j^{enc} where j represents the j^{th} encoder block:

$$L_j^{enc} = P(F(Conv(x))) \tag{4.2}$$

A stack of these blocks is used to create an image representation. Observe that in order to convert the output of the final pooling layer to a vector $\mathbf{g}$, a flattening layer that converts a matrix into a vector is used:

$$\mathbf{g} = Flatten(L_J^{enc}) \tag{4.3}$$

where J is the index of the final encoder's layer.

Decoder. The decoder tries to reconstruct the input image using the previously generated representation. The process of decoding the image representation $\mathbf{g}$ is similar to the one in the encoding phase but in reverse. First, a convolution layer is used to create feature maps, then an activation function is applied to the features. As for the final layer, instead of using a pooling layer which generates a smaller matrix, an up-sampling layer U is used to increase the size of the reconstructed image:

$$L_l^{dec} = U(F(Conv(\mathbf{g}))) \tag{4.4}$$

where L_l^{dec} indicates the l^{th} decoder block. The final output of the decoder is the reconstructed image $\mathbf{x}'$ which is then used to train the auto-encoder.

After creating the auto-encoder, I train it using the input image x_i and the reconstructed image x'_i. This will result in learning a representation vector $\mathbf{g}$ which is expected to capture useful information of the input image.

4.3.2 Adversarial Training

Creating an image representation using an auto-encoder alone could result in privacy issues [17]. For example, a well-trained gender classifier could predict the gender of a person using the corresponding image representation. To prevent this issue, adversarial training is integrated to create privacy-preserving image representations. In this component, a powerful adversary, i.e. gender classifier, is used to further improve the learned representation of the auto-encoder. Because the goal is to create a privacy-preserving representation for a given task $\mathcal{T}$, the loss value of this task is used as a penalty to generate private learned representation.

4.3.3 Optimization Algorithm

The training process in the proposed model consists of two parts. In the first part, each component is trained (auto-encoder, adversary, and the given task $\mathcal{T}$) separately:

- *Auto-encoder*: For the encoder component, stacked CNN blocks are used, each containing a 2D convolution layer with leaky ReLU as the activation function; and an average pooling which operates on the output of the activation function. The decoder has also two stacked CNN blocks. Each block has a convolution layer with a leaky ReLU activation function. The output of the activation function goes through an up-sampling layer to create an output image with similar size to the input image. The auto-encoder is trained using the Binary Cross Entropy loss function between the input image x_i and the reconstructed image x'_i. This loss function calculates how well the auto-encoder has predicted the image:

$$L_{AE} = x' \cdot \log x + (1 - x') \cdot \log(1 - x) \tag{4.5}$$

- *Adversary*: In this model, I use a high-quality gender classifier as the adversary. The adversary acquires an image representation $\mathbf{g}$ as input and predicts whether the image corresponds to a female or male face. I use a three-layer neural network to output gender probablity $\mathbf{o}$:

  $$\mathbf{o} = sigmoid(\mathbf{W_A^{(2)}}(\tanh(\mathbf{W_A^{(1)}}\mathbf{g} + \mathbf{b_A^{(1)}})) + \mathbf{b_A^{(2)}}) \tag{4.6}$$

  where $\mathbf{W_A^{(.)}}$, $\mathbf{b_A^{(.)}}$, are learnable weights. This classifier is also trained using the same Binary Cross Entropy (BCE) loss L_{Adv}.

- *Task $\mathcal{T}$*: I consider a 1-to-1 face matching task, which verifies if two input images are of the same individual. I use the well-known Siamese network for this task which acquires two images x_i and x'_i as input and returns their representations. The distance of the two representations is then calculated based on the similarity between x_i and x'_i. I train this model using the following loss function:

  $$L_{Siamese} = (1 - y)\frac{1}{2}D^2 + y\frac{1}{2}max(0, m - D)^2 \tag{4.7}$$

  where D as the Euclidean distance and m a constant margin.

Adversarial Training: After training each component separately, I use the following loss function to enhance the autoencoder for generating privacy and utility preserving representations based on the feedback from the utility and the adversary components:

$$L_{Total} = L_{AE} + \alpha L_{Siamese} - \beta L_{Adv} \tag{4.8}$$

where α and β indicate the contribution of each loss value. In my model, I use a powerful attacker to ensure privacy even from other unseen attackers. In my experiments I show that my model can preserve privacy of user's private attributes from different attackers.

4.4 Experiments

We performed multiple experiments to evaluate the performance of the proposed model. I aim to answer the following questions: (**Q1**) how well does my method protect users' private attribute, i.e., gender?; (**Q2**) how well does my method preserve the utility of an image with respect to a given task $\mathcal{T}$?; and (**Q3**) what is the relation between privacy and utility? To answer **Q1**, I use an adversary to test if it can detect gender based on the perturbed representations. For **Q2**, I study the performance of the utility task before and after perturbing the learned image representations. Finally, to answer **Q3**, I study how the effectiveness of preserving users' privacy impacts the utility of the learned representations.

4.4.1 Data

In this study, I use two different publicly-available datasets, CelebA [70] and VGG face datasets [93]. CelebA consists of over 200K celebrity images with various metadata [70]. VGG consists of 2,622 identities where each identity has different images [93]. In both datasets, I use gender as the private attribute.

4.4.2 AIA Implementation Details

The adversary is a gender classifier which has 6 convolution layers with 32, 64, 64, 128, 128, and 256 channels, respectively. After the convolution layers, I use a one-layer neural classifier with Softmax on the output layer. The auto-encoder is composed of an encoder and a decoder. The encoder has two convolution layers with 8 and 12 channels, while the decoder contains three convolution layers with $256, 128$, and 1 channels. The final convolution layer in the decoder converts the 128 features into 1 feature to generate the image corresponding to the associated input image. Finally, for the utility task $\mathcal{T}$, I use a Siamese network. This network has three convolution layers and three fully connected layers after flattening the output of the convolution layers. The three convolution layers have $4, 8$ and 8 channels, respectively. The fully connected network has an input layer, a hidden layer with 500 neurons, and an output layer with 128 neurons. The output of this network is an embedded image representation which is then used to calculate the distance between two input images.

4.5 Experimental Evaluation

We use the following baseline methods for comparision:

- **Original:** this method does not use any anonymization and only outputs the learned image representation from the auto-encoder's output.
- **Random:** this method randomly changes 50% of each learned image representation to make it private. Because each value in the image representation is within $[0.0, 1.0]$, I randomly select 50% of the numbers and change each to a random number sampled from $[0.0, 1.0]$.

- **AIA$\backslash$T:** this is a modified version of my proposed method that does not use the adversarial training for preserving the utility of the learned image representation.

4.5.1 Experimental Results

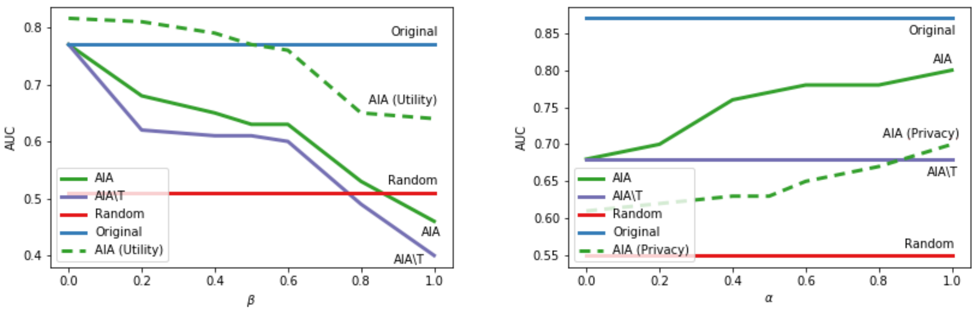
In this subsection, I evaluate AIA's performance using a gender classifier (adversary) and a similarity task (utility).

Privacy (Q1). Table 2 compares the accuracy of the different methods for gender-detection and the utility task, where $\alpha = \beta = 0.5$. In the Gender Privacy column, lower accuracy indicates that the gender classifier for that method was less effective at predicting a user's gender based on the image representations. I observe that AIA is better at protecting privacy than the original approach. While AIA cannot preserve users' privacy as well as AIA$\backslash$T and the random methods, it preserves utility significantly better than these methods. The random method, despite being more effective at privacy preservation, generated significantly lower utility, implying that it generated useless representations.

Table 2. Accuracy of Private-attribute and Utility Classifiers

	CelebA Dataset		VGG Dataset	
Method	Privacy (↓ better)	Utility (↑ better)	Privacy (↓ better)	Utility (↑ better)
Original	%78.01	%88.87	%81.21	%91.31
Random	%52.12	%56.89	%51.34	%53.67
AIA$\backslash$T	%62.53	%69.34	%65.67	%66.29
AIA	**%64.96**	**%78.64**	**%68.13**	**%77.96**

Utility (Q2). As illustrated in Table 2, my method performs better than the Random and AIA$\backslash$T methods for preserving the utility of the image representations. While the Original method has the highest accuracy level, it provides the worst privacy-



Gender Inference (↓ better) **Utility (↑ better)**

Figure 8. AUC Scores for Private-attribute Inference and Utility Tasks for Different Values of α and β

preserving guarantees. This highlights the need for changing image representations in order to preserve users' privacy. Changing image representations randomly will perverse users' privacy but it will also greatly decrease their utility. While AIA$\setminus$T is relatively effective preserving privacy, it lacks the utility benefits that AIA provides. This is because AIA$\setminus$T does not have any component that forces the auto-encoder to preserve the utility of image representations.

Privacy Utility Trade-off (Q3). AIA has two main parameters, β controls the contribution of the gender classifier, while α controls the contribution of the utility task. In Figure 8 I show the performance of my model and the baselines for different values of α and β using the CelebA dataset. For each parameter, I hold one of them as 0.5 and vary the other one from 0 to 1. As shown in this figure, achieving higher levels of utility with AIA results in lower levels of privacy assurance and vice versa. Figure 2.a shows AIA's utility level using a dashed line for different values of β while $\alpha = 0.5$. I can observe that using a β value larger than α could result in substantial utility loss. Figure 2.b shows the inverse. The dashed line in this figure shows the gender inference AUC values (where higher values correspond to lower privacy) for different values of α while $\beta = 0.5$. These results indicate that reaching higher levels

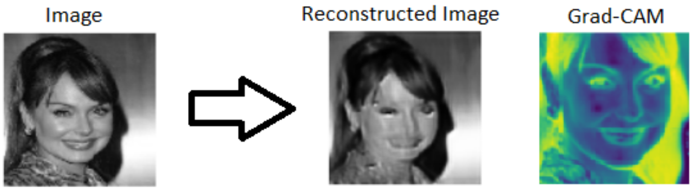


Figure 9. Adversarial Image Anonymizer's Visualization Results

of utility can result in significant privacy loss. The problem of utility and privacy preservation is consequently a multi-objective, trade-off optimization problem where each objective antagonizes the other. From Figure 8, I conclude that using similar values for α and β provides a reasonable balance between both privacy and utility. In general, a judicious choice for the two tuning parameters depends on the application domain. If privacy is more important than utility, β should be higher than α, and vice versa.

4.5.2 Visualization

To gain additional insights, I visualized and analyzed the learned image representations using a low value of $\alpha = 0.2$. I used Grad-CAM [112], a well-known CNN visualization technique. Given a label and an image, this method generates an activation map that can be used to identify the areas of the image that are most relevant to the label. Figure 9 shows both the reconstructed and Grad-CAM images for a sample photo and the result of the gender classifier. The reconstructed image in Figure 9 shows the outcome of the anonymization process. I used the decoder component from the auto-encoder to reconstruct the image using its representation. The Grad-CAM image shows the important parts of the image that resulted in the classifier predicting a female label. I can observe that the classifier focuses on the hair length and the eyes, as well as the location of the cheekbones and the shape of the chin. The adversarial training in AIA influences the auto-encoder to hide these pieces of information that result in more accurate gender labels.

4.6 Conclusion

Protecting the privacy of citizens has been a widespread concern in an era where the intentional or unintentional propagation of private information has been an unfortunate byproduct of machine learning. I recognized that limited attention has been given to mechanisms that simultaneously preserve an individual's privacy while preserving the intended utility of a machine or deep learning model. Thus, I proposed the AIA framework that uses an auto-encoder to learn image representations and then enhances these representations using adversarial training. Initial results showed an interesting trade-off between utility and privacy which results in outcomes that offer better privacy while having only a small impact on utility. The performance results showed that AIA performed well overall in comparison to the baseline methods.

Chapter 5

FAIRNESS: MITIGATING BIAS IN SESSION-BASED CYBERBULLYING DETECTION

5.1 Background

Cyberbullying is often characterized as a *repeated* rather than a one-off behavior [125]. This unique trait has motivated research that focuses on the detection of cyberbullying in entire *social media sessions.* In contrast to a single text, e.g., a Facebook comment or a tweet, a social media session is typically composed of an initial post (e.g., an image with a caption), a sequence of comments from different users, timestamps, spatial location, user profile information, and other social content such as number of likes [21]. Session-based cyberbullying detection presents a number of characteristics such as multi-modality and user interaction [21]. In this work, because my goal is to mitigate bias in natural language, I focus on text (i.e., a sequence of comments) in a social media session. Session-based cyberbullying detection is defined as follows:

5.2 Problem Statement

Cyberbullying Detection in a Social Media Session: considering a corpus of N social media sessions $\mathcal{C} = \{\int_1, \int_2, ..., \int_N\}$, in which each session consists of a sequence of comments denoted as $\{\mathbf{c}_1, ..., \mathbf{c}_C\}$. A session is labeled as either $y = 1$ denoting a bullying session or $y = 0$ denoting a non-bullying session. Let D be the dimension of extracted textual features (e.g., Bag of Words) $\mathbf{x}_i$ for $\mathbf{c}_i$. Session-based cyberbullying detection aims to learn a binary classifier using a sequence of textual data to identify if a social media session is a cyberbullying instance:

$$\mathcal{F} : \{\mathbf{x}_1, ..., \mathbf{x}_C\} \in \mathbb{R}^D \rightarrow \{0, 1\}. \tag{5.1}$$

5.3 Proposed Method

An unbiased model for cyberbullying detection makes decisions based on the semantics in a social media session instead of sensitive triggers potentially related to cyberbullying, such as "gay," "black," or "fat." In the presence of unintended bias, a model may present high performance for sessions with these sensitive triggers without knowing their semantics [32]. In this section, I first discuss how to define and assess bias in the context of session-based cyberbullying detection. I then present the details of my bias mitigation strategy.

5.3.1 Assessing Bias

Bias in a text classification model can be assessed by the *False Negative Equality Difference* (FNED) and *False Positive Equality Difference* (FPED) metrics, as used in previous studies such as [144, 37, 50]. They are a relaxation of *Equalized Odds* [10] and defined as

$$\text{FNED} = \sum_z |\text{FNR}_z - \text{FNR}_{overall}|, \tag{5.2}$$

$$\text{FPED} = \sum_z |\text{FPR}_z - \text{FPR}_{overall}|, \tag{5.3}$$

where z denotes cyberbullying-sensitive triggers, such as “gay,” “black,” and “Mexican.” The complete list of sensitive triggers can be found in Appendix A. FNR$_{overall}$ and FPR$_{overall}$ denote the False Negative Rate and False Positive Rate over the entire training dataset. Similarly, FNR$_z$ and FPR$_z$ are calculated over the subset of the data containing the sensitive triggers. An unbiased cyberbullying model meets the following condition:

$$P(\hat{Y}|Z) = P(\hat{Y}), \tag{5.4}$$

where $\hat{Y}$ stands for the predicted label. 5.4 implies that $\hat{Y}$ is independent of the cyberbullying-sensitive triggers Z –that is, a debiased model performs similarly for sessions with and without Z.

Note that the widely-used non-discrimination evaluation sets – Identity Phrase Templates Test Sets (IPTTS) [32] – are not applicable to my task. IPTTS are generated by predefined templates with slots for specific terms, e.g., “I am a boy” and “I am a girl.” They only include examples for single text, whereas a social media session includes a sequence of comments. As shown in subsection 5.1, the average number of comments in the Instagram dataset is 72, which can pose great challenges for generating synthetic social media sessions and the labeling process.

