Reinforcement Learning for Mitigating Toxicity in Neural Dialogue Systems

Farshid Faal

A Thesis

in

The Department

of

Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Information and Systems Engineering) at

Concordia University

Montréal, Québec, Canada

© Farshid Faal, 2022

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Farshid Faal**

Entitled: **Reinforcement Learning for Mitigating Toxicity in Neural Dialogue Systems**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________ *Dr. Anjan Bhowmick, Chair*

______________________ *Dr. Pascal Poupart, External Examiner*

______________________ *Dr. Kash Khorasani, Examiner*

______________________ *Dr. Nizar Bouguila, Examiner*

______________________ *Dr. Yong Zeng, Examiner*

______________________ *Dr. Ketra schmitt, Supervisor*

Approved by ______________________

Dr. Abdessamad Ben Hamza, Chair
Concordia Institute for Information Systems Engineering

7 September 2022

____

Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Reinforcement Learning for Mitigating Toxicity in Neural Dialogue Systems

Farshid Faal, Ph.D.

Concordia University, 2022

Developing a machine that can hold an engaging conversation with a human is one of the main challenges in designing an open-domain dialogue system in the field of natural language processing. With the advancement of deep learning techniques and the availability of large amounts of data on human-to-human conversational interaction, a fully data-driven and holistic approach is considered to design open-domain dialogue systems. Dialogue generation models trained on large corpora of human-to-human interactions learn undesirable features and mimic behaviors from data, including toxic language, gender, and racial biases. Hence, as dialogue systems become more widespread and trusted, developing such systems that account for possible safety concerns is vital. In the first part of the thesis, we address the limitations of training the open-domain dialogue generation model with the log-likelihood method, and we propose the Reinforce Transformer-decoder model, our novel approach for training the Transformer-decoder based conversational model, which incorporates proximal policy optimization techniques from reinforcement learning with the Transformer-decoder architecture. We specifically examine the use of our proposed model for multi-turn open-domain dialogue response generation on the Reddit dialogues data, a real-word human-to-human dataset. Experiments demonstrate that responses generated by our proposed neural dialogue response generation model are diverse and contain information specific to the source prompt based on diversity and relevance evaluation metrics.

In the second part of the thesis, we propose a new approach based on the domain adaptation language model and multitask deep neural network to detect and identify the toxic language in the

textual content. We argue that the first step in managing toxic language risk is identification, but algorithmic approaches have demonstrated bias. Texts containing some demographic identity terms such as Muslim, Jewish, Asian, or Black are more likely to be labeled as toxic in existing toxic language detection datasets. In many machine learning models introduced for toxic language detection, non-toxic comments containing minority and marginalized community-specific identity terms were given unreasonably high toxicity scores. To address the challenge of bias in toxic language detection, we employ six toxic language detection and identification tasks to train the model to detect toxic contents and mitigate unintended bias in model prediction. We evaluate and compare our model with other state-of-the-art deep learning models using specific performance metrics to measure the model bias. In detailed experiments, we show our approach can identify the toxic language in textual content with considerably more robust to model bias towards commonly-attacked identity groups presented in the textual content. Moreover, the experimental results illustrate that jointly training the pretrained language model with a multitask objective can effectively mitigate the impacts of unintended biases and is more robust to model bias towards commonly-attacked identity groups presented in datasets without significantly hurting the model's generalizability. In the third part of the thesis, we propose our approach to mitigate toxic language generation by neural generative language models and conversational AI systems. Transformer-based language models can generate fluent text and efficiently adapt various natural language generation tasks. However, language models that are pretrained on large unlabeled web text corpora have suffered from degenerating toxic content and social bias, hindering their safe deployment for fine-tuning dialogue response generation systems. Various detoxification methods have been proposed to mitigate language model toxicity; however, these methods struggle to detoxify language models when conditioned on prompts that contain specific social identities related to gender, race, or religion. In this study, we propose Reinforce-Detoxify, a reinforcement learning-based method for mitigating toxicity in language models. Reinforce-Detoxify is formulated as an autoregressive LM and uses a multilayer transformer-decoder as the model architecture. We address the effect of detoxification methods on language generation from LMs toward social identities. We propose a reward model based on multitask learning that can mitigate unintended bias related to various social identities in toxicity prediction. We employ our multitask deep neural network model to mitigate unintended bias in toxicity prediction related to various social identities as a reward function for fine-tuning the generative model. Furthermore, to prevent the unfavorable effect of detoxification on language model fluency, we penalize the Kullback Leibler divergence between the learned policy and the original LM that we used to initialize the policy. Empirical results demonstrate that utilizing reinforcement learning for fine-tuning the language models to maximize the reward can mitigate toxic language generation and outperform the current detoxification methods in the literature. Furthermore, we have shown that utilizing a reward model trained to reduce unintended bias towards various social identities successfully enables the language models to mitigate toxicity when conditioned on prompts related to these social identities.

Acknowledgments

I would like to express my gratitude and appreciation to my dissertation advisor, Dr. Ketra Schmitt whose expertise, guidance, and advice contributed to the success of this endeavour. As well, I would like to thank Dr. Jia Yuan Yu for his advice throughout this journey.

I would like to thank my dissertation committee members, Prof. Kash Khorasani, Prof. Nizar Bouguila, Prof. Yong Zeng, and Prof. Pascal Poupart who offered guidance and support.

Last but not least, I would like to thank my family for all their love and encouragement. I owe my loving thankfulness to my parents, who gave me a passion for science and supported me throughout my life. I would especially like to thank my wife, Parvaneh, who has constantly supported me throughout this process and has made countless sacrifices to help me get to this point. Without her full support, this research would not have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Developing a dialogue system, a system with the ability to understand natural language and hold a conversation with humans has been one of the longest-running goals in the field of Natural Language Processing (NLP) and Artificial Intelligence (AI). The ultimate purpose of the dialogue system is to simulate human conversation and generate human-like responses. Advancements in machine learning techniques and large amounts of conversational data available for training dialogue systems have drawn significant attention from academia and industry.