5.3.2 Mitigating Bias

Essentially, a debiasing session-based cyberbullying detection is a sequential decision-making process where decisions are updated periodically to assure high performance. In this debiasing framework, comments arrive and are observed sequentially. At each timestep, two decisions are made based on the feedback from past decisions: (1) predicting whether a session is bullying and (2) gauging the performance

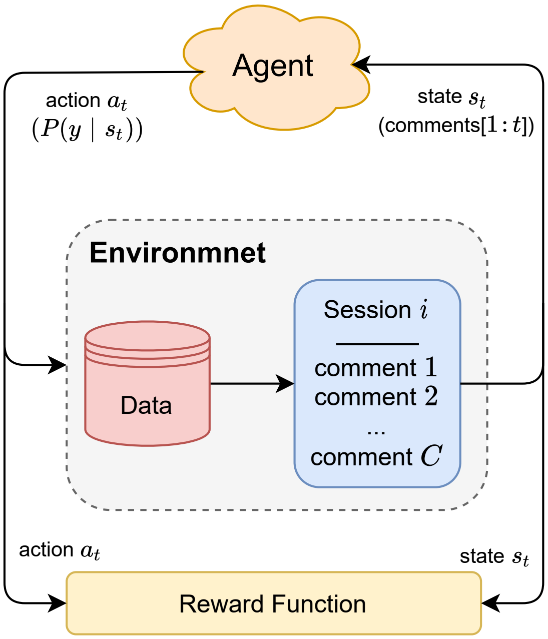


Figure 10. Overview of Proposed RL-based Session Classifier

differences between sessions with and without sensitive triggers. My debiasing strategy is built on the recent results of RL [117, 150, 83], particularly, the sequential Markov Decision Process (MDP). In this approach, an agent A interacts with an environment over discrete time steps t: the agent selects action a_t in response to state s_t. a_t causes the environment to change its state from s_t to s_{t+1} and returns a reward r_{t+1}. Therefore, each interaction between the agent and the environment creates an experience tuple $M_t = (s_t, a_t, s_{t+1}, r_{t+1})$. The experience tuple is used to train the agent A through different interactions with the environment. The agent's goal is to excel at a specific task, such as generating text [117] or summarizing text [55].

This work leverages techniques in RL to alleviate the unintended bias when classifying social media sessions into *bullying* or *non-bullying* based on user comments. In particular, a standard classifier $\mathcal{F}$ (e.g., HAN) as an RL agent and a sequence of comments observed at time $\{1, 2, ..., t\}$ as state s_t are considered. The agent selects an action $a_t \in \{\text{non-bullying}, \text{bullying}\}$ according to a policy function $\pi(s_t)$. $\pi(s_t)$ indicates the probability distribution of actions a in response to state s_t, whereas $\pi(s_t, a_t)$ shows the probability of choosing action a_t in response to state s_t. The action can be interpreted as the predicted label $\hat{y}$ using the input comments. The reward r_{t+1} is then calculated for the state-action set (s_t, a_t) and the cumulative discounted sum of rewards G_t is used to optimize the policy function $\pi(s_t)$.

Below, I provide details of the (1) environment, (2) states, (3) actions, and (4) the reward function for the proposed debiasing approach.

- *Environment* is a session comments loader. At each episode, the environment chooses a single session and returns its first t comments as state s_t. As such, states are independent from the agent's actions, as they do not affect the next state. When it reaches the maximum number of comments of the selected session C, the process is terminated.
- State s_t is a sequence of comments in a social media session posted by various users from time 1 through time t.
- Action a_t determines a session to be *bullying* or not, given the input comments or state s_t:

$$a_t \in \{\textit{bullying}, \textit{non-bullying}\}. \tag{5.5}$$

- Reward function R is used to optimize the policy function $\pi(s_t, a_t)$. It is defined based on how successfully the agent predicts the label for the input state s_t and how much bias the classifier currently has. The bias of a classifier is defined as the harmonic mean of FPED and FNED characterized by the sensitive triggers in cyberbullying. In a debiased classifier, both FPED and FNED are expected to be close to zero. The reward function R is defined as:

$$R = -l_{\mathcal{F}} - \beta \times \frac{2 \times \text{FPED} \times \text{FNED}}{\text{FPED} + \text{FNED}}, \tag{5.6}$$

Algorithm 1 Optimization Algorithm of RL-based Session Classifier

```
Require: The dataset $\{\mathbf{x}, z, y\}$, initialized $\pi_\theta(s_0, a_0)$, discount rate $\gamma$, balancing
    weight $\beta$, learning rate $lr$, number of episode $E$.
while Episode $e < E$ do
    Initialize $s_t, M$
    for $t \in \{0, 1, ..., C\}$ do
        $A$ selects action $a_t$ according to distribution $\pi(s_t)$
        $M \leftarrow M + (s_t, a_t, r_{t+1}, s_{t+1})$
        $s_t \leftarrow s_{t+1}$
        for each timestep $t$, reward in $M_t$ do
            $G_t \leftarrow \sum_{i=1}^{t} \gamma^i r_{i+1}$
        end for
        Calculate mean policy loss for all timesteps according to 5.8.
        Update the policy according to 5.7.
    end for
end while
```

where l indicates the prediction error of the classifier and β balances between prediction and the debiasing effect of $\mathcal{F}$. The reward function is calculated based on all sessions in the environment, evaluating the performance and bias of the classifier.

5.3.3 Optimization Algorithm

Given the environment, state, actions, and reward function, the optimal action selection strategy $\pi(s_t, a_t)$ is aimed to be learned. At each timestep t, a session with t comments is classified by the agent, and the reward r_{t+1} is calculated using 7.1, according to the agent's action a_t and state s_t. The goal of the agent is to maximize its reward according to 7.1. The policy gradient algorithm – REINFORCE [132] – is used to train the agent. As such, the agent possesses similar properties to a classifier, and the classifier's output distribution can be mapped to the agent's policy function $\pi(s_t, a_t)$. The following function is used to update the agent:

$$\Delta\theta = lr \nabla_\theta \mathcal{L}(\theta), \tag{5.7}$$

where lr denotes the learning rate, θ is the parameter w.r.t. the policy function $\pi_\theta(s_t, a_t)$, and $\mathcal{L}(\theta)$ indicates the policy loss:

$$\mathcal{L}(\theta) = \log(\pi_\theta(s_t, a_t) \cdot G_t), \tag{5.8}$$

where $G_t = \sum_{i=1}^{t} \gamma^i r_{i+1}$ is the cumulative sum of rewards with discount rate γ. The pseudo-code for the optimization algorithm can be seen in Algorithm 3.

5.4 Experiments

In this section, both quantitative and qualitative evaluations are conducted to examine the efficacy of the debiasing strategy. In particular, it is shown that this method can effectively mitigate the impacts of unintended data biases without impairing the model's prediction performance by answering:

(1) Can the unintended bias of machine learning models for detecting cyberbullying sessions be mitigated by leveraging techniques in RL?

(2) If so, will this debiasing strategy impair the cyberbullying detection performance? and

(3) If 'no' to (2), what is the source of gain?

5.4.1 Data.

Two benchmark datasets for cyberbullying detection – *Instagram* [47] and *Vine* [101] – are used for empirical evaluation. The number of sessions in *Instagram* and *Vine* is 2,218 and 970, respectively. Both datasets were crawled using a

Table 3. Statistics of Instagram and Vine Datasets

Datasets	# Sessions	# Bullying	# Non-bullying	# Comments
Instagram	2,218	678	1,540	155,260
Vine	970	304	666	78,250

snowball sampling method and manually annotated via the crowd-sourcing platform CrowdFlower.[2] Sessions containing less than 15 comments were removed to ensure data annotation quality. Annotators were asked to examine the image/video, associated caption, and all of the comments in a session before making the final decisions.

Instagram: Instagram[3] is a social networking site ranked as one of the top five networks with the highest percentage of users reporting experiences of cyberbullying [61]. Each social media session consists of image content, a corresponding caption, and a sequence of comments in temporal order. In total, this dataset is composed of 2,218 sessions, with an average number of 72 comments in each session.

Vine: Vine[4] was a mobile application that allowed users to upload and comment on six-second looping videos. Each social media session consists of video content, the corresponding caption, and a sequence of comments in temporal order. This dataset contains 970 sessions and each session contains, on average, 81 comments.

[2] https://www.figure-eight.com/

[3] https://www.instagram.com/

[4] https://vine.co/. It was shut down in 2017.

5.4.2 Experimental Setup

For social media sessions, standard fairness methods, such as identity swapping and data supplementation, are not applicable. I compare my approach with commonly used machine learning models for classification with sequential text data, including HAN, Convolutional Neural Network (CNN), and Gated Recurrent Unit (GRU), as well as a recent model proposed for session-based cyberbullying detection – HANCD [19]. HANCD leverages multi-task learning to jointly model the hierarchical structure of a social media session and the temporal dynamics of its sequence of comments to improve the performance of cyberbullying detection.

The state-of-the-art model *Constrained* [37] that imposes two fairness constraints on cyberbullying detection to mitigate biases is also included. In the implementation, the HANCD classifier is used as the cyberbullying model in *Constrained* for a fair comparison. The parameter w.r.t. the fairness constraints is set to 0.005, as suggested. Both HAN and HANCD use GRU to extract the context of the input data. 1-layer GRUs with a hidden size of 100 and 200 neurons for word and comment attention networks, respectively, are used. As my approach is model-agnostic, for each standard machine learning model, there is a corresponding debiased counterpart.

For the proposed method, $l_{\mathcal{F}}$ in the reward function (7.1) is computed as the cross entropy loss between the true label y and the predicted probability p:

$$l_{\mathcal{F}} = -\frac{1}{2}\sum_{i=1}^{2} y_i \log(p_i) + (1 - y_i)\log(1 - p_i). \tag{5.9}$$

In Algorithm 3, the classifier $\mathcal{F}$ is pre-trained for 5 iterations using loss function $l_{\mathcal{F}}$, learning rate $3e-3$, and the Adam optimizer [58]. $\mathcal{F}$ is then placed in the RL setting discussed in 5.3.2. The REINFORCE method is applied with $E = 500$ episodes, learning rate $1e-5$, $\beta = 1.0$, and $\gamma = 0.5$ using the Adam optimizer to further update the classifier.

Evaluations focus on both the prediction accuracy and the debiasing effect of a model. For prediction performance, standard metrics for binary classification, including Precision, Recall, F1, and AUC scores are adopted. Following [144, 37], FPED, FNED, and total bias (FPED+FNED) are used to evaluate how biased a model is w.r.t. sessions with and without sensitive triggers. Lower scores indicate less bias. For all models, pre-trained GloVe word embeddings [96] and 10-fold cross-validation with an 80/20 split are used for a fair comparison. Furthermore, I perform McNemar's test to examine whether a statistically significant difference between baseline and debiased models exists in terms of cyberbullying classification accuracy and equity. The best results are highlighted in bold font.

5.4.3 Can unintended bias be mitigated?

In this section, experimental results are shown to answer the first question: Can the proposed framework mitigate unintended bias?" As expected, the impact of the unintended bias embedded in the datasets for cyberbullying detection can be effectively mitigated by the proposed RL framework. Results for both *Instagram* and *Vine* are reported in Table 4. "De-" denotes a debiased model, e.g., De-HAN is a HAN debiased by the proposed RL framework. "Total" stands for the total bias (FPED+FNED). All McNemar's tests resulted in statistical significance with p-values < 0.05. Lower FPED and FNED indicate lower bias in the model.

We observe the following: (1) Compared to the standard classifiers, the debiased counterparts significantly improve FNED and FPED scores, indicating that my pro-

Table 4. Bias Comparisons of Different Cyberbullying Detection Models

Model	*Instagram*			*Vine*		
	FPED	FNED	Total	FPED	FNED	Total
Constrained	0.061	**0.073**	0.134	**0.018**	0.065	0.083
HAN	0.134	0.180	0.314	0.070	0.031	0.101
CNN	0.243	0.180	0.424	0.115	0.098	0.214
GRU	0.211	0.169	0.380	0.092	0.076	0.168
HANCD	0.125	0.167	0.293	0.063	0.042	0.105
De-HAN	0.057	0.078	0.135	0.020	**0.030**	**0.050**
De-CNN	0.198	0.178	0.376	0.099	0.081	0.180
De-GRU	0.116	0.156	0.272	0.072	0.035	0.107
De-HANCD	**0.050**	0.081	**0.131**	0.019	0.041	0.060

posed debiasing strategy can mitigate the unintended bias in data used for predicting cyberbullying sessions, regardless of the dataset or machine learning model. For example, when tested on *Instagram* with the HAN model, my debiasing method can decrease FPED, FNED, and total bias by 95.7%, 56.7%, and 57.0%, respectively. For *Vine*, the improvement with HAN is 71.4%, 3.3%, and 50.5%, respectively. (2) Total biases of standard classifiers come from both the FPRs and FNRs for the *Instagram* experiments, while the main contributor of biases is the FPRs for the *Vine* experiments. My approach mitigates total bias in both scenarios. (3) My debiasing strategy based on RL techniques is also more effective than the fairness constraints proposed in [37], as indicated by the decreased total biases for both *Instagram* and *Vine*. Comparing HANCD, Constrained, and De-HANCD, shows that Constrained decreases FPED by sacrificing FNED, while De-HANCD can decrease both.

In addition to the quantitative results, I provide qualitative analyses by visualizing FPED and FNED of both the standard and debiased HANCD models. In an experiment with *Instagram* for sessions containing ten sensitive triggers, as illustrated in Figure 11, it can be observed that compared to De-HANCD, HANCD is more

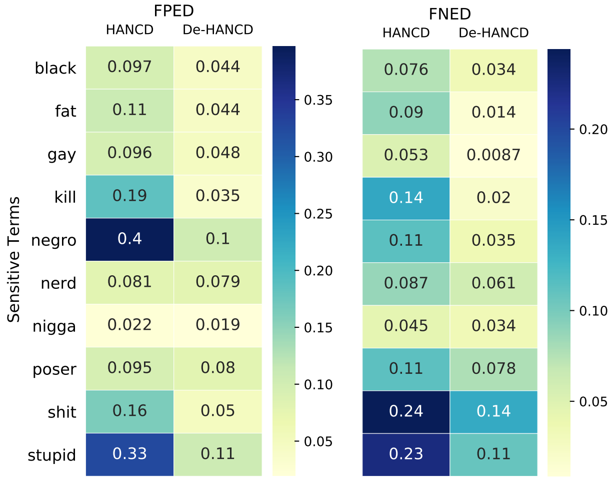


Figure 11. Comparison for Fairness Measures of HANCD and De-HANCD Methods on Instagram Dataset

biased toward some sensitive triggers, such as "fat" and "stupid." Demographic-identity related bias is also detected in HANCD. For example, sessions containing identity terms including "ne**o," "gay," and "ni**a" are more likely to be falsely identified as "bullying," as indicated by FPED. By contrast, De-HANCD mitigates various types of unintended biases and has more consistent performance across all of the sensitive triggers.

5.4.4 Is there a trade-off between accuracy and bias mitigation?

A dilemma often faced by researchers studying bias and fairness in machine learning is the trade-off between fairness and efficiency [8]. Under this trade-off theory, forcing cyberbullying classifiers to follow the proposed debiasing strategy would invariably decrease the accuracy. This section shows that, somewhat counterintuitively, my approach can outperform biased models w.r.t. overall cyberbullying detection accuracy, while also decreasing unintended biases in the data.

Table 5. Performance Comparisons of Different Models on Instagram Dataset

Model	AUC	PREC	REC	F1
Constrained	0.9042	0.8099	0.9101	0.8570
HAN	0.9032	**0.8434**	0.8879	0.8651
CNN	0.7120	0.6872	0.7380	0.7117
GRU	0.7352	0.7003	0.7265	0.7132
HANCD	0.9087	0.8218	**0.9206**	0.8684
De-HAN	0.9057	0.8292	0.9115	0.8684
De-CNN	0.7068	0.7011	0.6940	0.6975
De-GRU	0.7565	0.7355	0.7498	0.7426
De-HANCD	**0.9089**	0.8357	0.9102	**0.8714**

Table 6. Performance Comparisons of Different Models on Vine Dataset

Model	AUC	PREC	REC	F1
Constrained	0.8077	0.7644	0.8113	0.7871
HAN	0.8527	0.5203	0.8127	0.6344
CNN	0.6245	0.4603	0.7119	0.5591
GRU	0.6759	0.4801	0.7651	0.5900
HANCD	0.9223	0.6841	0.8590	0.7616
De-HAN	**0.9365**	0.8924	0.9079	0.9001
De-CNN	0.6288	0.4306	0.6532	0.5190
De-GRU	0.6890	0.5237	0.7568	0.6190
De-HANCD	0.9350	**0.9015**	**0.9156**	**0.9085**

Results are presented in Tables 5-6. It is seen that the proposed debiasing strategy can both alleviate the bias and retain high prediction accuracy. For instance, for *Instagram*, the highest AUC and F1 score of all evaluated models are achieved by the approach. For *Vine*, the improvement of De-HAN over HAN is 9.8% and 41.9% for AUC and F1 score, respectively. The improvement over Constrained is 15.8% and 15.4%, respectively. Biased models present much lower Precision than Recall for *Vine*. This result is in line with the findings in Table 4, where it is observed that the larger bias component is associated with FPRs in *Vine*. This indicates that when the sample size is small, these models overfit sensitive triggers for detecting bullying instances. The debiasing strategy effectively reduces models' reliance on those terms and utilizes contextual information for prediction.

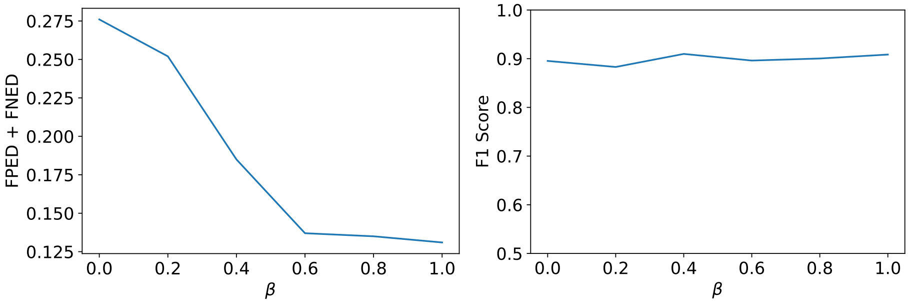


Figure 12. Total Bias (Left Figure) and F1 Score (Right Figure) of *De-HANCD* Using Different Values of β

5.4.5 What is the source of gain?

What is the ingredient that enables my approach to achieve both the lowest bias and highest accuracy? This non-compromising approach may be attributed to the proposed RL framework that effectively captures contextual information. In this section, the impact of parameter in 7.1 is examined by varying $\beta \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Performance w.r.t. bias mitigation (total bias) and cyberbullying detection (F1 score) is shown in 12.

The results clearly show the efficacy of the proposed RL framework for bias mitigation. In particular, as is increased, more effort toward alleviating biases by minimizing both FPED and FNED simultaneously is put by the RL agent. Moreover, by interacting with the environment, the RL agent also leverages contextual information in order to minimize the prediction error and receive a larger reward. As a result, the RL agent largely reduces biases while improving the prediction accuracy, as shown by the slight increase in detection performance of the classifier in Figure 12.

5.5 Conclusion

In this work, unintended biases in datasets for session-based cyberbullying detection were examined. In contrast to conventional data for bias mitigation in text classification, social media sessions consist of a sequence of comments with rich contextual information. To alleviate these unintended biases, an effective debiasing strategy by leveraging techniques in RL is proposed. The approach is context-aware and model-agnostic, and does not require additional resources or annotations aside from a pre-defined set of potentially sensitive triggers related to cyberbullying. Empirical evaluations demonstrated that the approach can mitigate unintended bias in the data without impairing a model's prediction accuracy.

Chapter 6

ROBUSTNESS: GENERATING TOPIC-PRESERVING SYNTHETIC NEWS

6.1 Background

Text generation is an important task for Natural Language Processing (NLP). With the rise of deep neural networks such as Recurrent Neural Networks (RNNs) and Long Shot Term Memory (LSTM) cells [38], there has been significant performance improvement in language modeling and text generation. Text generation has many different applications such as paraphrase generation and data augmentation. One important application of text generation in NLP is synthetic news content generation [143].

Recently, internet has proliferated a plethora of disinformation and fake news [2, 120]. Moreover, recent advancements in language models such as GPT-2 [100] allow one to generate synthetic news based on limited information. For example, models like Generative Adversarial Network (GAN) [41] can generate long readable text from noise, and GPT-2 [100] can write news stories and fiction stories given simple contexts such as part of a sentence or a topic. In the context of news generation, Grover is a causal language model that can generate fake news using different variables such as *domain*, *date*, *authors*, and *headline* [143]. While Grover is shown effective, it requires many conditional variables to generate relevant news. To study machine-generated news, I propose a model to generate realistic synthetic news. Throughout this paper, I refer to realistic news as news similar to human-written words. The crucial task of synthetic news generation enables us to (1) automatically generate news and (2) use synthetic news to study the differences between human-generated and machine-generated news.

For example, one major problem in fake news detection is the challenge to differentiate between human and machine-generated text [143].

With the advances in language models (e.g., GPT-2 and GPT-3), miscreants can leverage them to spread fake news through social media. To tackle this problem, as the first step, I need ample synthetic news with which researchers can study the nuances between human- and machine-generated text to detect disinformation on social media.

Existing methods may fall short when generating realistic news controlled by a specific context. For instance, fake news usually has a catchy style and should stay on topic to make its audience believe it, as in the example of “A shocking news report claims Kourtney Kardashian’s pregnant again”. The shortcomings in existing language models and the lack of a proper machine-generated news dataset underscore the importance of topic-preserving and stylized synthetic news generation. Moreover, fine-tuning language models do not help us in this matter as it is non-trivial to enforce topic preservation on a language model directly. In essence, I address the challenge of generating topic-preserving realistic synthetic news.

Our solutions to these challenges result in a novel framework RLTG (Reinforcement Learning-based Text Generator), for generating topic-preserving realistic news. The proposed framework RLTG consists of three major components: (1) a language model component to generate a probability distribution over a vocabulary for the next word, given a text input; (2) a Reinforcement Learning (RL) component capable of leveraging the language model to control news generation; and (3) a fake news detection module as an adversary to help the RL agent generate realistic fake news contents.

6.2 Problem Statement

Our goal is to generate synthetic news content $\mathbf{S_T}$ given topic $\mathbf{S_0}$. The generated news content $\mathbf{S_T}$ should be related to the given topic $\mathbf{S_0}$ and it should have a similar style to real news. A piece of news content has a similar style to real news if it cannot be detected as fake news using a classifier. Here, I study the following problem:

Let $\mathcal{X} = \{(\mathbf{S_0^1}, \mathbf{x_1}), (\mathbf{S_0^2}, \mathbf{x_2}), ..., (\mathbf{S_0^N}, \mathbf{x_N})\}$ denote a set of N news with topic $\mathbf{S_0}$ and content $\mathbf{x}$. Both topic $\mathbf{S_0} = \{w_0^s, w_1^s, ..., w_k^s\}$ and news content $\mathbf{x} = \{w_0, w_1, ..., w_l\}$ consist of words w. I consider topic as the news title or the first few words of a news content. In general, S_t shows the generated text at time t. Given a set of news dataset $\mathcal{X}$, learn a reinforcement learning agent F that can generate news content $\mathbf{S_T}$ based on a given topic $\mathbf{S_0}$ such that: (1) $\mathbf{S_T}$ is related to the given topic $\mathbf{S_0}$; and (2) $\mathbf{S_T}$ has a similar style to real news.

6.3 Proposed Method

In this section, I discuss the adversarial reinforcement learning-based synthetic news content generator. The input of this model is a topic $\mathbf{S_0} = \{w_1^s, w_2^s, ..., w_k^s\}$. My model, then, generates a new sequence $\mathbf{S_T} = \{w_1, ..., w_k, w_{k+1}, ..., w_T\}$ which is the generated news content. Our model consists of several components: (1) a language model component that is in charge of generating a probability distribution over vocabulary words ; (2) an RL component that will select a word based on the language model's output; and (3) an adversarial component that will help the RL agent choose proper words from the language model's output. First, I go through news content generation using adversarial RL, and then I discuss using an adversary to generate realistic fake news.

6.3.1 Topic-preserving News Generation

Existing language models are proposed to generate general or domain-specific texts [31, 100]. Although I can fine-tune these models (e.g., fine-tuning GPT-2) according to my need using a related dataset, I do not have control over its output because I cannot enforce topic-preservation or realistic synthetic news generation on the model. Following the success of Reinforcement Learning (RL) [117], I propose an adversarial RL method to control the generated output of a language model. In recent studies, RL has been used to update a model's parameters [64, 67]. In this work I explore a new direction by using RL as a standalone component to leverage the language model's output to generate text. The main advantage of using RL alongside GPT-2 is that we can use non-differential metrics in the reward function to generate a coherent text. Moreover, it enables us to have more control on the output of the language model by leveraging adversaries or changing the reward function.

In adversarial RL an agent keeps interacting with a defined environment to learn an optimized action selection policy $\pi(s)$ for each state. An RL agent is trained to choose the next word w for current generated news $\mathbf{S_t}$ according to a reward function and an adversary.

Figure 13 shows the high-level structure of RLTG. In this model, the adversarial reinforcement learning agent gets a state s_t as input, then returns an action a_t which indicates an index to one of the top words from the language model L's output. Each interaction between the agent and the environment creates an experience tuple

$(s_t, a_t, s_{t+1}, r_{t+1})$, meaning that the agent chose action a_t given the state s_t. After action a_t, the state will change to s_{t+1} and the environment returns reward r_{t+1}. This tuple is then used to train the agent. An RL model relies on four main parts: environment, state, action, and reward function.

- ***Environment*** is where the RL agent interacts to learn the best action for each state. In my problem, the environment includes a language model L, an adversary ADV, and a state creator component M. The language model L takes an input text and returns a probability distribution over vocabulary $P \in R^{1\times|V|}$ and hidden states $H \in R^{1\times e}$, where e indicates the embedding size. The adversary ADV gets an input text and returns a score for the reward function. Finally, the state creator M gets the outputs of the language model as input and returns a vector $\mathbf{s} \in !R^{1\times|s|}$ ($|s|$ shows state size) which acts as the input state s for the agent.

- ***State*** shows the agent's current situation. The agent uses the state to determine a subsequent optimal action. The state is the output of the state creator component M. Because my goal is to select the best next word for the current generated news $\mathbf{S_t}$ at time t, the state should contain information about both the context of the current generated news $\mathbf{S_t}$, and information about the next word choices. To this end, I design two separate neural networks AE^1 and AE^2 to encode this information. AE^1 is used to create the context vector c_g using hidden state $\mathbf{H}$ from the language model L's output, while the AE^2 is used to create a context vector c_w given previous top K words of the language model L's output. For both cases, I train and use autoencoders [5]. An autoencoder is an unsupervised neural network that learns to compress and encode data and then to reconstruct the input using the encoded data. It has two components, an encoder and a decoder. The encoder takes an input and returns a vector $\mathbf{v}$ which is interpreted as the context vector, containing important information about the input.

The decoder is the reverse of the encoder: given the encoded vector $\mathbf{v}$, it tries to reconstruct the original input to the network. After training an autoencoder, I can use the context vector v containing important information about the input [5].

In my method, the first autoencoder AE^1, gets the hidden state $\mathbf{H}$ as input and returns the reconstructed hidden state $\mathbf{H'}$. This autoencoder uses Multi-Layer Perceptron (MLP) networks as both encoder and decoder. The purpose of this autoencoder is to reduce the dimension of the hidden state $\mathbf{H}$. After training this autoencoder on a set of hidden states $\mathbf{H}$, I get the output of the encoder as context vector $\mathbf{c_g}$.

The second autoencoder AE^2, inspired by [27], uses Convolutional Neural Network (CNN) as both encoder and decoder. To this end, each word from top K words is passed through an embedding layer to convert it to a vector $\mathbf{w} \in R^{1 \times e}$. The embedded words $\mathbf{w}$ are then concatenated to form a matrix m with size of $(K \times e)$. After training this autoencoder using different top K words, I consider the output of the encoder as the context vector $\mathbf{c_w}$. Having both context vectors c_g and c_w, I then concatenate both context vectors $\mathbf{s} = Concat(\mathbf{c_g}, \mathbf{c_w})$ to create the state for the RL environment.

• ***Actions*** indicate the agent's response to a given state s. As the agent's goal is to select words, the action set A can be equal to choosing a word from the vocabulary set $\mathcal{V}$. By choosing $\mathcal{V}$ as the agent's action set, I encounter two problems: First, it takes a long time to train an agent on a large action set as the agent should try every action to find the best action a for each given state s [34]. Secondly, by having a large action set A, the agent may not be able to see every state-action set (s, a) in a limited time, and, it may result in underfitting [34]. To solve these problems, I make use of the language model L's output. One of the outputs of the language model is the probability distribution over vocabulary $\mathcal{V}$. The probability distribution indicates

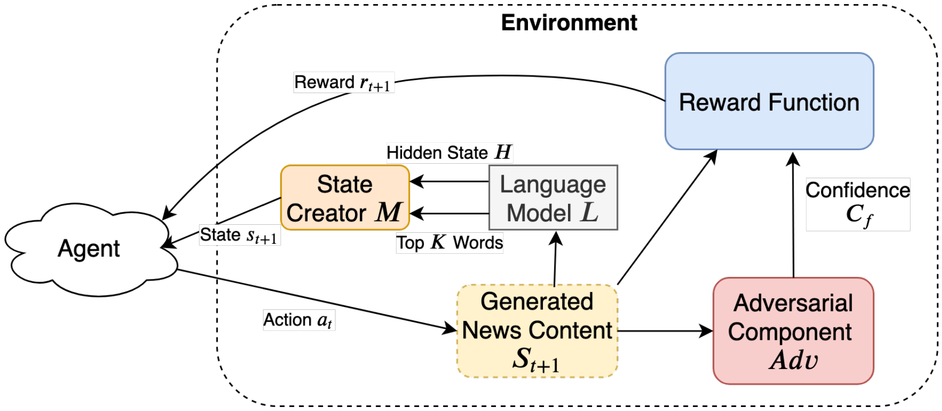


Figure 13. Architecture of Reinforcement Learning-based Text Generator

what are the best options to sample the next word for a given text $\mathbf{S_t}$. Here, I select top K words of the probability distribution as the action set, leading to a small action set.

- ***Reward Function*** evaluates agent's actions for each given set (s_t, a_t). During training, the agent uses the reward function to learn the best strategy for selecting actions. In this paper, the goal is to generate synthetic news content that is related to a given topic. To this end, I use cosine similarity to measure the similarity between the given embedded topic $\mathbf{S_0}$ and the current generated synthetic news $\mathbf{S_t}$. The reason behind using the embedded topic and generated synthetic news at time t, is that using the exact words in the Cosine similarity function may result in an agent that chooses a topic word to maximize this similarity:

$$CosineSim(\mathbf{S_0'}, \mathbf{S_t'}) = \frac{\mathbf{S_0'} \cdot \mathbf{S_t'}}{||\mathbf{S_0'}|| \cdot ||\mathbf{S_t'}||} \tag{6.1}$$

where S' is the embedded topic/news using the language model L. I use the language model L's hidden state H as the embedding for an input text as it shows the context of an input text [13].

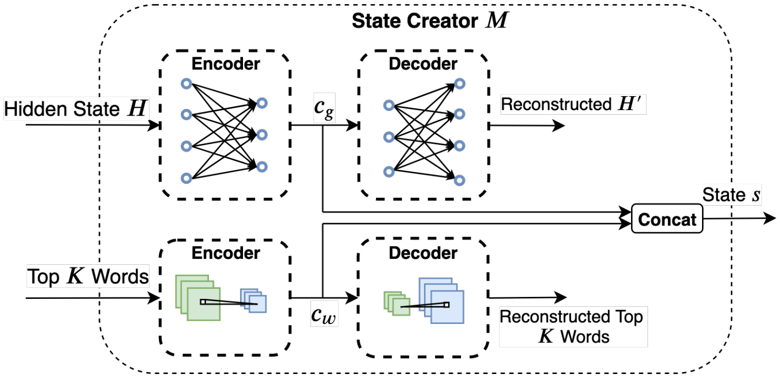


Figure 14. Architecture of State Creator

Furthermore, for generating news content, the model should consider the writing style of news content. I define style as having a similar word sequence as the referenced news. To this end, for a given synthetically generated news $\mathbf{S_t}$, I calculate the BLEU score [91] between $\mathbf{S_t}$ and news contents $\mathcal{X}$ to maintain news style. The BLEU score simply measures how many words overlap between the generated news $\mathbf{S_t}$ and the reference news contents $\mathcal{X}$. As the BLEU metric gives higher scores to similar sequential words, it can be used as a fluency metric in the designed reward function. The reward function is defined as follows:

$$r_t = \alpha CosineSim(\mathbf{S'^{i}_0}, \mathbf{S'_t}) + \beta BLEU(\mathbf{S_t}, \mathcal{X}) \tag{6.2}$$

where α controls the contribution of Cosine similarity term, and β controls the contribution of BLEU score.

6.3.2 Using Adversaries to Generate Realistic Synthetic News

Up to this point, I have considered the style and topic-preservation. To ensure that the generated news has a similar writing to real news, I use a fake news detection component as an adversary to determine whether the generated news is considered fake or true. Thus, I add an additional term to the reward function:

$$\begin{aligned} r_t =& \alpha CosineSim(\mathbf{S_0^{'}}, \mathbf{S_t'}) + \\ &+ \beta BLEU(\mathbf{S_t}, \mathcal{X}) + \lambda(1 - C_f(\mathbf{S_t})) \end{aligned} \tag{6.3}$$

Where $C_f \in [0, 1]$ is the confidence of the fake news classifier given an input, and λ shows the importance of this term. The confidence shows the probability of a news content being fake.