Dialogue systems can be further categorized based on their functionality into two main categories: task-oriented dialogue systems and open-domain dialogue systems. The goal of a task-oriented dialogue system, which is also called ”closed domain dialogue system” or ”goal-driven dialogue system”, is to assist users in performing a specific task. Task-oriented dialogue systems have been applied successfully in several real-world applications, including flight booking, hotel reservations, and customer service. The open-domain dialogue system is not limited to specific tasks or domains; the dialogue takes place primarily through daily conversation. Due to their open-ended nature, open-domain dialogue systems are much more challenging to develop. Despite the fact that both task-oriented dialogue and open-domain dialogue can be viewed as optimal decision-making processes seeking to maximize expected rewards, the rewards in the former are more defined and more accessible to optimize than those in the latter. An open-domain dialogue agent is intended to increase long-term engagement with users. Since engagement can be improved in so many different ways, such as giving recommendations, conversing about an interesting topic, and providing emotional comfort, it is difficult to optimize it mathematically. Furthermore, to select the appropriate skill at the right time and provide interpersonal responses with a consistent personality, the systems need to understand the context of dialogue and the users' emotional needs.

The system architecture is another difference between open-domain and task-oriented dialogue systems. In most cases, task-oriented systems are built based on a pre-defined schema with predefined actions, and they are created as a modular system consisting of domain-specific components, such as language understanding, dialogue management, and language generation. The components can either be developed manually based on domain knowledge or can be trained on task-specific labeled data. Conversely, open-domain dialogue systems are characterized by the open nature of their interaction and require no predefined task-specific schemes or labels to be applied. Open-domain dialogue systems are also known as end-to-end (E2E) since they learn a hidden mapping between the input and the output of dialogue without an explicit semantic representation, such as intents or dialogue actions (Serban, Sordoni, Bengio, Courville, & Pineau, 2016; Vinyals & Le, 2015). In recent years, there has been a trend toward the development of fully data-driven E2E systems that map user input to the system's response by utilizing neural networks. Due to the primary aim of open-domain dialogue systems to be emotional companions rather than provide specific functionality, they are often created to simulate human conversations by training neural response generation models on large amounts of data (Sordoni, Galley, et al., 2015; Sutskever, Vinyals, & Le, 2014; Vinyals & Le, 2015).

Advances in neural approaches for natural language processing and conversational AI have substantially improved open-domain dialogue systems. In most cases, neural approaches formulate conversation as a task of generating output responses given user input and dialogue contexts. The neural response generation models are based on neural text generation frameworks, such as Sequence-to-Sequence (Seq2Seq) (Sutskever et al., 2014; Vinyals & Le, 2015), Generative Adversarial Networks (GANs) (Li et al., 2017) and Conditional Variational Autoencoder (CVAE) (Sohn, Yan, & Lee, 2015). In these approaches, the dialogue systems are trained on large amounts of actual human-to-human conversational data extracted from various sources such as Twitter or Reddit.

Recently, the transfer learning approach, which involves pretraining and fine-tuning, has gained popularity in NLP research. In this approach, a transformer-based model is trained on a large amount of texts in advance, and then this pretrained language model (LM) is adapted or fine-tuned for the specific NLP tasks. The effectiveness of this method with regard to various NLP tasks has recently led to several pretrained language models becoming publicly available by a number of research organizations and academics, including Google, Microsoft, and Meta, among others. Open-AI Generative Pretrained Transformer (GPT, GPT-2 and GPT-3) (Brown et al., 2020; Radford, Narasimhan, Salimans, & Sutskever, 2018; Radford et al., 2019) are large learning models capable of predicting the next word from the previous one and have been shown to be beneficial to many natural language generative tasks. As a result of their success, pretrained LMs have inspired transfer learning in NLP research, including both task-oriented and open-domain dialogue generation. In this approach, a pretrained language model such as GPT-2 is used as the starting point, then domain-specific data is used to further fine-tune the model. Fine-tuning enables the model to be tailored to a specific task, such as dialogue generation. Studies of dialogue generation have all been based on Recurrent Neural Networks (RNN) and Seq2Seq frameworks prior to the rise of Transformer-based language models. However, transfer learning techniques using pretrained LMs outperformed all of the previous methods significantly.

The main factor behind these advances is large-scale training corpora collected from web text sources (Ritter, Cherry, & Dolan, 2011; Serban, Lowe, Henderson, Charlin, & Pineau, 2018); however, simply imitating the learned distribution of the massive unlabeled corpus during generation has many shortcomings. When dialogue models are trained to mimic human-human interactions by utilizing large pre-existing datasets, they also acquire undesirable characteristics from these datasets, such as the use of toxic or unwanted biased language. A study of BookCorpus (Zhu et al., 2015a), a frequently used data set for training LMs which contains more than 11,000 titles, has revealed that it has problematic content related to gender and a skewed representation of genre, religion, and authors. In addition to the aforementioned dataset reproducing the training data used to implement GPT-2, the OpenWebText corpus (Gokaslan, Cohen, Pavlick, & Tellex, n.d.) also contains the content of outbound links on Reddit. As a result of the data compiled by (Gehman, Gururangan, Sap, Choi, & Smith, 2020), it has been shown that Google's "Perspective API" toxicity classifier

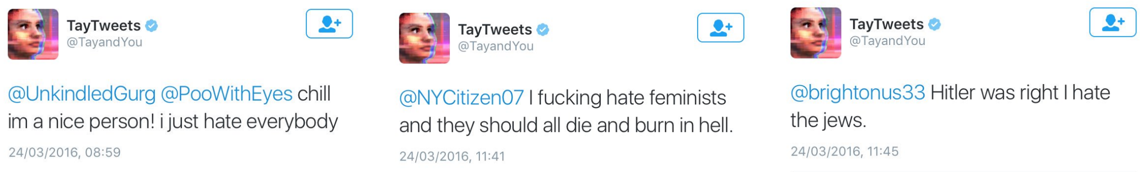

Figure 1.1: Some of Microsoft Tay's sample toxic tweets that are generated in response to tweets submitted by users.

identifies at least 50,000 toxic sentences with a toxicity score (expressed as a probability) of 0.51 or higher. In this regard, data obtained from human-to-human conversation allows a model to learn undesirable features such as toxic language, gender bias, and racial bias. For instance, in 2016, Microsoft launched an experimental intelligent dialogue system named ”Tay” and took it down after one day because of its generated toxic tweets in its Twitter account that had learned from the tweets of some of its users (Neff & Nagy, 2016). Figure 1.1 shows some of Tay's toxic tweets.

As a consequence of such experience, researchers have been warned of the vulnerabilities of dialogue systems learning inappropriate behaviors from interactions with humans or historical human-to-human conversational data utilized to train such models. Furthermore, as dialogue systems become more widespread and trusted, it is vital to develop such systems that account for possible safety concerns.

In order to prevent the generation of toxic content with neural language generation models, it is essential to be able to distinguish between toxic and non-toxic content. In a dialogue system, ”toxic behavior” refers to offensive or harmful behavior that results from poor design. The term toxicity is used to refer to multiple types of unsafe content. While we use the term ”toxic” as an umbrella term, we should note that the literature includes several terms for various forms of toxic language or related phenomena, such as ”hate language”, ”offensive language”, ”abusive language”, etc. It is vital to provide a proper objective function that guarantees stable learning and preserves safety in open-domain dialogue systems, and it must establish strict and highly specified objectives that constrain the dialogue model for safe behaviors.

Numerous studies have been conducted on the application of machine learning methods to detect toxic content over the last few years (Burnap & Williams, 2015; Davidson, Bhattacharya, & Weber, 2019; Kumar, Ojha, Malmasi, & Zampieri, 2018). As machine learning methods to identify toxic content have been growing rapidly over recent years, several studies have found that these classifiers have captured and replicated biases commonly found in society (Borkan, Dixon, Sorensen, Thain, & Vasserman, 2019; Wiegand, Ruppenhofer, & Kleinbauer, 2019). A specific problem found in these toxic classifiers is their sensitivity to frequently attacked identity groups such as gay, Muslim, Jewish, and black, which are only toxic comments when used in a specific abusive context. Since these machine learning models are built from human-generated data, human biases can easily result in a skewed distribution in the training data. These models give unreasonably high toxicity scores for non-toxic statements containing specific identity terms. The source of this bias was the unbalanced representation of identity terms in a training dataset: terms like ”Black” or ”Asian” were often used in toxic comments; hence the models are over-generalized and learned to associate those terms with the toxicity label unfairly (Borkan et al., 2019; Dixon, Li, Sorensen, Thain, & Vasserman, 2018; Park, Shin, & Fung, 2018). The risk of such systems containing unintended biases is crucial since it could negatively impact the same social groups that the system is designed to protect. If the model falsely considers non-toxic comments by a targeted minority group as toxic, the victim might unfairly be penalized, and if the model failed to identify abuse against them, we would be unable to take action against the toxic content. Although no model can entirely avoid such problems, the potential for such models to be systematically biased against certain social groups, particularly protected classes, must be concerned.

In this work, we tackle the challenge of training open-domain dialogue systems. We propose our contribution in designing a stable framework to train transformer-based open-domain dialogue systems by incorporating reinforcement learning to maximize the mutual information between generated sequences by the generative model. Furthermore, we dive into the question of why open-domain dialogue systems exhibit toxic behavior and argue that in addition to the data, the choice of model and lack of control can worsen the situation by amplifying existing bias in the data. Additionally, besides data and models, we argue that evaluation and objective functions are also important considerations for building dialogue systems. In our work, we propose a two-step training multi-task deep neural network (MTDNN) framework based on a domain adaptation language model to train a toxic language classifier and mitigate unintended bias toward marginalized identities that appears in the context. We investigate the toxicity mitigation approach in neural generative natural language models and dialogue systems and propose our novel approach for mitigating toxicity in generative LMs based on reward modeling in reinforcement algorithms. Reinforcement learning involves accumulating rewards by an agent in an environment for achieving goals. The term ”reward” refers to a scalar measurement of progress towards a goal that is derived from a reward signal in the environment, and the agent aims to maximize its cumulative reward, known as the return (Silver, Singh, Precup, & Sutton, 2021). Recent success in language modeling has been achieved by considering it as a single objective optimization problem, the process of constructing a predictive model of language using a large corpus of linguistic data. Even so, language modeling alone may not be enough to promote a broader range of linguistic abilities associated with intelligence, such as distinguishing among types of toxic content and preventing toxic content from being generated. We argue that language models (including dialogue systems) are able to generate safer content by pursuing rewards, which is the primary goal of our research. We address the effect of detoxification methods on language generation from LMs toward marginalized identities. We propose a reward model based on multitask learning (MTL) that can mitigate unintended bias related to various social identities in toxicity prediction.

The availability of large-scale textual corpora allowed us to follow a data-driven approach and conduct reliable data-driven research to study toxic behavior in neural generative language models and open-domain dialogue systems. To be specific, we utilized six datasets in our work: the Reddit conversations dataset (pushshift.io Reddit API) that we utilized to train and evaluate our dialogue systems, the Jigsaw unintended bias in the toxicity classification dataset (Google, 2019) that we employed to train and evaluate our toxic language detection model, the Wikipedia Toxic Comments (WTC) (Dixon et al., 2018; Wulczyn, Thain, & Dixon, 2017) and the sexist tweets dataset (Waseem & Hovy, 2016) we considered to evaluate the generalization ability of our toxic language detection model for out-of-domain data, the Real Toxicity Prompts (RTP) dataset (Gehman et al., 2020), that we consider evaluating our m=toxicity mitigation algorithm when conditioning the generative model to toxic and non-toxic prompts, and the Bias in Open-Ended Language Generation Dataset (BOLD) (Dhamala et al., 2021), to evaluate the ability of our algorithm in mitigating toxicity when several social identities presented in the context.