For training the agent, I use news dataset $\mathcal{X} = \{(\mathbf{S_0^1}, \mathbf{x_1}), (\mathbf{S_0^2}, \mathbf{x_2})\ , ..., (\mathbf{S_0^N}, \mathbf{x_N})\}$ in which $\mathbf{S_0^i}$ shows the topic of i^{th} news and $\mathbf{x_i}$ shows the content of that news. During training, the agent chooses an action a_t leading to selecting word $w_t \in \mathcal{V}$, which is then added to current generated news $\mathbf{S_t} = \{w_1, w_2, ..., w_k, ..., w_t\}$ to generate $\mathbf{S_{t+1}} = \{w_1, ..., w_k, ..., w_t, w_{t+1}\}$. The modified text $\mathbf{S_{t+1}}$ is then passed to the adversary C_f and the reward function to calculate the reward value r_{t+1} considering news content $\mathbf{x}$. Furthermore, the modified text $\mathbf{S_{t+1}}$ is passed to the language model L. Using the outputs of the language model L the environment generates next state s_{t+1}. In the following I discuss the details of using adversarial reinforcement learning.

In adversarial reinforcement learning, the goal is to learn an action policy $\pi(s)$ which leads to maximum amount of accumulated reward $R = \sum_{t=0}^{t=T} r_t$ where T is the terminal time. To find the best action selection policy $\pi(s)$, I use experiences in form of $(s_t, a_t, s_{t+1}, r_{t+1})$ to train the agent. There are different algorithms to train an agent. Policy gradient and Q-Learning are two popular algorithms for training an agent [130]. I use Deep Q-Learning which is an advanced variant of Q-Learning.

In Deep Q-Learning (DQL), the agent uses a neural network as a function approximator to find an action regarding a given state s. The input of this neural network is state s and the outputs are the values for $(s, a_i)_{i=0}^{|A|}$ where $|A|$ is the number of actions.

In DQL, the goal is to learn the following function:

$$Q^*(s_t, a_t) = E_{s_{t+1}}[r_{t+1} + \gamma max_{a'} Q^*(s_{t+1}, a')] \tag{6.4}$$

where Q-function $Q(s, a)$ returns the expected accumulated reward R if the agent selects action a in response to state s and $Q^*(s, a)$ denotes the optimal Q-function which returns the maximum possible accumulated reward R using the optimal policy $\pi(s)$. In this formula, the future rewards are discounted using the γ parameter. I adjust γ with respect to the importance of future rewards.

In practice, it is not feasible to estimate $Q^*(s, a)$ in Equation 6.4. To overcome this problem, I use a function approximator to estimate the Q-function $Q^*(s, a) \cong Q(s, a; \theta)$. As neural networks are excellent function approximators [28], DQL leverages a neural network with parameters θ called Deep Q-Network (DQN) to find the Q-function $Q(s, a; \theta)$ by minimizing the following loss function:

$$L(\theta) = E_{s_t, a_t, s_{t+1}, r_{t+1}}[(y - Q(s, a; \theta))^2] \tag{6.5}$$

where y is the target Q-value calculated using Equation 6.6:

$$y = E_{s_{t+1}}[r_{t+1} + \gamma Q(s_{t+1}, a'; \theta')] \tag{6.6}$$

where θ' is the DQN's parameters from the previous iteration.

Finally, I update the DQN parameters using the derivation of Equation 6.5 with respect to θ:

$$\begin{aligned} \nabla_\theta L(\theta) =& E_{s_t, a_t, s_{t+1}, r_{t+1}}[(r + \gamma max_{a'} Q(s_{t+1}, a'; \theta') - \\ & - Q(s_t, a_t; \theta)) \nabla_\theta Q(s_t, a_t; \theta)] \end{aligned} \tag{6.7}$$

Algorithm 2 Learning Process of Reinforcement Learning-based Text Generator

```
Require: $L$, $\epsilon$, $T$, $M$.
Initialize replay memory $R$, environment, policy, and target networks
while training is not terminal do
    $H, topK \leftarrow L(topic)$
    $s_t \leftarrow M(H, topK)$
    for $t \in \{0, 1, ..., T\}$ do
        Choose action $a_t$ using $\epsilon$-greedy
        Perform $a_t$ on $s_t$ and get $(s_{t+1}, r_{t+1})$
        $R \leftarrow R + (s_t, a_t, r_{t+1}, s_{t+1})$
        $s_t \leftarrow s_{t+1}$
        for $(s, a, s', r) \in$ sampled mini-batch $b$ from $R$ do
            Update DQN weights using Eq. 6.7 w.r.t. policy and target networks
        end for
        if exchange condition met then
            Exchange weights between policy and target network
        end if
    end for
end while
```

We have specifically used DQL with memory replay and two networks as target and policy, respectively. The memory replay helps the agent to remember past experiences. The training algorithm is presented in Algorithm 2[5].

6.4 Experiments

In this section, I conduct experiments to evaluate the performance of my method. In these experiments, I try to answer the following questions: **Q1:** How well my method can generate synthetic news in comparison to existing methods in terms of topic similarity? **Q2:** How fluent is the generated synthetic news using RLTG? and **Q3:** How well humans evaluate the RLTG's generated synthetic news?

To answer the first question Q1, I consider Cosine similarity; for Q2, I use ROUGE-L metric; and for Q3, I perform human evaluation using a survey to assess RLTG's

[5] The source code will become publicly available upon acceptance.

generated synthetic news in terms of content, title, overall readability, and being realistic or not.

6.4.1 Data

We utilize FakeNewsNet dataset [121] to fine-tune GPT-2 and train my model. This dataset consists of news data $\mathcal{X}$ from two different platforms *GossipCop* and *Politifact*. GossipCop is a fact-checking website, which reports on celebrity news. Politifact is a similar platform, which checks the truth of political news and reports. In this dataset, news are classified into *real* or *fake*. *Politifact* contains $2,645$ true and $2,770$ fake news, while the *GossipCop* includes $3,586$ true and $2,230$ fake news respectively. Dataset statistics are shown in Table 7. We consider the first few words of each news x_i content as topic $\mathbf{S_0}$.

Table 7. Statistics of FakeNewsNet Dataset

Platform	**G**	**P**
# True news	3,586	2,645
# Fake news	2,230	2,770
# Total News	5,816	5,415

6.4.2 Implementation Details

In this part I go through the parameters and implementation details of RLTG. In my model, I use a fine-tuned GPT-2 language model as L. To fine-tune the GPT-2 language model, I first load a pre-trained "GPT-2 medium", then I use FakeNewsNet dataset for 5 iterations to fine-tune the language model. Note that this language model has 12 hidden layers. Each hidden layer returns a tensor with size of (batch size, sequence length, hidden size), where the hidden size in "GPT-2 medium" is 768.

As it is mentioned in the proposed method, the RL agent has a neural network which acts as a function approximator. This network gets a state as input and returns the Q-value for each $(s, a_i)_{i=1}^{K}$ set. This network has 3 layers. The first hidden layer has 1024 nodes, the second and third layer has 512 and 256 nodes, respectively. The output size of this network is equal to the number of actions. Here, the number of actions is 50, meaning that the agent chooses between the top 50 words of GPT-2's output probability. The reason I chose 50 is that among values $\{10, 25, 50, 75\}$, it showed a better reward performance than others, with $K = 75$ having a similar performance. Moreover, the the output size of the DQN network is equal to the size of state $\mathbf{s}$. To construct state $\mathbf{s}$, as in Figure 14, I have trained 2 autoencoders and concatenate the output of each encoder to create the state. The first autoencoder is considered for extracting the context of generated news using hidden state H. This autoencoder uses Multi-Layer Perceptron (MLP) to encode and reconstruct the hidden state H. The output of encoder part has 256 nodes. The second autoencoder uses Convolutional Neural Networks (CNN) to extract information about best words positions. The encoder of this autoencoder has an output layer with size of 128. The final size of state $\mathbf{s}$ is 384.

The RL agent is trained on randomly selected topics for 50000 episodes. The agent can choose between top $K = 50$ words from the language model L's output. Each episode has a terminal time of $T = 50$. As the final generated news is more important than the early generated news, a high discount factor $\gamma = 0.9$ is used. As it is mentioned in the proposed method section, I use Deep Q-Learning to train my RL agent. In this algorithm I construct a memory with size 10000 to save the experiences $(s_t, a_t, s_{t+1}, r_{t+1})$. Each experience means that the RL agent chose action a_t in state s_t. The selected action a_t resulted in transition to a new state s_{t+1} and the environment returned a reward r_{t+1}. I then use the memory array to update my model using Equation 6.5. The batch size for sampling experiences from memory is 32. During the training process I use ϵ-greedy to choose action a_t. This algorithm considers a random action with probability of ϵ and chooses the best action based on Q-values with a probability of $1 - \epsilon$. I use the following decay function to lower the value of ϵ. This function lower the ϵ according to the number of past iterations and exponentially decreases it by a constant rate $\epsilon = \epsilon_{min} + (\epsilon_{max} - \epsilon_{min})e^{\frac{-steps}{decay\_rate}}$ where *steps* is the number of past iterations and *decay_rate* controls how fast the ϵ should decrease. I use $\epsilon_{max} = 0.98$, $\epsilon_{min} = 0.02$ and the decay rate equal to $5,000$.As for the reward function parameters, I set $\alpha = \beta = \lambda = 0.5$. In this case $r \in [0, 1.5]$.

As illustrated in Figure 13, I use a fake news classifier as an adversary to calculate the value of reward function. The architecture of the fake news classifier is shown in Figure 15. The hidden size of bi-directional GRU is 128, resulting in a context vector of 256. The neural network classifier has an input size of 256, hidden size of 128, and output size of 1. I train this classifier before training the agent using Binary Cross Entropy (BCE) loss function. As DQL has a variance during training, I train my model 5 times independently, then I select the agent with the highest average rewards.

To train the RLTG model, I used a publicly available dataset, FakeNewsNet [121] that can be accessed through https://github.com/KaiDMML/FakeNewsNet. To compare RLTG to GPT-2, I have used the Hugging Face package (https://huggingface.co) to use and fine-tuned the GPT-2 model. The GPT-2 model is fine-tuned for 2 iterations using the FakeNewsNet dataset. Moreover, to run the experiments on

Grover [143], I used their publicly available package through https://rowanzellers.com/grover.

6.4.3 Experimental Design

Different baselines are used for comparison. As the proposed model is based on the OpenAI's GPT-2 language model, this language model alone is used as a baseline to determine how using an RL agent on this model can improve its results. Furthermore, the RL agent alone is also included as a baseline to generate synthetic news. The main goal, generating synthetic news content, is close to the Grover [143], so this work has been selected as a baseline. Finally, the SeqGAN method is selected because it incorporates GAN with reinforcement learning. Following is the description of the baselines:

- **GPT-2 [100]:** a language model capable of generating long text. This language model is based on transformers and has three different variations based on it's number of layers and parameters: small (117M parameters) , medium (345M parameters), and large (774M parameters). GPT-2 medium is used for fine-tuning it needs less resources.
- **Fine-tuned GPT-2 (FTGPT-2):** similar to GPT-2, but has been fine-tuned using FakeNewsNet dataset.
- **RL:** in this baseline RL technique without using a language model is used to train an agent. In this case, all components except the *actions* are the same as the proposed RLTG method. The action set in this baseline is all word in the vocabulary set V. The training process of this baseline is similar to my model.
- **Grover [143]:** a conditional language model which can generate text based on

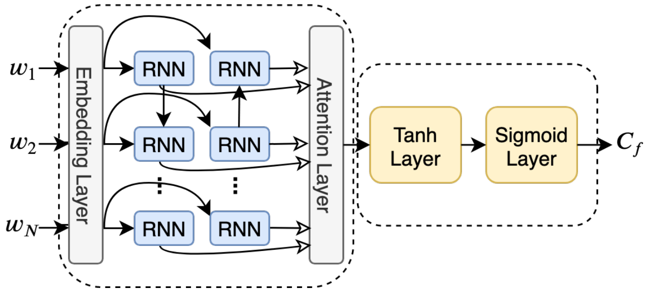


Figure 15. Architecture of Fake News Adversary Classifier

given parameters: domain, date, authors, and headline. The goal of Grover is to generate news content based on different parameters. While the results are promising, it seems this language model is very dependent on *domain* parameter, which will be explored during my evaluation.

- **SeqGAN (SeqG) [141]:** is a text generation method, which models data generator as a stochastic policy in reinforcement learning. They then use policy gradient method to train their model.
- **PPLM [29]:** is a method that leverages GPT-2 to generate domain-specific text.

6.4.4 Experimental Results

We evaluate my model's performance regarding Q1 - Q3.

Topic Similarity (Q1). To answer this question, cosine similarity as in Equation 6.1 is used to calculate the similarity between the embedding of the given topic $\mathbf{S_0}$ and the generated news $\mathbf{S_T}$. The RL agent is not wanted to exactly select topic words to maximize its reward, so text embeddings are used to calculate the similarity. In this case, words are chosen by the agent to maximize the context similarity between both the topic and the generated text." For a fair comparison, a fixed sentence length of 200 for text generation is used.

Table 8 shows the performance of RLTG against other baselines. RLTG can outperform other baselines because it considers topic similarity during the training process. Although fine-tuned GPT-2 falls behind RLTG and Grover, it has achieved a high similarity comparing to other methods. This is due to the fact that the fine-tuned GPT-2 tends to repeat itself. Note that the performance of the RL baseline is behind all models. The reason behind it is that the action set in this case is very large and the agent cannot converge easily. Furthermore, using a language model to narrow down the possible actions can have a huge impact on training the model. Finally, comparing the performance of RLTG and baselines to PPLM shows that PPLM lacks fluency. I suspect this is due to the used classifier and the fact that the GPT-2 used in this model cannot generate text in the news domain.

Fluency Test (Q2). To answer this question, both perplexity and ROUGE-L metrics are used. Perplexity may not be suitable for showing the effectiveness of a model in open-domain text generation [68], but in this case, the focus is on domain-specific news generation. Table 8 shows the results of the fluency test. Lower perplexity means the generated news is more concentrated, and it is less variant. Furthermore, the ROUGE-L score applies Longest Common Subsequence between the news contents X and generated news content ST to calculate the final score. The the ROUGE-L metric is selected for evaluation because the method is trained to achieve a high BLEU score, and using the BLEU score for evaluation is not fair. ROUGE-L measures how many words from the reference sentences have appeared in the generated news. ROUGE-L gives higher scores to sequential words and can be used as a fluency metric. The FakeNewsNet dataset is used as the reference sentences. A higher ROUGE-L score means the generated news is more fluent.

Table 8. Topic Similarity, Perplexity, and Rouge-L Score for Model's Generated News

	RLTG	GPT-2	FTGPT-2	Grover	SeqG	PPLM	RL
Similarity (↑ better)	**0.342**	0.176	0.241	0.313	0.301	0.254	0.153
Perplexity (↓ better)	**14.8**	22.3	19.8	15.3	17.4	21.8	30.4
ROUGE-L (↑ better)	**28.4%**	23.1%	24.6%	27.3%	21.5%	18.6%	17.2%

Human Evaluation (Q3). To further investigate the quality of the generated text, a human study is conducted to evaluate the generated news without knowing the origin of the news (human or machine generated). In this human study, participants are asked to give a score from 1 to 3 about topic similarity, writing style, content quality, and overall evaluation of the given text. Table 9 shows the designed questionnaire for evaluating the performance of the language models. A similar measure as [143] is used and a new question regarding the topic similarity is included.

For comparison, best performing models, RLTG and Grover are considered. 75 articles (25 human generated, 25 RLTG generated, and 25 Grover generated) are included. The results are provided in Figure 16.

6.4.5 Parameter Analysis

In this part, I study the effects of different parameters of my model on the quality of the generated news. These parameters include the reward function's parameters, α, β, and γ. The α parameter indicates the importance of topic similarity, β shows the

Table 9. Human Evaluation Questionnaire

#	Measure	Question
1	Realistic	Is the style of this article consistent? (3). Yes, this sounds like an article I would find at an online news source. (2). Sort of, but there are certain sentences that are awkward or strange. (1). No, it reads like it's written by a madman.
2	Content	Does the content of this article make sense? (3). Yes, this article reads coherently. (2). Sort of, but I don't understand what the author means in certain places. (1). No, I have no (or almost no) idea what the author is trying to say.
3	Title	Does the article sound like it's around a topic? (3). Yes, I feel that this article is talking about a single topic. (2). Sort of, I'm not sure what the article is about. (1). No, it seems this article is gibberish.
4	Overall	Does the article read like it comes from a trustworthy source? (3). Yes, I feel that this article could come from a news source I would trust. (2). Sort of, but something seems a bit fishy. (1). No, this seems like it comes from an unreliable source.

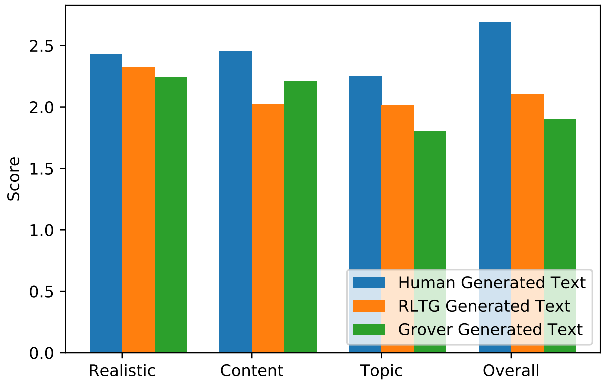


Figure 16. Human Evaluation Results

importance of readability according to the BLEU score, and γ shows the importance of the adversary which is the fake news classifier.