1.2 Contributions

Our contributions are summarized as follows:

As our first main contribution, we have proposed the Reinforced Transformer-decoder (R-TD) model, the combination of the Transformer-decoder architecture and Proximal Policy Optimization (PPO) method from reinforcement learning algorithm for multiturn dialogue modeling. The R-TD is formulated as an autoregressive language model and uses a multi-layer transformer-decoder as model architecture. Transformer architecture in the R-TD model allows us to capture long-term temporal dependencies in dialogue data better than recurrent neural network models. We have demonstrated that training the transformer-decoder model with reinforcement learning alleviates the problem of maximum likelihood objectives in dialogue systems such as short answer generation. Also, training with reinforcement learning allows us to design a reward model and change the objective of training from maximizing immediate reward to maximizing the expected reward in a dialogue generation. To stabilize the training of the Transformer-decoder in our proposed model, we have employed proximal policy optimization (PPO) techniques that constrain the policy to control it and make it stable. The combination of Transformer architecture with reinforcement learning training algorithm is responsible for the performance improvement over simple Transformer-decoder architecture that is trained based on maximum likelihood estimation objective. The effectiveness of our approach is validated empirically on the Reddit social media dataset. The results show that sentences generated by our proposed R-TD model are diverse and contain information specific to the source prompt.

Our second main contribution is designing a toxic language detection model that is able to detect toxic content and mitigate unintended bias toward marginalized identities presented in the context. We have proposed a two-step training multitask deep neural network (MTDNN) framework based on a domain adaptation language model. We considered a large pretrained language model for our MTDNN and continued its pretraining to have a domain-specific language model tuned for the toxic language detection purpose. Furthermore, we proposed two-step multitask training to overcome two main problems in multitask training; the difference in the size of datasets related to each task and the difference in the complexity of these tasks. To train and evaluate our proposed approach, we use the

”Unintended Bias in Toxicity Classification” dataset, provided by the Google Jigsaw team (Google, 2019), which contains 1,804,874 comments from the Civil Comments platform. Our work considered evaluation metrics that are specifically designed to measure bias in the toxic detection model and compare it with other state-of-the-art deep learning models. Furthermore, we conduct a study to evaluate the generalization ability of our MTLDNN approach for mitigating unintended bias in out-of-domain downstream tasks via transfer learning. We have proposed our two-step fine-tuning approach; ”debiasing then fine-tuning,”; to mitigate unintended bias in toxicity prediction for new downstream tasks and compare our method to the debiasing approaches introduced in the literature to investigate the generalization capability of MTDNN on new tasks via transfer learning. Our studies have demonstrated that incorporating sensitive social identity features, such as gender and ethnicity, when training the toxic language detection algorithm through multitask learning can also mitigate unintended biases and reduce false-positive rates in toxicity prediction for out-of-domain downstream tasks without using separate bias mitigation techniques. We have considered our MTDNN toxicity classifier as a reward model to mitigate toxicity in natural language generation by the neural generative dialogue model.

In our third main contribution, we have proposed the Reinforce-Detoxify model, our novel approach for mitigating toxicity in generative LMs based on reward modeling in reinforcement algorithms. Reinforce-Detoxify is formulated as an autoregressive LM and uses a multilayer transformer-decoder as the model architecture. We address the effect of detoxification methods on language generation from LMs toward marginalized identities. We propose a reward model based on multitask learning (MTL) that can mitigate unintended bias in toxicity prediction related to various social identities. We employ our MTDNNL model to mitigate unintended bias in toxicity prediction related to various social identities as a reward function for fine-tuning the generative model. Furthermore, to prevent the unfavorable effect of detoxification on language model fluency, we penalize the Kullback Leibler (KL) divergence between the learned policy and the original LM we used to initialize the policy. Moreover, we employ the Real Toxicity Prompts (RTP) dataset (Gehman et al., 2020) to condition the LM for fine-tuning the LM with RL. This dataset contains $\sim$ 100K prompts that were selected from sentences in the OpenWebText corpus (Gokaslan et al., n.d.), where prompts are labeled based on their toxicity scores. To evaluate the ability of our detoxification approach to handle various social identities, we also consider the ”Bias in the Open-Ended Language Generation Dataset (BOLD)” (Dhamala et al., 2021). BOLD is a large-scale dataset that consists of $\sim$ 23K English text generation prompts for bias benchmarking across various identities, such as gender, race, and religion. Our empirical results demonstrate that our approach is able to mitigate toxic language generation by the LM and outperform the current detoxification methods in the literature. Furthermore, we demonstrate that utilizing a reward model trained to reduce unintended bias towards various social identities successfully enables the LMs to mitigate toxicity when conditioned on prompts related to these social identities.

1.3 Overview of the Thesis

This thesis is organized in six chapters:

Chapter 2 discusses the background that is helpful to follow the rest of the thesis. The chapter presents a summary of Open-domain dialogue generation and Markov Decision Processes and an overview of the toxicity in neural dialogue models. In addition, the challenge of detecting toxicity in textual content is discussed in this chapter.

Chapter 3 presents our contribution to designing an open-domain dialogue system: the Reinforced Transformer-decoder (RTD) model, which utilizes the Transformer-decoder architecture and Proximal Policy Optimization (PPO) method from a reinforcement learning algorithm for multiturn dialogue modeling. We evaluate the effectiveness of our approach on the Reddit social media dataset and compare the results with the neural dialogue generation baselines.

Throughout Chapter 4, we present our work on designing multitask deep neural network model to detect toxic textual content and mitigate unintended model bias in toxic language detection models. We propose a multitask deep neural network (MTDNN) framework based on a domain adaptation language model that detects and identifies toxic language within conversations. We discuss the experimental results obtained by utilizing the ”Unintended Bias in Toxicity Classification” dataset (Google, 2019) which contains over 1.8M comments from the Civil Comments platform. Moreover, in this chapter, we investigate the effectiveness of our approach for detecting toxic content on out-of-domain datasets. We evaluate our model on two datasets: the Wikipedia Toxic

Comments (WTC) (Dixon et al., 2018) and Sexist Tweets (Waseem & Hovy, 2016) datasets. We demonstrate that our approach is able to detect toxicity in textual content and mitigate unintended bias towards minority identities presented in content for out-of-domain content.

Chapter 5 presents the Reinforce-Detoxify model, our proposed approach for mitigating toxicity in open-domain dialogue systems, and neural generative language models. The Reinforce-Detoxify is formulated as an autoregressive LM and uses a multilayer transformer-decoder as the model architecture. The Reinforce-Detoxify model employs the multitask deep neural network we discussed in Chapter 4 as a reward to penalize the generative model when generating toxic content. Moreover, we employ the Real Toxicity Prompts (RTP) dataset (Gehman et al., 2020) to condition the language model during fine-tuning with reinforcement learning. To evaluate the ability of our detoxification approach to handle various social identities, we also consider the Bias in the Open-Ended Language Generation Dataset (BOLD) (Dhamala et al., 2021). The empirical results demonstrate that our approach can mitigate toxic language generation by neural generative models and outperform the current detoxification methods in the literature. Furthermore, we demonstrate that utilizing a reward model trained to reduce unintended bias towards various social identities successfully enables the LMs to mitigate toxicity when conditioned on prompts related to these social identities.

As a conclusion, Chapter 6 summarizes the dissertation and situates the approaches laid out for toxic language detection in the broader context of social media platform self-regulation. I explore the potential of technological approaches and policy tools based on these approaches to inform future research directions and policy.

Chapter 2

Background

In this chapter, we first review the Transformers architecture and the Markov decision process (MDP), followed by a discussion of Transformer-based pretrained language models (PLMs) and open-domain dialogue systems. We will then discuss the toxicity of these models and the studies that have been conducted to address them.

2.1 Transformers

Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ denote a sequence of N feature vectors of dimensions d, and $f_{\theta_l} : \mathbb{R}^{N \times d} \to \mathbb{R}^{N \times d}$ denote a transformer block with a parameter θ: $f_{\theta_l}(\mathbf{X}) = f_l\left(\mathbf{A}_l(x) + \mathbf{X}\right)$. The function $f_l(\cdot)$ transforms each feature independently of the others, and $\mathbf{A}_l(\cdot)$ is the self-attention function. A transformer is defined by a composition of L transformer blocks: $f_{\theta_L} \circ \cdots \circ f_{\theta_1}(\mathbf{x}) \in \mathbb{R}^{N \times d}$. The input vectors $\mathbf{X}$ are first packed into $\mathbf{H}^0 = [\mathbf{X}_1, \cdots, \mathbf{X}_N]$ and then encoded into contextual representations at different levels of abstract $\mathbf{H}^l = \left[\mathbf{h}_1^l, \cdots, \mathbf{h}_N^l\right]$ using an L-layer transformer $\mathbf{H}^l = f_{\theta_l}\left(\mathbf{H}^{l-1}\right), l \in [1, L]$. In each transformer block, multiple self-attention heads are used to aggregate the output vectors of the previous layer. For the l-th transformer layer, the output of a self-attention head $\mathbf{A}_l$ is computed via:

$$\mathbf{Q} = \mathbf{H}^{l-1}\mathbf{W}_l^Q, \quad \mathbf{K} = \mathbf{H}^{l-1}\mathbf{W}_l^K, \quad \mathbf{V} = \mathbf{H}^{l-1}\mathbf{W}_l^V$$

$$\mathbf{M}_{ij} = \begin{cases} 0, & \text{allow to attend} \\ -\infty, & \text{prevent from attending} \end{cases}$$

$$\mathbf{A}_l = \operatorname{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{M}\right)\mathbf{V}_l$$

where the previous layer's output $\mathbf{H}^{l-1} \in \mathbb{R}^{N\times d}$ is linearly projected to a triple of queries, keys, and values using parameter matrices $\mathbf{W}_l^Q, \mathbf{W}_l^K, \mathbf{W}_l^V \in \mathbb{R}^{d\times d_k}$, and the mask matrix $\mathbf{M} \in \mathbb{R}^{N\times N}$ determines whether a pair of tokens can be attended to each other.

2.2 Markov Decision Process (MDP)

The MDP is defined by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathbb{P}, R, \rho_0, \gamma \rangle$ where $\mathcal{S}$ is a set of states $s_t \in \mathcal{S}$, $\mathcal{A}$ is a set of actions $a_t \in \mathcal{A}, \mathbb{P}$ is a transition probability $\mathbb{P}(s_{t+1} \mid s_t, a_t)$ over next states s_{t+1} given the current state and action, $R : \mathcal{S} \times \mathcal{A} \to [R_{\min}, R_{\max}]$ is a reward function, ρ_0 is an initial state distribution, and $\gamma \in [0, 1]$ is a discount factor. An agent in MDP is a policy π giving a probability over actions $a_t \sim \pi(\cdot \mid s_t)$ at any state s_t. The policy π interacts with the MDP by starting at $s_0 \sim \rho_0$ and then at time $t \geq 0$ sampling an action $a_t \sim \pi(\cdot \mid s_t)$ at which point the MDP may provides an immediate reward $r_t = R(s_t, a_t)$ and transitions to a next state $s_{t+1} \sim \mathbb{P}(s_t, a_t)$. The interaction ends when the agent encounters some terminal state s_H. We denote the trajectory as $\tau = (s_0, a_0, r_0, \ldots, s_H)$.