Furthermore, in Figure 17 the reverse confidence of classifier $(1 - C_f)$ of the fake news classifier for several periods of training iterations is shown. This figure shows

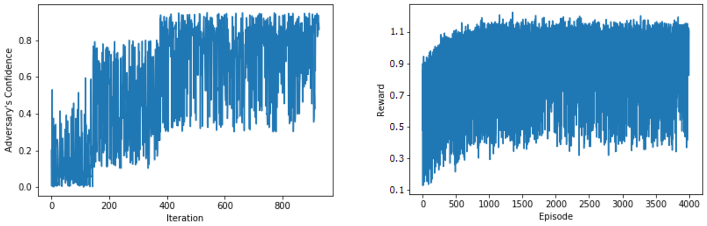


Figure 17. Reverse of Adversary's Confidence $(1 - C_f)$ on the Left, and RL Agent's Rewards on the Right

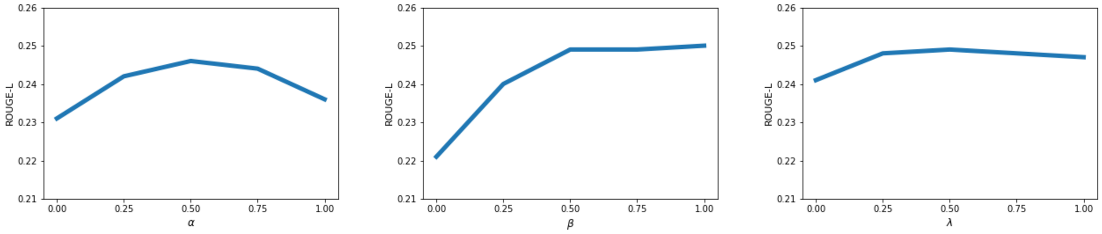


Figure 18. Impact of Different Reward Function Parameters (α, β, and λ from Left to Right) on RLTG's Rouge-L Score

the confidence for the final generated news at terminal time T. From this figure, it is concluded that the agent can generate realistic fake news.

First, the convergence of the training algorithm is assessed by showing the RL algorithm's reward values during the training. Figure 17 shows the mean reward for each episode over for each iteration. The results indicates that the average reward of agent is increasing over time, meaning that the agent is learning a policy $\pi(s)$ which can result in larger reward values. At first the rewards are low which is as a result of randomness during early episodes, but it increases as the agent learns better actions for each state s. Note that Figure 17 only shows the reward values for the first 4000 iterations to show its increasing behavior.

To measure the impact of different components in RLTG, two different tests are performed. On a high level, RLTG is composed of two main components, the GPT-2 language model, and the RL agent. In the first test, the performance of each high level component, RL and GPT-2, in RLTG is studied. Table 8 indicates the performance for RL, GPT-2, and RLTG. It is noticed that the RL model performs poorly which is expected due to the GPT-2's superior architecture. In the second test, the importance of different components in the RL part of the RLTG is studied by analyzing the effects of α, β, and γ on the reward function in Equation 6.3. The effect of these parameters is illustrated by changing them as $\alpha, \beta, \lambda \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$ and calculating the ROUGE-L score.

Figure 18 shows the effect of each component on the ROUGE-L score. For each parameter, I consider other parameter values as 0.5. While α and λ have little effect on the ROUGE-L score, the β parameter has a larger impact as it considers how well the generated news content overlaps with the given domain according to dataset $\mathcal{X}$.

We further study the effect of using a fake news detection classifier as an adversary to see if the generated news is realistic enough not to be detected as fake. By studying the reward values in Figure 17, I can see that the agent can generate news content which the adversary cannot easily detect as fake. The trained adversary has an accuracy of %81.3 and AUC of %75.3. Although it seems that the γ parameter does not contribute to generating a higher-quality text, according to the confidence values presented in Figure 17, RLTG can avoid producing obvious fake synthetic news by using features that bypasses the adversary.

6.4.6 Case Studies of Generated Synthetic News

For illustration purposes, I show some examples of the synthetic news generated by RLTG and by the baselines. Table 10 shows two examples of generated news by

Table 10. Sample Generated News given a News Topic

Model	Text
RLTG	**the wedding of prince harry and meghan markle** was held with *royal family guests* attending the *ceremony*. After nearly four hours on site, they met again with *wedding guests* from a host community to gather for one particular special day. In May there would been even-clothing on all royal guests arriving as many couples wore the customised *royal* gown...
FTGPT-2	**the wedding of prince harry and meghan markle** was held on the grounds of St. George's Chapel in Windsor, England, on May 19, 1814. The bride and *groom* were Prince William and Kate Middleton, and the *groom* was Prince Charles. The *bride*...
GPT-2	**the wedding of prince harry and meghan markle** was held on the *wedding* of prince harry and meghan markle was held on the day of the *wedding* of prince harry and meghan markle was held on the day...
Grover	**the wedding of prince harry and meghan markle**. Like many of the *gala* trends I *celebrate* today, the *wedding* of prince harry and meghan markle has taken center stage at the MWC. The mom and pop couples still roping in floral designs, staging weddings, and of course, speaking on stage at the hands of this mortal pair.

different models. The topic-related words are highlighted to show how well RLTG preserves the given topic in the generated synthetic news. Finally, Table 11 illustrates the results of the human evaluation experiment for five generated synthetic news.

6.5 Conclusion

Text generation is crucial and can be used in different NLP applications. One application of text generation is news content generation. In this work, I challenge the problem of generating realistic topic-preserving news by leveraging a pre-trained language model. To this end, I propose a reinforced model RLTG to control a language model toward news content generation. This model uses Deep Q-Learning to train an agent capable of selecting words from a language model's output to generate realistic topic-preserving news. There are various future directions that can benefit from this

Table 11. Sample Generated News and Their Average Human Evaluation Scores given a News Topic

Generated news (The **bold** part is the given topic)	Mean score
meghan markle and prince harry on monday: the british royal family are not welcome. “They do not like me, but are happy that I got married. I can be proud and have some nice holidays together.”	2.33
share fans of netflix’s cult favorite scifi drama series sense8 have a very special gift of nostalgia from them that can never truly forget. a beloved franchise in some small but undeniable shape. So this, I know how fans would look after that they would not enjoy being seen. it will become apparent soon when people who enjoy being on such shows, may also like to share these experiences on facebook to have them see those great and unique stories.	2.25
jade is my first friend to ever be pregnant in her mid 30ies. I can feel good about the baby! she will make an excellent wife. I just don’t expect a kind person or family who wants the support of an anesthesia for that to occur to a man with diabetes.	2.00
imdbcom inc takes no responsibility for the content or accuracy of its claims. It has taken place today (August 21), when an article appeared at Wired, which says: “there will be one man with guns on their faces who at a certain level who can take on ISIS”. It has gone into further information on what constitutes terrorist threats (or just about all terrorist activity).	2.50
robert pattinson says he was just kidding around about being asked by anorexist about a possible relationship, “so I were like, I don’t know if he’s a guy that I want. We’re just trying something out, and I just don’t have that. So he said I don’t want that. He was like a little boy and I don’t want it.” I have no plans that are to go into this story.	2.67

work. One future direction is to study different types of biases in the generated news and create a de-biasing model that removes bias and unfairness from pre-trained text generation models. Most importantly, I can study and analyze the hidden features and differences between real and synthetic news content.

6.6 Ethics Statement

This work aims to advance research efforts in synthetic news generation, a topic that has yet to be studied extensively. Here, considering current solutions, much work remains to elucidate how to build an effective news generation model. I believe that generating synthetic news is a stepping stone that enables us to further investigate and detect machine-generated fake news.

Chapter 7

ROBUSTNESS: DOMAIN ADAPTIVE FAKE NEWS DETECTION VIA REINFORCEMENT LEARNING

7.1 Background

With people spending more time on social media platforms[6], it is not surprising that social media has evolved into the primary source of news among subscribers, in lieu of the more traditional news delivery systems such as early morning shows, newspapers, and websites affiliated with media companies. For example, it was reported that 1 in 5 U.S. adults used social media as their primary source of political news during the 2020 U.S. presidential elections [80].

The ease and speed of disseminating new information via social media, however, have created networks of disinformation that propagate as fast as any social media post. For example during the COVID pandemic era, around 800 fatalities, 5000 hospitalizations, and 60 permanent injuries were recorded as a result of false claims that household bleach was an effective panacea for the SARS-CoV-2 virus [51, 26]. Unlike traditional news delivery where trained reporters and editors fact-check information, curation of news on social media has largely been crowd-sourced, i.e., social media users themselves are the producers and consumers of information.

In general, humans have been found to fare worse than machines at prediction tasks [111, 113] such as distinguishing between fake or legitimate news. Machine

[6] In 2020, internet users spent an average of 145 minutes per day on social media [127].

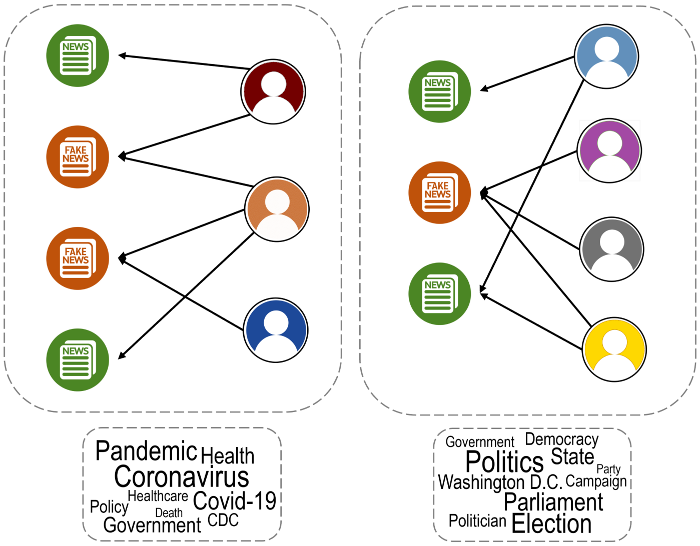


Figure 19. Comparing Two Domains, Healthcare and Politics, in Fake News Detection

learning models for automated fake news detection have even been shown to perform better than the most seasoned linguists [97]. To this end, many automated fake news detection algorithms have largely focused on improving predictive performance for a specific news domain (e.g., political news). The primary issue with these existing state-of-the-art detection algorithms is that while they perform well for the domain they were trained on (e.g., politics), they perform poorly in other domains (e.g., healthcare). The limited cross-domain effectiveness of algorithms to detect fake news are mostly due to (1) the reliance of content-based approaches on word usage that are specific to a domain, (2) the model's bias towards event-specific features, and (3) domain-specific user-news interaction patterns (Figure 19). As one of the contributions of this work, I will empirically demonstrate that the advertised performance of SOTA methods is not robust across domains.

Additionally, due to the high cost and specialized expertise required for data annotation, limited training data is available for effectively training an automated model across domains. This calls for using auxiliary information such as users' comments and motivational factors [118, 53] as value-adding pieces for fake news detection.

To address these challenges, I propose a domain-adaptive model called **RE**inforced **A**daptive **L**earning **F**ake **N**ews **D**etection (REAL-FND), which uses generalized and domain-independent features to distinguish between fake and legitimate news. The proposed model is based on previous evidence that illustrates how domain-invariant features could be used to improve the robustness and generality of fake news detection methods. As an example of a domain-invariant feature, it has been shown that fake news publishers use click-bait writing styles to attract specific audiences [147]. On the other hand, patterns extracted from the social context provide rich information for fake news classification within a domain. For example, a user's comment providing evidence in refuting a piece of news is a valuable source of auxiliary information [119]. Or, if a specific user is a tagged fake news propagator, the related user-news interaction could be leveraged as an additional source of information [53].

In REAL-FND, instead of applying the commonly-used method of adversarial learning in training the cross-domain model, I transform the learned representation from the source to the target domain by deploying a reinforcement learning (RL) component. Other RL-based methods employ the agent to modify the parameters of the model. However, I use the RL agent to modify the learned representations to ensure that domain-specific features are obscured while domain-invariant components are maintained. An RL agent provides more flexibility over adversarial training because any classifier's confidence values could be directly optimized (i.e., the objective function does not need to be differentiable).

7.2 Problem Statement

Let $\mathcal{D}_s = \{ (\mathbf{x}^s_1, y^s_1), (\mathbf{x}^s_2, y^s_2), ..., (\mathbf{x}^s_N, y^s_N)\}$ and $\mathcal{D}_t = \{ (\mathbf{x}^t_1, y^t_1), (\mathbf{x}^t_2, y^t_2), ..., (\mathbf{x}^t_M, y^t_M)\}$ denote a set of N and M news article with labels from source and target domains, respectively. Each news article $\mathbf{x}_i$ includes a content which is a sequence of K words $\{w_1, w_2, ..., w_K\}$, a set of comments $\mathbf{c}_j \in \mathcal{C}$, and user-news interactions $\mathbf{u}_j \in \mathcal{U}$. User-news interactions $\mathbf{u}_i$ is a binary vector indicating users who posted, re-posted, or liked a tweet about news $\mathbf{x}_i$. The goal of the reinforced domain adaptive agent is to learn a function that converts the news representation $\mathbf{x}_i \in \mathcal{D}_t$ from target domain to source domain. The problem is formally defined as follows:

Definition (Domain Adaptive Fake News Detection). Given news articles from two separate domains $\mathcal{D}_s$ and $\mathcal{D}_t$, corresponding users' comments $\mathcal{C}_s$ and $\mathcal{C}_t$, and user-news interactions $\mathcal{U}_s$ and $\mathcal{U}_t$ from the source S and target T domains, respectively, learn a domain-independent news article representation using $\mathcal{D}_s$ and a small portion of $\mathcal{D}_t$ that can be classified correctly by the fake news classifier F.

7.3 Proposed Method

In this section, I describe my proposed model, **RE**inforced domain **A**daptation **L**earning for **F**ake **N**ews **D**etection (REAL-FND). The input of this model is the news articles from source domain $\mathcal{D}_s$ and a portion (γ) of the target data set $\mathcal{D}_t$. As shown in Figure 20, the REAL-FND model has two main components: (1) the news article encoder, and (2) the reinforcement learning agent. In the following subsections, I explain these two components in detail.

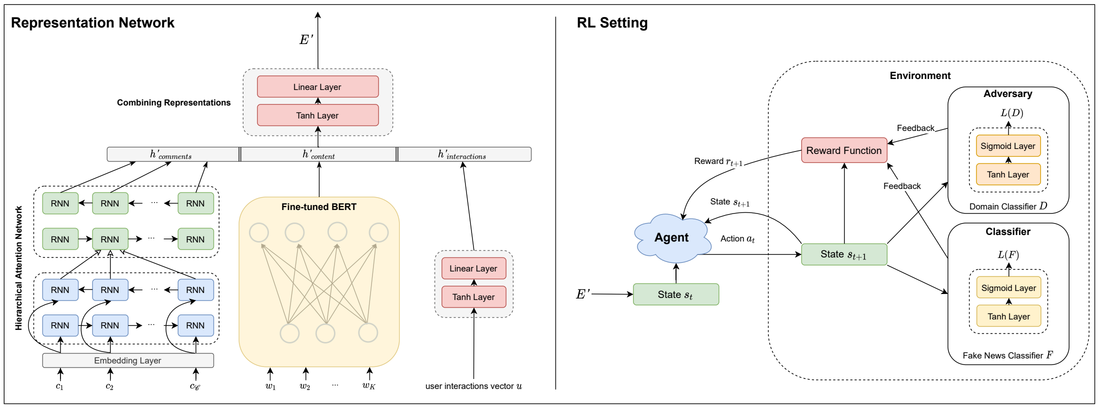


Figure 20. Architecture of Proposed RL-based Fake News Detection

7.3.1 News Article Encoder

The problem of detecting fake news requires learning a comprehensive representation that includes information about news content and its related auxiliary information. I consider news comments and user-news interactions as auxiliary information for fake news detection. Recurrent neural networks (RNN) such as Long Short Term Memory (LSTM) have been proven to be effective in modeling sequential data [120]. However, Transformers, due to their attentive nature, create a better text representation that includes vital information about the input in an efficient manner [136]. Thus, in this work, I use Bidirectional Encoder Representations from Transformers (BERT) in creating the representation vector for the news content.

BERT is a pre-trained language model that uses transformer encoders to perform various NLP-based tasks such as text generation, sentiment analysis, and natural language understanding. Previous research have used BERT to achieve state-of-the-art performance on different applications of sentiment analysis tasks [136, 140, 65]. Although BERT has been pre-trained on a large corpus of textual data, it should be fine-tuned to perform well on a specific task [140, 69]. To create a well-defined news content representation, I fine-tune BERT using news articles $\mathbf{x} = \{w_1, w_2, ..., w_K\}$ from source domain dataset $\mathcal{D}_s$ and a portion γ of the target dataset $\mathcal{D}_t$.