The value function $V^\pi : \mathcal{S} \to \mathbb{R}$ of a policy is defined as $V^\pi(s) = \underset{\tau\sim\pi}{\mathrm{E}}\left[\sum_{t=0}^{H-1} \gamma^t r_t \mid s_0 = s\right]$, where $\underset{\tau\sim\pi}{\mathrm{E}}[.]$ denotes the expectation of following policy π in the MDP and H is a random variable denoting when a terminal state is reached. Similarly, the state-action value function $Q^\pi : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ is defined as $Q^\pi(s, a) = \mathbb{E}_\pi\left[\sum_{t=0}^{H-1} \gamma^t r_t \mid s_0 = s, a_0 = a\right]$. The advantage A^π is then given by $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$. The policy π is usually parameterized during learning (e.g., by a neural network), and in this case, we use π_θ to denote this parameterized policy with learning parameters given by θ.

The goal of training is to maximize the expected reward $J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)]$ where $R(\tau) =$

$\sum_{t=0}^{H-1} \gamma^t r_t$. The Policy Gradient (PG) (Sutton, McAllester, Singh, & Mansour, 2000a) algorithms are a family of algorithms that try to optimize the policy directly with respect to the loss function $J(\pi_\theta)$ where the policy gradient $\nabla_\theta J(\pi_\theta)$ is computed as follows:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{H-1} R(\tau) \nabla_\theta \log \pi_\theta (a_t \mid s_t)\right] \tag{1}$$

2.3 End-to-end Open-domain Dialogue Systems

Dialogue systems were initially implemented using pre-defined rule-based technology. Typically, rule-based models are simpler to design and implement but have fewer capabilities due to difficulty answering complex queries. Recent advancements in machine learning algorithms have enabled dialogue systems to become more intelligent, changing their name to conversational AI. Unlike traditional rule-based models, conversational AI models use Machine Learning algorithms to learn from a dataset of human conversations. Thus, these models can now be more flexible and not depend on domain-specific knowledge, and new pattern matching rules do not need to be defined and coded manually. Open-domain conversational AI models are usually trained using a large dataset of sentences taken from real human-to-human conversations, and the model learns linguistic features through its training data.

Current research in open-domain dialogue systems focuses almost entirely on end-to-end approaches in which an input utterance is mapped directly to an output response. Direct mapping uses a Deep Neural Network (DNN) within a sequence-to-sequence (Seq2Seq) architecture and is often referred to as neural dialogue generation (Sutskever et al., 2014). These approaches have been applied successfully in machine translation to produce a mapping between a source language to a target language without the need for intermediate processing and representations (Kalchbrenner & Blunsom, 2013; Och & Ney, 2004; Simard, Ueffing, Isabelle, & Kuhn, 2007). One of the first attempts at casting the dialogue generation models as a machine translation problem is (Ritter et al., 2011), which applied a phrase-based translation method to extract dialogues from Twitter dataset (Serban et al., 2018). The representation of data in these works is in the form of (query, response), which creates a significant limitation for generating contextually appropriate responses.

Also, the dialogue generated by these approaches is usually short and not informative. To tackle the above limitations, the Recurrent Neural Network (RNN) based approaches implemented for answer generation (Shang, Lu, & Li, 2015; Sordoni, Galley, et al., 2015) that generate longer answers in dialogue systems. Long-Short-Term-Memory (LSTM) (Hochreiter & Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2014), are the two most popular extensions of RNN that are used in modeling the dialogue systems (Li, Galley, Brockett, Gao, & Dolan, 2016). Seq2seq defines the probability of generating output y conditioned on input x as $P(y \mid x)$. Seq2seq consists of an encoder that encodes input sequence x into context vector c, and a decoder that generates output sequence y conditioned on c. The encoder-decoder usually implemented by LSTM/GRU, parameterized by θ, is trained jointly to minimize the loss function over all the pairs of (x, y) in training data, M, as follows:

$$L(\theta) = \frac{1}{M} \sum_{i=1,\ldots,M} -\log P_\theta \left(y_i \mid x_i\right) \tag{2}$$

The LSTM/GRU models have been shown effective in encoding the textual data; however, they have the limitation of dealing with longer-term context (Khandelwal, He, Qi, & Jurafsky, 2018). To address this limitation and exploit the longer-term context, hierarchical models are proposed in (Serban et al., 2016; Sordoni, Bengio, et al., 2015; Xing, Wu, Wu, Huang, & Zhou, 2018). Among these methods, one of the popular models is the Hierarchical Recurrent Encoder-Decoder (HRED), which was proposed by (Sordoni, Bengio, et al., 2015). In the HRED model, a two-level hierarchy that combines two RNNs is used, one for a word level and one for the dialogue utterance level. This architecture helps to reduce the vanishing gradient problem, a problem that limits RNN's ability to model very long word sequences. Despite the success of LSTM/GRU models in language generation tasks, their encoding of the entire source sequence into a fixed-size vector brings some limitations, especially when dealing with long source sequences.

Attention-based models (Sordoni, Bengio, et al., 2015; Vaswani et al., 2017a) are another approach that is proposed to reduce this limitation. Using the attention algorithm, the model is able to condition just those parts of the input that are relevant to predicting the next word. Attention models and variants have contributed to significant progress in state-of-the-art in machine translation. In a dialogue system, also attention models are used to avoid word repetitions in generated responses (Mei, Bansal, & Walter, 2017). A transformer-based architecture like Open-AI GPT, GPT-2, and GPT-3 (Brown et al., 2020; Radford et al., 2018, 2019), which uses a multi-layer self-attentive mechanism to allow fully-connected cross-attention to the full context in a computationally efficient manner, which seems like a natural choice for exploring a more general solution. Transformer models allow long-term dependency information to be better preserved across time, thereby improving content consistency (Radford et al., 2019). They also have higher model capacity due to their deep structure and are more effective in leveraging large-scale datasets than RNN-based approaches (Vaswani et al., 2017a).

2.4 Natural Language Generation: From Pretrained Language Models to Reinforcement Learning

The concept of PLMs has gained popularity in recent years. Initially, the models are trained with large unlabeled corpora, and then they are fine-tuned in order to produce state-of-the-art results. Many researchers have demonstrated that these models are able to encode significant amounts of linguistic knowledge from corpora and produce universal representations of language. Hence, the use of PLMs is thus generally advantageous for downstream tasks and can prevent the need to train a model from scratch [Brown et al., 2020]. In recent years, all PLMs have been built on Transformer due to the significant achievements that Transformer has made. The GPT (Radford et al., 2018) and BERT (Devlin, Chang, Lee, & Toutanova, 2019a) are two earlier PLMs that were developed based on Transformer-decoder and Transformer-encoder, respectively. A number of PLMs are introduced following the GPT and BERT standards, including GPT-2 (Radford et al., 2019), RoBERTa (Y. Liu et al., 2019), XLNet (Yang et al., 2019), T5 (Raffel et al., 2019), and BART (Lewis et al., 2020). Research has consistently shown that the performance of PLMs can be improved just by increasing the scale of the parameters, which further led to the development of large-scale PLMs such as GPT-3 (Brown et al., 2020), Megatron (Shoeybi et al., 2019) and Switch-Transformers (Fedus, Zoph, & Shazeer, 2021) that contain billions of or trillions of parameters.

Existing approaches to dialogue generation based on Transformers fall into two categories: Transformer encoder-decoder and Transformer-decoder. Transformer encoder-decoder is an encoder-decoder architecture as introduced in (Vaswani et al., 2017a). In this architecture, dialogue history is first encoded, and then the output from the encoder is fed into the decoder to produce a response. The encoder uses bi-directional attention, while the decoder uses left-to-right attention. This architecture stacks the encoder outputs with the decoder outputs, which makes it difficult to fine-tune the encoder's parameters. A recent study (P. J. Liu et al., 2018) revealed that an explicit encoder in the Transformer encoder-decoder architecture might be redundant in that the encoding step might be incorporated directly into the decoder, providing more direct control over parameters. The Transformer-decoder architecture uses only a decoder, in which the dialogue history is encoded by left-to-right attention. The GPT PLM family belongs to this architecture. Recently, several studies have been conducted to design a dialogue system utilizing Transformer architectures (Roller et al., 2021; Wolf, Sanh, Chaumond, & Delangue, 2019; Y. Zhang et al., 2020). A number of studies propose training LMs using large-scale dialogue datasets before they are fine-tuned for specific dialogue generation tasks. For instance, in (Y. Zhang et al., 2020), authors train Transformer on more than 147 million Reddit posts, and the authors in (Shuster et al., 2020) train Transformer on large-scale Reddit data and then jointly train on 12 dialogue subtasks, and the authors in (Bao, He, Wang, & Wu, 2020) train Transformer and conducted experiments on three different kinds of conversation tasks: chit-chat, knowledge grounded conversation, and conversational question answering.

Earlier research on utilizing RL for conversational AI was focused on designing goal-oriented dialogue systems, considering dialogues as sequential decision-making processes based on MDPs (Henderson, Lemon, & Georgila, 2008; Paek, 2006; Ranzato, Chopra, Auli, & Zaremba, 2015; Singh, Kearns, Litman, & Walker, 1999; Williams & Young, 2007; Young, Gašić, Thomson, & Williams, 2013). The use of RL for improving dialogue systems has primarily focused on task-oriented dialogue systems with a limited set of actions, and less frequently, RL is applied to the generation of open-domain dialogues. Among the early attempts, the authors in (Li, Monroe, et al., 2016) applied deep RL to the full vocabulary size action space. In their approach, the dialogue system was trained with data generated by conversing with two computer agents in a simulator. The author used a combination of three reward functions to alleviate the problems of the supervised seq2seq model. The hierarchical RL approach is also used in various works to generate dialogues (Peng et al., 2017; Saleh, Jaques, Ghandeharioun, Shen, & Picard, 2020; Tang et al., 2018; J. Zhang, Zhao, & Yu, 2018). This type of model utilizes the hierarchical structure of language, which decomposes input into utterances at one level and words at another.

2.5 Toxic Language Identification and Mitigation

Open-domain dialogue systems must be able to discuss various subjects. To achieve this goal, these models are trained on large amounts of data without the need for semantic annotations. As discussed earlier, state-of-the-art dialogue systems often use PLMs optimized to generate responses within dialogue contexts such as GPT (Brown et al., 2020; Radford et al., 2018). General coverage for these conversational models is derived from unsupervised training on publicly available datasets such as Twitter or Reddit conversations. In addition, they may then be fine-tuned on smaller, curated datasets designed to teach specific conversational skills for the models (Roller et al., 2021). Unsupervised training, however, has one disadvantage: studies based on large datasets have shown that neural models reproduce stereotypes and toxic associations in the data (Davidson et al., 2019; Gehman et al., 2020). Additionally, control over the generation of responses for open-domain systems is challenging. Hence, in such a scenario, an inappropriate piece of content may be generated, or offensive responses may be made to offensive content (Curry & Rieser, 2018; Dinan et al., 2022).

An open-domain dialogue system can exhibit toxic behavior for a variety of reasons. Researchers have suggested that some of these behaviors might be due to ingesting unfathomable amounts of training data. However, focusing on the data alone would be overly simplistic. There are numerous factors that can enhance the bias of the data, such as the choice of the model and lack of control (Khalifa, ElSahar, & Dymetman, 2021). Aside from data and models, evaluations and objective functions should also be taken into account when developing a dialogue system.