In addition to the news content $\mathbf{x}_i$, I also consider the article's comments and user-news interactions. In the experiments, I show that using auxiliary information leads to better detection performance as the model accounts for user's reliability and feedback on the news article. Nonetheless, previous studies also have shown that comments on news articles can improve fake news detection [119] by extracting semantic clues confirming or disapproving the authenticity of the content. Due to the fact that not all comments are useful for fake news classification, I use Hierarchical Attention Network (HAN) to encode the comments of a news article. The hierarchical structure of HAN facilitates the importance of every comment, as well as the salient word features. To create a representation for the news article comments, I pre-train HAN by stacking it with a feed-forward neural network classifier. After pre-training HAN, I remove the feed-forward classifier and only use the HAN to encode the news article comments. Note that in case a news article does not have any comments, I use vectors with zero values for comments representation.

Moreover, in addition to the news article comments, I also consider the user-news interactions to improve fake news detection. For a news article $\mathbf{x}$, the user-news interactions $\mathbf{u}$ is a binary vector where $\mathbf{u}_i$ indicates user i has tweeted, re-tweeted, or commented on a tweet about that news. Thus, the user-news interactions vector is a representation of the user behaviour toward a news article. To encode this information, I use a feed-forward neural network that takes the binary vector of user-news interaction as input and returns a representation containing important information about that interaction.

After the representation networks (i.e., HAN, BERT, and the feed-forward neural network) were constructed and BERT and HAN were pre-trained, the output of these three components was concatenated into one vector $E = (h'_{comments}||h'_{content}||h'_{interactions})$ (indicates concatenation) and passed to another feed-forward network to combine this information into a single vector $\mathbf{E}'$ [54]. Once the representation network was stacked with a feed-forward neural network classifier, a fake news classifier was trained using the source domain data $\mathcal{D}_s$ and a portion γ of the target domain data $\mathcal{D}_t$. After the fake news classifier F was trained, the weights of the representation network were frozen and a domain classifier D was trained using the representation $\mathbf{E}'$ of the news articles. By the end of this process, three sub-components were created: (1) a representation network that encodes news content, comments, and user-news interactions, (2) a single-domain fake news classifier, and (3) a domain classifier. In the following section, the second component of the model (i.e., Figure 20-RL Setting) that uses reinforcement learning to create a domain adaptive representation using $\mathbf{E}'$ is discussed.

7.3.2 Reinforced Domain Adaptation

Inspired by the success of RL [52, 135, 83], I model this problem using RL to automatically learn how to convert textual representations from both source domain data $\mathcal{D}_s$ and target domain data $\mathcal{D}_t$. Instead of applying a commonly-used approach of adversarial learning to train both F and D classifiers, an RL-based technique is utilized. RL would transform the representation to a new one such that it works well on the fake news classifier F, but not on the D classifier (i.e., the adversary). In this approach, the agent interacts with the environment by choosing an action a_t in response to a given state s_t. The performed action results in state s_{t+1} with reward

r_{t+1}. The tuple $(s_t, a_t, s_{t+1}, r_{t+1})$ is called an experience which will be used to update the parameters of the RL agent.

In my RL model, an agent is trained to change the news article representation $\mathbf{E}'$ into a new representation that deceives the domain classifier, but preserves the accuracy for the fake news classifier. The RL agent learns this transition by changing the values in the input vector $\mathbf{E}'$ and receiving feedback from both fake news classifier F and domain classifier D. To create an RL setting, I define four main parts of RL in my problem, i.e., environment, state, action, and reward.

- **Environment**: the environment in my model includes the RL agent, pre-trained fake news classifier F, and the pre-trained domain classifier D. In each turn, the RL agent performs an action on the news article representation E' (known as state s_t) by changing one of its values (performing action a_t). The modified representation is passed through classifiers F and D to get their confidence scores to calculate the reward value. Finally, the reward value and the modified representation is passed to the agent as reward r_{t+1} and state s_{t+1}, respectively.
- **State**: the state is the current news article representation $\mathbf{E}'$ that describes the current situation for the RL agent.
- **Actions**: actions are defined as selecting one of the values in news article representation $\mathbf{E}'$ and changing it by adding or subtracting a small value σ. The total number of actions are $|E'| \times 2$.
- **Reward**: since the aim is to nudge the agent into creating a domain adaptive news article representation, the reward function looks into how much the agent was successful in removing the domain-specific features from the input representation. Specifically, I define the reward function at state s_{t+1} according to the confidence of both the domain classifier and the fake news classifier. Considering

the news article embedding $\mathbf{E'}_{t+1}$ with its label i (being fake or real) and domain label j at state s_{t+1}, the reward function is defined as follows:

$$r_{t+1}(s_{t+1}) = \alpha Pr_F(l = i|\mathbf{E'}_{t+1}) - \beta Pr_D(l = j|\mathbf{E'}_{t+1}) \tag{7.1}$$

where l indicates the label.

7.3.3 Optimization Algorithm

Given the RL setting, the aim is to learn the optimal action selection strategy $\pi(s_t, a_t)$. Algorithm 3[7] shows this optimization process. At each timestep t, the RL agent changes one of the values in the news article representation, $s_t = \mathbf{E'}_t$, and gets the reward value, r_{t+1}, based on the modified representation $s_{t+1} = \mathbf{E'}_{t+1}$ and the selected action a_t. The goal of the agent is to maximize its reward according to Equation 7.1.

To train the agent, I use the REINFORCE algorithm, which uses policy gradient to update the agent [146]. Considering the agent's policy according to parameters θ as $\pi_\theta(s_t, a_t)$, the REINFORCE algorithm uses the following loss function to evaluate the agent:

$$\mathcal{L}(\theta) = \log(\pi_\theta(s_t, a_t) \cdot G_t) \tag{7.2}$$

where $G_t = \sum_{i=1}^{t} \lambda^i r_{i+1}$ is the cumulative sum of discounted reward, and λ indicates the discount rate. The gradient of the loss function is used to update the agent:

$$\nabla\theta = \mathrm{lr}\nabla_\theta \mathcal{L}(\theta) \tag{7.3}$$

[7] The source code will become publicly available upon acceptance.

Algorithm 3 Learning Process of Proposed Rl-based Fake News Detection

```
Require: News article representations $\mathbf{E}' \in \mathcal{D}$ - fake news classifier $F$ - domain
    classifier $D$ - parameters $\alpha, \beta$, and $\lambda$ - terminal time $T$.
Initialize state $s_t$ and memory $M$.
while training is not terminal do
    $s_t \leftarrow \mathbf{E}'$
    for $t \in \{0, 1, ..., T\}$ do
        Choose action $a_t$ according to current distribution $\pi(s_t)$
        Perform $a_t$ on $s_t$ and get $(s_{t+1}, r_{t+1})$
        $M \leftarrow M + (s_t, a_t, r_{t+1}, s_{t+1})$
        $s_t \leftarrow s_{t+1}$
        for each timestep $t$, reward $r$ in $M_t$ do
            $G_t \leftarrow \sum_{i=1}^{t} \lambda^i r_{i+1}$
        end for
        Calculate policy loss according to Equation 7.2
        Update the agent's policy according to Equation 7.3
    end for
end while
```

where lr indicates the learning rate.

7.4 Experiments

In the designed experiments, an attempt is made to understand how the differences between domains affect the performance of fake news detection models. Moreover, an investigation is conducted into how the proposed RL-based approach will overcome performance degradation due to the differences in news domains. The main evaluation questions are as follows:

- **Q1.** How well does the current fake news detection methods perform on social media data?
- **Q2.** How well does the proposed model detect fake news on a *target domain* $\mathcal{D}_t$ after training the model on a *source domain* $\mathcal{D}_s$?

- **Q3.** How do auxiliary information contribute to the improvement of the fake news detection performance?

Q1 evaluates the quality of fake news detection models. This question is answered by training and testing fake news detection models on the same domain and comparing their performance. **Q2** studies the effect of domain differences on the performance of fake news detection models. A similar approach to **Q1** is used to answer this question by training on one domain, but testing on another. In **Q3** we perform ablation studies to analyze the impact of auxiliary information, the reward function parameters α and β, and the portion of target domain data γ needed to achieve an acceptable detection performance.

7.4.1 Datasets

We use the well-known fake news data repository FakeNewsNet [122], which contains news articles along with their auxiliary information such as users' metadata and news comments. These news articles have been fact-checked with two popular fact-checking platforms - *Politifact* and *GossipCop*. Politifact fact-checks news related to the U.S. political system, while GossipCop fact-checks news from the entertainment industry. In addition to the existing Politifact news articles from FakeNewsNet, the dataset is enriched by adding $5,000$ annotated Politifact news from the dataset introduced by Rashkin et al. [103]. Table 12 shows the statistics of the final dataset. The Politifact news articles from [103] include truth ratings from 0 to 5, in which only news with label $\in \{0, 4, 5\}$ are considered.

Table 12. Statistics of Politifact and GossipCop Datasets

	Politifact	**GossipCop**
# True News	2,645	3,586
# Fake News	2,770	2,230
# News	5,415	5,819
# News with Comments	415	5,819
# Users	60,053	43,918
# Unique Users	100,520	

7.4.2 Data Pre-processing

Each news article from the final dataset contains news content, users' comments, and their meta-data. The textual data (i.e., news content and users' comments) is pre-processed by removing the punctuation, mentions, and out-of-vocabulary words. Further, as BERT has a limitation of getting 512 words as input, the news content and every comment are truncated to include their first 512 words. Finally, the users' meta-data is utilized to create a user-news interaction matrix by tracking every user's interactions with news articles.

7.4.3 Implementation Details

The training process has been conducted in three stages: (1) pre-training the representation networks, (2) training the fake news and domain classifiers, and (3) training the RL agent. The following expands the implementation details of each stage:

Pre-training: In this stage, BERT and HAN networks are fine-tuned to generate a reasonable text representation from news content and users' comments, respectively. Motivated by the low memory consumption of the distilled version of Bert [107], the base model of Distilled BERT is used for creating the textual representation of the news contents. The model is fine-tuned for 3 iterations using a classifier on top of it. Moreover, to create the representations related to the user's comments, HAN is pre-trained by stacking it with a simple fake news classifier and fine-tuning it for 5 iterations. After both models are fine-tuned, the classifier module in both distilled Bert and HAN is removed. These pre-trained networks are placed in the final architecture of the model.

Training classifiers F and D: With passing the news content through the representation network in Figure 20, a fake news classifier and a domain classifier are trained using the news content representations. During the training, the weights of BERT and HAN networks are not updated. In this stage, a dropout with $p = 0.2$ was used for both fake news and domain classifiers. Both classifiers use a similar feed-forward neural network with a single hidden layer of 256 neurons. The networks are trained using Adam optimizer with a learning rate of $1e-5$ and a Cross Entropy loss function:

$$\mathcal{L}_{CE} = -\frac{1}{M}\sum_{i=1}^{M}(y_i \log(p_i) + (1-y_i)\log(1-p_i)) \tag{7.4}$$

Table 13. Performance of Proposed Model and Baselines, Trained Only on One Domain and Tested on Both Politifact and Gossipcop Domains

Model	**GossipCop → Politifact**		**GossipCop → GossipCop**		**Politifact → Politifact**		**Politifact → GossipCop**	
	F1	**AUC**	**F1**	**AUC**	**F1**	**AUC**	**F1**	**AUC**
CSI	0.532±0.11	0.563±0.14	0.811±0.05	0.832±0.08	0.756±0.02	0.785±0.10	0.589±0.10	0.610±0.13
URG	0.448±0.11	0.450±0.11	0.770±0.04	0.792±0.10	0.526±0.01	0.741±0.08	0.460±0.09	0.532±0.15
DEF	0.621±0.10	0.687±0.11	0.857±0.03	0.893±0.07	0.768±0.01	0.793±0.06	0.423±0.06	0.561±0.11
BBL	0.695±0.07	0.715±0.09	**0.918**±0.02	0.947±0.08	0.824±0.02	**0.868**±0.04	0.558±0.09	0.597±0.05
UDA	0.673±0.08	0.695±0.09	0.901±0.02	0.923±0.06	0.727±0.02	0.762±0.02	0.596±0.09	0.602±0.07
EMB	0.704±0.07	0.719±0.08	0.914±0.04	**0.952**±0.05	0.835±0.04	0.852±0.03	0.586±0.06	0.601±0.05
SRE	0.656±0.09	0.714±0.11	0.867±0.05	0.931±0.07	0.720±0.01	0.751±0.05	0.429±0.08	0.521±0.09
REAL-FND	**0.726**±0.11	**0.728**±0.10	0.905±0.10	0.933±0.08	**0.844**±0.05	0.810±0.08	**0.601**±0.09	**0.612**±0.11

Training the RL agent: Once finished with the first two stages, the fake news classifier and the domain classifier are placed as the reward function of the RL setting to train the agent. I train the agent using Algorithm 3 for $2,000$ episodes performing only $T = 20$ actions. As the long term reward function is important, I used a discount

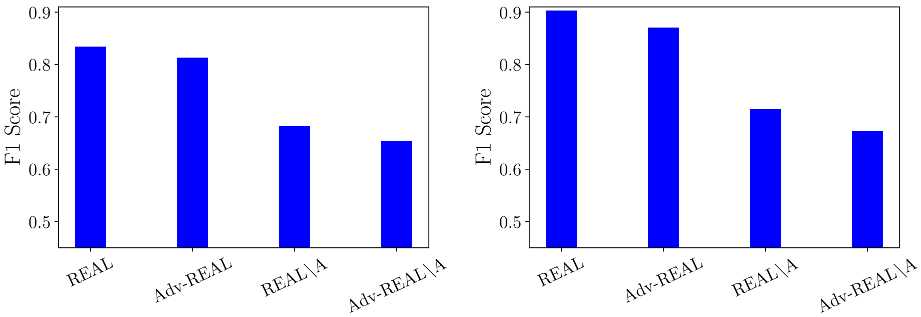


Figure 21. Impact of Different Components of RL-based Fake News Detection

rate of $\lambda = 0.99$. In updating the agent network, Adam optimizer is applied, and the parameters are set as $\alpha = \beta = 0.5$, $\sigma = 0.01$, and learning rate $lr = 1e - 4$. The RL agent uses a feed-forward neural network with 2 hidden layers of 512 and 256 neurons.

7.4.4 Baselines

For evaluating the effectiveness of REAL-FND on the fake news detection task, it is compared to the baseline models described below. For a comprehensive comparison, state-of-the-art baselines that (1) only use news content (BERT-BiLSTM and BERT-UDA), (2) use both news content and users' comments (TCNN-URG and dEFEND), (3) consider users' interactions (CSI), and (4) consider the propagation network (RoBERT-EMB) are considered. In addition to these baselines, two variants of REAL-FND, Simple-REAL-FND and Adv-REAL-FND that use bi-directional GRU instead of BERT/HAN and adversarial training, respectively, are considered. These two baselines help to study the effectiveness of using a complex architecture such as BERT and RL for domain adaptive fake news detection.

- **TCNN-URG (URG)** [98]: Based on TCNN [57], this model uses convolutional neural networks to capture various granularity of news content as well as including users' comments.
- **CSI** [106]: CSI is a hybrid model that utilizes news content, users' comments, and the news source. The news representation is modeled using LSTM neural network using Doc2Vec [62] that outputs an embedding for both news content and users' comments.
- **dEFEND (DEF)** [119]: This model uses a co-attention network to model both news content and users' comments. dEFEND captures explainable content information for fake news detection.
- **BERT-UDA (UDA)** [40]: This model uses feature-based and instance-based domain adaptation on BERT to create domain-independent news content representation.
- **BERT-BiLSTM (BBL)** [137]: This model uses a complex neural network model including BERT [136], bi-directional LSTM layer, and a capsule layer to classify news content. This model uses transfer learning to work across different domains.
- **RoBERTa-EMB (EMB)** [124]: This model adopt instance-based domain adaptation. It uses RoBERTa-base [69] to create news content representation, while uses an unsupervised neural network to create the propagation network's representation [123].
- **Simple-REAL-FND (SRE)**: To study the effect of using complex networks such as BERT and HAN for creating the representation of the news content and users comments, BERT and HAN are replaced with a bi-directional Gated Recurrent Unit (BiGRU) stacked with a location-based attention layer [24].

7.4.5 Experimental Results

To answer questions **Q1** and **Q2**, REAL-FND and the baselines are trained on a source domain S and tested on both source domain S and target domain T. k-fold validation is used and the average AUC and F1 scores are calculated. For single-domain and cross-domain, the number for folds is set as $k = 9$ and $k = 10$, respectively. In single-domain training, the source domain data $\mathcal{D}_s$ is used, while for training REAL-FND and the baselines in a cross-domain setting, the source domain data $\mathcal{D}_s$ combined with portion of the target domain data $\mathcal{D}_t$ is used. In the subsequent subsection it will be shown in Figure 18 that performance improvements taper off after using 30%($\gamma = 0.3$) of the target domain data. Finally, to answer **Q3**, an ablation study is performed to measure the impact of using auxiliary information and reinforcement learning.

Fake News Detection Results (Q1). Table 13 shows a comparison of the baselines with REAL-FND. Training is applied on a single domain dataset $\mathcal{D}_s$ and tested on both source and target domains. For this experiment, I removed the domain classifier's feedback from the RL agent by setting the parameter $\beta = 0$. From the results I conclude that (1) all baseline models and REAL-FND perform reliably well when both training and testing news come from a single domain, and (2) due to the considerable decrease in performance when testing with news from another domain, it appears that the *Politifact* and *GossipCop* news domains have different properties. These results imply that the evaluated models are not agnostic to domain differences. REAL-FND performed better than baselines for the majority of scenarios. The only case where REAL-FND is under-performed is when the model is trained and tested on the *GossipCop* dataset. The better performance of REAL-FND on the *Politifact* domain suggests that BBL may have overfitting issues and REAL-FND benefits the use of auxiliary information for creating a more general fake news classifier that performs well on both domains. It is worth mentioning that EMB and BBL models are similar to each other in terms of model architecture except that EMB utilizes the user's propagation network as well. Comparing the results between these two models also reveals that using auxiliary information can be helpful in detecting fake news.

Cross-Domain Results (Q2). Table 14 shows the performance of models on the target domain. **source** → **target** indicates the models have been trained on the **source** domain and tested on the **target** domain. In this case, I use the source domain data in addition to a small portion of the target domain data. In this experiment, used $\gamma = 0.3$ of nine-folds of the target domain dataset $\mathcal{D}_t$ in addition to all the source domain dataset $\mathcal{D}_s$. Performance measures are calculated based on the tenth-fold of the target domain dataset. The results indicate that REAL-FND detects fake news in the target domain most efficiently, in comparison to the baselines. For example, compared with the best baselines, both F1 and AUC scores have improved. This indicates that most baselines suffer from overfitting on the source domain. Moreover, the results show that using a small portion of the target domain dataset can lead to a large increase in the cross-domain classification, indicating that the RL agent learns more domain-independent features by the feedback received from the domain classifier. It is also notable that Simple-REAL-FND (SRE) perform surprisingly well despite using weaker network architecture than the most baselines.

Impact of RL and Auxiliary Information (Q3). To show the relative impact of using reinforcement learning and auxiliary information, the following variants of REAL-FND are created:

- **REAL-FND\A**: To study the effects of using auxiliary information in fake

Table 14. Cross-domain Fake News Detection Results on Target Domain

Model	GossipCop → Politifact		Politifact → GossipCop	
	AUC	**F1**	**AUC**	**F1**
CSI	0.581±0.03	0.547±0.02	0.612±0.03	0.598±0.02
URG	0.503±0.04	0.486±0.02	0.545±0.04	0.471±0.02
DEF	0.712±0.02	0.634±0.01	0.583±0.02	0.583±0.01
BBL	0.748±0.01	0.711±0.01	0.634±0.01	0.634±0.01
UDA	0.812±0.02	0.778±0.01	0.702±0.01	0.702±0.01
EMB	0.870±0.01	0.876±0.02	0.846±0.03	0.795±0.03
SRE	0.885±0.03	0.881±0.04	0.838±0.03	0.791±0.05
REAL-FND	0.901±0.02	0.892±0.03	0.862±0.05	0.815±0.04

news detection, a variant of REAL-FND is created which does not use users' comments and user-news interactions. Thus, this model only uses BERT to create news article representations.

- **Adv-REAL-FND (ARE)**: To study the effect of the RL agent in domain adaptation, adversarial training is used to create a domain adaptive news article representation. In this model, the representation network, fake news classifier F, ARE domain classifier D are used in an adversarial setting using the following loss function to train a fake news classifier and representation network that is capable of creating domain-independent features.

$$\min_F \max_D \mathcal{L}_{CE}(F) - \mathcal{L}_{CE}(D), \tag{7.5}$$

Where L is the binary cross entropy loss similar to Equation 7.4. In this model, there is no need to pre-train the fake news classifier F and the domain classifier D. Instead, the representation network, fake news classifier F, and domain classifier D is trained as a whole using Equation 7.5.

- **Adv-REAL-FND\A**: In this model, both RL component and the auxiliary information are removed.

According to Figure 21, it is notable that removing the users' comments, and the user-news interactions severely impacts the performance of cross-domain fake news detection. Although using adversarial training for cross-domain fake news detection performs well, reinforcement learning allows us to adopt the news article representation using any pre-trained domain and fake news classifier in the reward function without considering its differentiability.

7.5 Conclusion

Collecting and integrating news articles from different domains and providing human annotations by fact-checking the contents for the purpose of aggregating a dataset are resource-intensive activities that hinder the effective training of automated fake news detection models. Although many deep learning models have been proposed for fake news detection and some have exhibited good results on the domain they were trained on, it is shown in this work that they have limited effectiveness in other domains. To overcome these challenges, the **RE**inforced domain **A**daptation **L**earning for **F**ake **N**ews **D**etection (REAL-FND) task is proposed which could effectively classify fake news on two separate domains using only a small portion of the target domain data. Further, REAL-FND also leverages auxiliary information to enhance fake news detection performance. Experiments on real-world datasets show that in comparison to the current SOTA, REAL-FND adopts better to a new domain by using auxiliary information and reinforcement learning and achieves high performance in a single-domain setting.

Chapter 8

CONCLUSION AND FUTURE WORK

This chapter provides an overview of my research findings and their wider implications while also exploring potential avenues for future research.

8.1 Summary

In this dissertation, I investigate the privacy, fairness, and robustness aspects of trustworthy AI. Principled approaches are provided to exploit social theories to design methods that improve the social aspects of an AI/ML model in challenging environments. Three innovative research tasks are studied in particular: (1) joint utility and privacy-preserving data representation; (2) fairness in sequential data classification; and (3) robust fake news detection. Novel frameworks, mostly based on reinforcement learning, are proposed to tackle these challenges from content and the unique properties of a related social aspect.

In joint utility and privacy-preserving data representation, adversary-based approaches are proposed to leverage a powerful attacker to modify data representations. The proposed frameworks in this area have two goals: (1) preserving users' privacy according to their sensitive attributes (e.g., gender, age, and location); and (2) preserving the usefulness of the learned data representation. Several important findings concerning the learning data representations are: (1) AI/ML models un-intentionally learn users' private attributes and hide them in the data representations; (2) Users' private attributes can be derived from non-private attributes, making feature removal, not a good solution for preserving users' privacy; and (3) AI/ML models often link private attributes with the data label, thus removing private attributes often hurts AI/ML model's performance. These findings are the groundwork for my adversary-based frameworks for creating utility and privacy-preserving data representations.

In fairness for sequential data classification, a reinforcement learning-based approach is proposed to optimize a session-based text classifier to consider the fairness aspect of the input data. In the specific application of cyberbullying detection, it is observed that text classifiers can become biased towards users' sensitive attributes such as dialect. Leveraging fairness measurements, the cyberbullying detection classifier is optimized to re-assign its weights and adapt to users' sensitive information. It is observed that in this scenario, the classifier does not forget users' sensitive attributes but rather uses them to improve its performance and fairness.

To study the robustness aspect of AI/ML methods, fake news detection was chosen to be studied. Early observations of fake news detection methods indicate that such methods require large datasets to perform well and one important challenge in detecting fake news is the lack of data in different domains. Two ways to address this challenge are explored: (1) generating synthetic data using a model-agnostic framework. This method can be used to generate topic-preserving machine-generated news and enables us to further study the differences and similarities between human and machine-generated news; (2) proposing a domain adaptation method that uses domain knowledge to guide a fake news classifier towards learning domain-independent data representations.

Overall, this dissertation investigates the trustworthiness of AI/ML models and their social impact in terms of privacy, fairness, and robustness. With rapid and recent advances in both natural language processing and vision areas, studying these three aspects facilitates further exploration of recent AI/ML models, such as large language models or parameter-efficient tuning methods. The methodologies and techniques presented in this dissertation fall into a novel paradigm - using reinforcement learning to improve social measures - that has important implications for broadening applications in the NLP area.

8.2 Future Work

There are several future directions for using representation learning to create trustworthy AI/ML models in each category:

- Privacy: Although there are many research works on privacy in ML models, there is still room for improvement. Recent advances in large language models (LLM) show that studying these models for privacy violations and creating methods to remove sensitive information from LLMs are necessary. Recent research on LLMs indicates that these models are sensitive to information repetition in training datasets. For example, if a sentence appears once in a dataset, the language models may be able to memorize it [15]. Another work shows that too much repetition also decreases users privacy in LLMs [11]. They further define several metrics based on perplexity to quantify users' privacy in language models. This research can be a stepping stone to moving toward increasing users' privacy in language models.
- Fairness: Given my finding that word- and comment-level semantics impact the performance difference in toxicity detection models, future research can incorporate such a hierarchical structure and mitigate biases in a hierarchical manner. Moreover, the fairness-related approaches could benefit from additional

studies about the ways semantic context influences sequential debiasing. Experiments are done on automating the process of detecting keywords that are sensitive and cause unfairness in the AI/ML model. The results indicate that current classifiers rely on several keywords to predict the label, known as shortcut learning phenomenon [33]. One future direction is to use a reinforced approach to nudge the classifier into using a wider range of keywords for classification, lowering the chances of shortcut learning.

- Robustness: One important enhancement using RL focuses on limited supervision to address the effects of limited or imprecise data annotations by applying weak supervision learning in a domain adaptation setting. The inconsistencies in multimodal information due to the breadth of the types and sources of information in natural language processing applications that are required for cross-domain text classification were addressed. For future work, the effects of data generation and tuning methods on creating a robust model that does not change its decision by a minor change in the input can be studied.
- Combination of properties: Trustworthy AI properties can sometimes correlate with each other. In this direction, the goal is to explore the intersections of privacy, fairness, and robustness. We can first study the interactions and effects of trustworthy AI on each other. Then, proposing methods that can improve several aspects of a trustworthy AI. This direction sets a stepping stone for future research on how to combine trustworthy AI properties. As an example, privacy and fairness can be considered. Fairness can be achieved through two main methods: (1) fairness through attribute removal, and (2) fairness through using sensitive attributes. By choosing fairness through removal of sensitive

attributes, when the sensitive attributes are also considered as private attributes, removing them can affect both privacy and fairness.

REFERENCES

[1] Adversa AI. *The Road to Secure and Trusted AI.* 2022. URL: https://adversa.ai/report-secure-and-trusted-ai/?utm_source=report&utm_medium=report_decade&utm_campaign=latest_version.

[2] Hunt Allcott and Matthew Gentzkow. "Social media and fake news in the 2016 election". In: *Journal of economic perspectives* 31.2 (2017), pp. 211–36.

[3] Amazon. *Rekognition Face Verification API.* https://aws.amazon.com/rekognition/.

[4] Balamurugan Anandan et al. "t-Plausibility: Generalizing Words to Desensitize Text." In: *Trans. Data Privacy* 5.3 (2012), pp. 505–534.

[5] Sarath Chandar AP et al. "An autoencoder approach to learning bilingual word representations". In: *Advances in neural information processing systems.* 2014, pp. 1853–1861.

[6] Ghazaleh Beigi et al. "Privacy-aware recommendation with private-attribute protection using adversarial learning". In: *Proceedings of the 13th International Conference on Web Search and Data Mining.* 2020, pp. 34–42.

[7] Shai Ben-David et al. "A theory of learning from different domains". In: *Machine learning* 79.1-2 (2010), pp. 151–175.

[8] Dimitris Bertsimas, Vivek F Farias, and Nikolaos Trichakis. "On the efficiency-fairness trade-off". In: *Management Science* 58.12 (2012), pp. 2234–2250.

[9] Tolga Bolukbasi et al. "Man is to computer programmer as woman is to homemaker? debiasing word embeddings". In: *Advances in neural information processing systems* 29 (2016), pp. 4349–4357.

[10] Daniel Borkan et al. "Nuanced metrics for measuring unintended bias with real data for text classification". In: *Companion proceedings of the 2019 world wide web conference.* 2019, pp. 491–500.

[11] Hannah Brown et al. "What Does it Mean for a Language Model to Preserve Privacy?" In: *2022 ACM Conference on Fairness, Accountability, and Transparency.* 2022, pp. 2280–2292.

[12] Miles Brundage et al. "Toward trustworthy AI development: mechanisms for supporting verifiable claims". In: *arXiv preprint arXiv:2004.07213* (2020).

[13] Paweł Budzianowski and Ivan Vulić. "Hello, It's GPT-2–How Can I Help You? Towards the Use of Pretrained Language Models for Task-Oriented Dialogue Systems". In: *arXiv preprint arXiv:1907.05774* (2019).

[14] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. "Semantics derived automatically from language corpora contain human-like biases". In: *Science* 356.6334 (2017), pp. 183–186.

[15] Nicholas Carlini et al. "Quantifying memorization across neural language models". In: *arXiv preprint arXiv:2202.07646* (2022).

[16] Mahawaga Arachchige Pathum Chamikara et al. "Privacy preserving face recognition utilizing differential privacy". In: *Computers & Security* 97 (2020), p. 101951.

[17] Jiawei Chen, Janusz Konrad, and Prakash Ishwar. "Vgan-based image representation learning for privacy-preserving facial expression recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018.

[18] Jiawei Chen, Janusz Konrad, and Prakash Ishwar. "Vgan-based image representation learning for privacy-preserving facial expression recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 1570–1579.

[19] Lu Cheng et al. "Hierarchical attention networks for cyberbullying detection on the instagram social network". In: *SDM*. SIAM. 2019, pp. 235–243.

[20] Lu Cheng et al. "Mitigating bias in session-based cyberbullying detection: A non-compromising approach". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021, pp. 2158–2168.

[21] Lu Cheng et al. "Session-based Cyberbullying Detection: Problems and Challenges". In: *IEEE Internet Computing* (2020).

[22] Lu Cheng et al. "Xbully: Cyberbullying detection within a multi-modal context". In: *WSDM*. 2019, pp. 339–347.

[23] Kyunghyun Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[24] Junyoung Chung et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014).

[25] Maximin Coavoux, Shashi Narayan, and Shay B Cohen. "Privacy-preserving Neural Representations of Text". In: *arXiv preprint arXiv:1808.09408* (2018).

[26] Alistair Coleman. *'hundreds dead' because of covid-19 misinformation.* Aug. 2020. URL: https://www.bbc.com/news/world-53755067.

[27] Alexis Conneau et al. "Very deep convolutional networks for text classification". In: *arXiv preprint arXiv:1606.01781* (2016).

[28] George Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.

[29] Sumanth Dathathri et al. "Plug and play language models: a simple approach to controlled text generation". In: *arXiv preprint arXiv:1912.02164* (2019).

[30] Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. "Racial bias in hate speech and abusive language detection datasets". In: *arXiv preprint arXiv:1905.12516* (2019).

[31] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[32] Lucas Dixon et al. "Measuring and mitigating unintended bias in text classification". In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society.* 2018, pp. 67–73.

[33] Mengnan Du et al. "Shortcut learning of large language models in natural language understanding: A survey". In: *arXiv preprint arXiv:2208.11857* (2022).

[34] Gabriel Dulac-Arnold et al. "Deep reinforcement learning in large discrete action spaces". In: *arXiv preprint arXiv:1512.07679* (2015).

[35] Face++. *Face Searching API.* https://www.faceplusplus.com/face-searching/.

[36] Nikhil Garg et al. "Word embeddings quantify 100 years of gender and ethnic stereotypes". In: *Proceedings of the National Academy of Sciences* 115.16 (2018), E3635–E3644.

[37] Oguzhan Gencoglu. "Cyberbullying detection with fairness constraints". In: *IEEE Internet Computing* 25.1 (2020), pp. 20–29.

[38] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. “Learning to forget: Continual prediction with LSTM”. In: (1999).

[39] Mor Geva, Yoav Goldberg, and Jonathan Berant. “Are we modeling the task or the annotator? an investigation of annotator bias in natural language understanding datasets”. In: *arXiv preprint arXiv:1908.07898* (2019).

[40] Chenggong Gong, Jianfei Yu, and Rui Xia. “Unified Feature and Instance Based Domain Adaptation for End-to-End Aspect-based Sentiment Analysis”. In: *EMNLP*. 2020.

[41] Jiaxian Guo et al. “Long text generation via adversarial training with leaked information”. In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[42] Shangwei Guo, Tao Xiang, and Xiaoguo Li. “Towards efficient privacy-preserving face recognition in the cloud”. In: *Signal Processing* 164 (2019), pp. 320–328.

[43] Maryam Heidari and James H Jr Jones. “Using BERT to Extract Topic-Independent Sentiment Features for Social Media Bot Detection”. In: *IEEE 2020 11th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2020*. 2020.

[44] Maryam Heidari, James H Jr Jones, and Ozlem Uzuner. “Deep Contextualized Word Embedding for Text-based Online User Profiling to Detect Social Bots on Twitter”. In: *IEEE 2020 International Conference on Data Mining Workshops (ICDMW), ICDMW 2020*. 2020.

[45] Maryam Heidari and Setareh Rafatirad. “Using Transfer Learning Approach to Implement Convolutional Neural Network to Recommend Airline Tickets by Using Online Reviews”. In: *IEEE 2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization, SMAP 2020*. 2020.

[46] Briland Hitaj, Giuseppe Ateniese, and Fernando Pérez-Cruz. “Deep models under the GAN: information leakage from collaborative deep learning”. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2017, pp. 603–618.

[47] Homa Hosseinmardi et al. “Detection of cyberbullying incidents on the instagram social network”. In: *arXiv preprint arXiv:1503.03909* (2015).

[48] Dirk Hovy, Anders Johannsen, and Anders Søgaard. "User review sites as a resource for large-scale sociolinguistic studies". In: *Proceedings of the 24th International Conference on World Wide Web*. 2015.

[49] Po-Sen Huang et al. "Reducing sentiment bias in language models via counterfactual evaluation". In: *arXiv preprint arXiv:1911.03064* (2019).

[50] Xiaolei Huang et al. "Multilingual Twitter corpus and baselines for evaluating demographic bias in hate speech recognition". In: *arXiv preprint arXiv:2002.10361* (2020).

[51] Md Saiful Islam et al. "COVID-19–related infodemic and its impact on public health: A global social media analysis". In: *The American journal of tropical medicine and hygiene* 103.4 (2020), p. 1621.

[52] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. "Reinforcement learning: A survey". In: *Journal of artificial intelligence research* 4 (1996), pp. 237–285.

[53] Mansooreh Karami, Tahora H Nazer, and Huan Liu. "Profiling Fake News Spreaders on Social Media through Psychological and Motivational Factors". In: *Proceedings of the 32nd ACM Conference on Hypertext and Social Media*. 2021, pp. 225–230.

[54] Mansooreh Karami et al. "" Let's Eat Grandma": When Punctuation Matters in Sentence Representation for Sentiment Analysis". In: *arXiv preprint arXiv:2101.03029* (2020).

[55] Yaser Keneshloo, Naren Ramakrishnan, and Chandan K Reddy. "Deep transfer reinforcement learning for text summarization". In: *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM. 2019, pp. 675–683.

[56] Daniel Kifer and Ashwin Machanavajjhala. "No free lunch in data privacy". In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*. 2011, pp. 193–204.

[57] Yoon Kim. "Convolutional neural networks for sentence classification". In: *arXiv preprint arXiv:1408.5882* (2014).

[58] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[59] Wouter M Kouw et al. “Feature-level domain adaptation”. In: *The Journal of Machine Learning Research* (2016).

[60] Keita Kurita et al. “Measuring bias in contextualized word representations”. In: *arXiv preprint arXiv:1906.07337* (2019).

[61] Ditch the Label Anti Bullying Charity. *Ditch the Label Anti Bullying Charity: The annual cyberbullying survey 2013*. https://www.ditchthelabel.org/wp-content/uploads/2016/07/cyberbullying2013.pdf. Accessed: 2020-09-18. 2013.

[62] Quoc Le and Tomas Mikolov. “Distributed representations of sentences and documents”. In: *ICML*. 2014.

[63] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *nature* 521.7553 (2015), pp. 436–444.

[64] Jiwei Li et al. “Deep reinforcement learning for dialogue generation”. In: *arXiv preprint arXiv:1606.01541* (2016).

[65] Xinlong Li et al. “Enhancing BERT representation with context-aware embedding for aspect-based sentiment analysis”. In: *IEEE Access* 8 (2020), pp. 46868–46876.

[66] Yitong Li, Timothy Baldwin, and Trevor Cohn. “Towards robust and privacy-preserving text representations”. In: *arXiv preprint arXiv:1805.06093* (2018).

[67] Zichao Li et al. “Paraphrase generation with deep reinforcement learning”. In: *arXiv preprint arXiv:1711.00279* (2017).

[68] Chia-Wei Liu et al. “How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation”. In: *arXiv preprint arXiv:1603.08023* (2016).

[69] Yinhan Liu et al. “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692* (2019).

[70] Ziwei Liu et al. “Deep Learning Face Attributes in the Wild”. In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015.

[71] Marco Lui and Timothy Baldwin. “langid. py: An off-the-shelf language identification tool”. In: *Proceedings of the ACL 2012 system demonstrations*. 2012.

[72] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. “Effective approaches to attention-based neural machine translation”. In: *arXiv preprint arXiv:1508.04025* (2015).

[73] Zhuo Ma et al. “Lightweight privacy-preserving ensemble classification for face recognition”. In: *IEEE Internet of Things Journal* 6.3 (2019).

[74] Alireza Makhzani et al. “Adversarial autoencoders”. In: *arXiv preprint arXiv:1511.05644* (2015).

[75] *Malicious cyber-attack*. https://www.theguardian.com/world/2019/jun/10/malicious-cyber-attack-exposes-travelers-photos-says-us-customs-agency. 2019.

[76] Yunlong Mao et al. “A privacy-preserving deep learning approach for face recognition with edge computing”. In: *Proc. USENIX Workshop Hot Topics Edge Comput.(HotEdge)*. 2018, pp. 1–6.

[77] Yunlong Mao et al. “A privacy-preserving deep learning approach for face recognition with edge computing”. In: *Proc. USENIX Workshop Hot Topics Edge Comput.(HotEdge)*. 2018, pp. 1–6.

[78] Microsoft. *Azure Face API*. https://azure.microsoft.com/en-us/services/cognitive-services/face/.

[79] Tomas Mikolov et al. “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[80] Amy Mitchell and Mark Jurkowitz. *Americans Who Mainly Get Their News on Social Media Are Less Engaged, Less Knowledgeable*. Aug. 2020. URL: https://www.journalism.org/2020/07/30/americans-who-mainly-get-their-news-on-social-media-are-less-engaged-less-knowledgeable/.

[81] Volodymyr Mnih et al. “Playing atari with deep reinforcement learning”. In: *arXiv preprint arXiv:1312.5602* (2013).

[82] Ahmadreza Mosallanezhad, Ghazaleh Beigi, and Huan Liu. “Deep Reinforcement Learning-based Text Anonymization against Private-Attribute Inference”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Associa-

tion for Computational Linguistics, Nov. 2019, pp. 2360–2369. URL: https://aclanthology.org/D19-1240.

[83] Ahmadreza Mosallanezhad, Ghazaleh Beigi, and Huan Liu. "Deep reinforcement learning-based text anonymization against private-attribute inference". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019, pp. 2360–2369.

[84] Ahmadreza Mosallanezhad, Kai Shu, and Huan Liu. "Generating Topic-Preserving Synthetic News". In: *2021 IEEE International Conference on Big Data (Big Data)*. IEEE. 2021, pp. 490–499.

[85] Ahmadreza Mosallanezhad et al. "Domain Adaptive Fake News Detection via Reinforcement Learning". In: *Proceedings of the ACM Web Conference 2022*. 2022, pp. 3632–3640.

[86] Ahmadreza Mosallanezhad et al. "Toward Privacy and Utility Preserving Image Representation". In: *SBP Conference* (2020).

[87] Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. "Hate speech detection and racial bias mitigation in social media based on BERT model". In: *PloS one* 15.8 (2020), e0237861.

[88] Arjun Mukherjee and Bing Liu. "Improving gender classification of blog authors". In: *Empirical Methods in natural Language Processing (EMNLP)*. 2010.

[89] Moin Nadeem, Anna Bethke, and Siva Reddy. "Stereoset: Measuring stereotypical bias in pretrained language models". In: *arXiv preprint arXiv:2004.09456* (2020).

[90] Ramesh Nallapati et al. "Abstractive text summarization using sequence-to-sequence rnns and beyond". In: *arXiv preprint arXiv:1602.06023* (2016).

[91] Kishore Papineni et al. "BLEU: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting on association for computational linguistics*. ACL. 2002, pp. 311–318.

[92] Ji Ho Park, Jamin Shin, and Pascale Fung. "Reducing gender bias in abusive language detection". In: *arXiv preprint arXiv:1808.07231* (2018).

[93] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. "Deep face recognition". In: (2015).

[94] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. “A picture of search.” In: *1st international conference on Scalable information systems (InfoScale)*. 2006.

[95] Romain Paulus, Caiming Xiong, and Richard Socher. “A deep reinforced model for abstractive summarization”. In: *arXiv preprint arXiv:1705.04304* (2017).

[96] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “Glove: Global vectors for word representation”. In: *EMNLP*. 2014.

[97] Verónica Pérez-Rosas et al. “Automatic Detection of Fake News”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 3391–3401. URL: https://aclanthology.org/C18-1287.

[98] Feng Qian et al. “Neural User Response Generator: Fake News Detection with Collective User Intelligence.” In: *IJCAI*. 2018.

[99] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks”. In: *arXiv preprint arXiv:1511.06434* (2015).

[100] Alec Radford et al. “Language models are unsupervised multitask learners”. In: *OpenAI Blog* 1.8 (2019).

[101] Rahat Ibn Rafiq et al. “Careful what you share in six seconds: Detecting cyberbullying instances in Vine”. In: *ASONAM 2015*. IEEE. 2015, pp. 617–622.

[102] Marc’Aurelio Ranzato et al. “Sequence level training with recurrent neural networks”. In: *arXiv preprint arXiv:1511.06732* (2015).

[103] Hannah Rashkin et al. “Truth of varying shades: Analyzing language in fake news and political fact-checking”. In: *Proceedings of the 2017 conference on empirical methods in natural language processing*. 2017, pp. 2931–2937.

[104] Zhongzheng Ren, Yong Jae Lee, and Michael S Ryoo. “Learning to anonymize faces for privacy preserving action detection”. In: *Proceedings of the european conference on computer vision (ECCV)*. 2018, pp. 620–636.

[105] Mohammad Al-Rubaie and J Morris Chang. “Privacy-preserving machine learning: Threats and solutions”. In: *IEEE Security & Privacy* 17.2 (2019), pp. 49–58.

[106] Natali Ruchansky, Sungyong Seo, and Yan Liu. “Csi: A hybrid deep model for fake news detection”. In: *CIKM*. 2017.

[107] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *arXiv preprint arXiv:1910.01108* (2019).

[108] Maarten Sap et al. "The risk of racial bias in hate speech detection". In: *Proceedings of the 57th annual meeting of the association for computational linguistics*. 2019, pp. 1668–1678.

[109] Beatrice Savoldi et al. "Gender bias in machine translation". In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 845–874.

[110] Yücel Saygin, Dilek Hakkini-Tur, and Gökhan Tur. "Sanitization and anonymization of document repositories". In: *Web and information security*. 2006.

[111] Tal Schuster et al. "The Limitations of Stylometry for Detecting Machine-Generated Fake News". In: *Computational Linguistics* 46.2 (June 2020), pp. 499–510. eprint: https://direct.mit.edu/coli/article-pdf/46/2/499/1847559/coli_a_00380.pdf. URL: https://doi.org/10.1162/coli%5C_a%5C_00380.

[112] Ramprasaath R Selvaraju et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE international conference on computer vision*. 2017.

[113] Shaban Shabani and Maria Sokhn. "Hybrid machine-crowd approach for fake news detection". In: *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*. IEEE. 2018, pp. 299–306.

[114] Shawn Shan et al. "Fawkes: Protecting privacy against unauthorized deep learning models". In: *29th USENIX Security Symposium (USENIX Security 20)*. 2020, pp. 1589–1604.

[115] Lifeng Shang, Zhengdong Lu, and Hang Li. "Neural Responding Machine for Short-Text Conversation". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*. 2015.

[116] Zhan Shi et al. "Toward diverse text generation with inverse reinforcement learning". In: *27th International Joint Conference on Artificial Intelligence*. 2018.

[117] Zhan Shi et al. "Toward diverse text generation with inverse reinforcement learning". In: *arXiv preprint arXiv:1804.11258* (2018).

[118] Kai Shu, Suhang Wang, and Huan Liu. “Beyond news contents: The role of social context for fake news detection”. In: *WSDM*. ACM. 2019.

[119] Kai Shu et al. “dEFEND: Explainable Fake News Detection”. In: *KDD*. 2019.

[120] Kai Shu et al. “Fake news detection on social media: A data mining perspective”. In: *ACM SIGKDD Explorations Newsletter* 19.1 (2017), pp. 22–36.

[121] Kai Shu et al. “Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media”. In: *arXiv preprint arXiv:1809.01286* (2018).

[122] Kai Shu et al. “Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media”. In: *Big data* 8.3 (2020), pp. 171–188.

[123] Amila Silva et al. “Embedding Partial Propagation Network for Fake News Early Detection.” In: *CIKM (Workshops)*. 2020.

[124] Amila Silva et al. “Embracing Domain Differences in Fake News: Cross-domain Fake News Detection using Multi-modal Data”. In: *arXiv preprint arXiv:2102.06314* (2021).

[125] Peter K Smith et al. “Cyberbullying: Its nature and impact in secondary school pupils”. In: *Journal of child psychology and psychiatry* 49.4 (2008), pp. 376–385.

[126] Gabriel Stanovsky, Noah A Smith, and Luke Zettlemoyer. “Evaluating gender bias in machine translation”. In: *arXiv preprint arXiv:1906.00591* (2019).

[127] Published by Statista Research Department and Sep 7. *Daily social media usage worldwide*. Sept. 2021. URL: https://www.statista.com/statistics/433871/daily-social-media-usage-worldwide/.

[128] Tony Sun et al. “Mitigating gender bias in natural language processing: Literature review”. In: *arXiv preprint arXiv:1906.08976* (2019).

[129] Yueming Sun and Yi Zhang. “Conversational recommender system”. In: *ACM SIGIR Conference on Research & Development in Information Retrieval*. 2018.

[130] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*. Vol. 135. MIT press Cambridge, 1998.

[131] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[132] Richard S Sutton et al. "Policy gradient methods for reinforcement learning with function approximation." In: *NIPs*. Vol. 99. Citeseer. 1999, pp. 1057–1063.

[133] Chandra Thapa et al. "Splitfed: When federated learning meets split learning". In: *arXiv preprint arXiv:2004.12088* (2020).

[134] Nurislam Tursynbek, Aleksandr Petiushko, and Ivan Oseledets. "Robustness threats of differential privacy". In: *arXiv preprint arXiv:2012.07828* (2020).

[135] Hado Van Hasselt, Arthur Guez, and David Silver. "Deep reinforcement learning with double q-learning". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1. 2016.

[136] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[137] George-Alexandru Vlad et al. "Sentence-Level Propaganda Detection in News Articles with Transfer Learning and BERT-BiLSTM-Capsule Model". In: *Proceedings of the Second Workshop on NLP for Internet Freedom*. 2019, pp. 148–154.

[138] Svitlana Volkova et al. "Inferring latent user properties from texts published in social media". In: *29th AAAI Conference on Artificial Intelligence*. 2015.

[139] Ge Wen et al. "Split-Net: Improving face recognition in one forwarding operation". In: *Neurocomputing* 314 (2018), pp. 94–100.

[140] Hu Xu et al. "BERT post-training for review reading comprehension and aspect-based sentiment analysis". In: *arXiv preprint arXiv:1904.02232* (2019).

[141] Lantao Yu et al. "Seqgan: Sequence generative adversarial nets with policy gradient". In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[142] Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu. *Social media mining: an introduction*. Cambridge University Press, 2014.

[143] Rowan Zellers et al. "Defending Against Neural Fake News". In: *arXiv preprint arXiv:1905.12616* (2019).

[144] Guanhua Zhang et al. "Demographics should not be the reason of toxicity: Mitigating discrimination in text classifications with instance weighting". In: *ACL*. 2020.

[145] Jinxue Zhang et al. “Privacy-Preserving Social Media Data Outsourcing”. In: *IEEE INFOCOM Conference on Computer Communications*. 2018.

[146] Junzi Zhang et al. “Sample efficient reinforcement learning with REINFORCE”. In: *arXiv preprint arXiv:2010.11364* (2020).

[147] Xinyi Zhou et al. “Fake news early detection: A theory-driven model”. In: *Digital Threats: Research and Practice* 1.2 (2020), pp. 1–25.

[148] Yuqing Zhu et al. “Private-knn: Practical differential privacy for computer vision”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11854–11862.

[149] Fuzhen Zhuang et al. “Supervised representation learning: Transfer learning with deep autoencoders”. In: *AAAI*. 2015.

[150] Wei Zou et al. “A reinforced generation of adversarial examples for neural machine translation”. In: *arXiv preprint arXiv:1911.03677* (2019).