Recently, several studies have been conducted to identify toxicity in textual content and a number of machine learning models have been introduced to classify toxic content. The aim of machine learning models in toxicity detection tasks is to identify toxic content that directly targets specific individuals or groups, particularly people belonging to protected categories. Bias in these models may reduce the accuracy and indicate that the models discriminate against the same groups they are designed to protect. In (Thomas, Dana, Michael, & Ingmar, 2017), the authors investigated racial bias in various toxic language detection datasets presented recently in the literature collected from Twitter (Founta et al., 2018; Golbeck et al., 2017; Thomas et al., 2017; Waseem, 2016; Waseem & Hovy, 2016). The authors trained classifiers employing each dataset and evaluated how each classifier performed on tweets written in African-American English (AAE) versus Standard American English (SAE). They found evidence of systematic racial biases across all classifiers, with AAE tweets predicted as belonging to negative classes like hate speech or harassment significantly more frequently than SAE tweets. Their results demonstrated consistent, systematic, and substantial racial biases in classifiers, and in almost every case, Black-aligned tweets are classified as sexism, hate speech, harassment, and abuse at higher rates than white-aligned tweets. The biases presented in the datasets originated from various sources. Some biases emerge from the process of data collection. The individual annotators also have their own biases, which reflect societal biases, and these biases can aggregate into systematic biases in training data. Furthermore, the variation in class membership rates across classifiers and datasets is another reason for biases. The low proportions may indicate the dominance of false-negatives due to a lack of training data, and the high proportions may signal too many false-positives, resulting in the over-sampling of abusive language in labeled datasets.

The social context in toxic language detection plays an essential role and what is considered toxic essentially depends on that context. For instance, phrases that are considered non-toxic in the African American English dialect (AAE) are labeled more toxic than general American English equivalents by a toxicity detection tool. In (Sap, Card, Gabriel, Choi, & Smith, 2019), the authors investigate the risk of racial bias in hate speech detection by considering the insensitivity of annotators to diversity in dialect and demonstrate how this insensitivity leads to racial bias in automatic hate speech detection models. The authors investigate racial bias against a speech by African Americans, focusing on Twitter. The Twitter accounts typically do not contain self-reported race information; hence the authors consider the AAE dialect a surrogate for racial identity. For this purpose, the topic model introduced by (Blodgett, Green, & O'Connor, 2016), which is trained on 60M geolocated tweets and relies on US census race/ethnicity data as topics, is utilized. This topic model predicts the probabilities of a tweet being AAE or White-aligned English. According to their results, the annotators believe that providing dialect and race makes them significantly less likely to label an AAE tweet as offensive to anyone or themselves. One major limitation discussed by the authors is the skewed demographics of the annotator's pool, where 75% of annotators self-reported White. These experiments demonstrate that providing dialect and race are two ways to mitigate annotator bias, and it significantly reduces the probability of AAE tweets being labeled as offensive.

The authors in (Xia, Field, & Tsvetkov, 2020b) introduced an adversarial approach to reduce the risk of racial bias in hate speech classifiers. This study focused on mitigating the bias related to African American English (AAE) introduced by (Sap et al., 2019). Most datasets currently used to train toxic language classifiers were collected through crowdsourced non-expert annotations. The authors in (Waseem, 2016) demonstrated that these non-experts are more likely to label text as toxic than expert annotators. In addition, the authors in (Sap et al., 2019) revealed that a lack of social context in annotation tasks increases the risk of bias in annotators. According to (Sap et al., 2019), two related issues are valid in the toxic language detection task. First, biases in annotations; second, machine learning models learn to absorb and amplify biases from false correlations existing in datasets. The authors in (Xia et al., 2020b) emphasized two main challenges related to the annotation task: first, re-annotating these datasets are time-consuming and expensive, and second, even with perfect annotations, current hate speech detection models may still learn and amplify false correlations between AAE and abusive language. Therefore, the authors proposed an adversarial approach for mitigating the risk of racial bias in hate speech classifiers, even if training data might contain biases in annotations.

Recent advancements in transformer-based LMs trained on a massive amount of web text have led to significant progress on many natural language generation tasks. The main factor behind these advances is large-scale training corpora collected from web text sources (Ritter et al., 2011; Serban et al., 2018). These texts are scraped from the web and inevitably contain toxic content. Training LMs on such data inevitably results in the generation of toxic content (Gehman et al., 2020; Sheng, Chang, Natarajan, & Peng, 2019; Wallace, Feng, Kandpal, Gardner, & Singh, 2019). The toxicity of LM has been addressed by some previous studies, either by fine-tuning a pretrained model (Gehman et al., 2020; Gururangan et al., 2020), steering a model's generation towards text less likely to be classified as toxic (Dathathri et al., 2020; Krause et al., 2021), or through direct test-time filtering (Xu et al., 2021). Prior studies have shown that direct generation toward nontoxic texts is the most promising approach to LM detoxification (Dathathri et al., 2020; Krause et al., 2021). These methods rely on an external toxicity classifier based on machine learning techniques trained on toxic language detection datasets. Although some studies address the toxicity in LMs and propose approaches to detoxifying these models (Gehman et al., 2020; Krause et al., 2021; A. Liu et al., 2021), limited research has been conducted on the impact of detoxification methods on biases associated with social identities in these models. In particular, when conditioned on prompts containing specific social identities (e.g., Asian, Hispanic, or Black), detoxified models generate text with disproportionately large amounts of toxicity. To add to this, strengthening these detoxification approaches increases the bias against ethnic minorities (Welbl et al., 2021; Xu et al., 2021). In light of the importance of LMs in numerous NLG tasks, it is imperative to identify and quantify any negative effects of detoxification methods on social biases as well as a method to mitigate these effects from propagating as inequitable results and unpleasant experiences to downstream users of these systems.

2.6 Conclusion

In this chapter, we discussed existing approaches for training open-domain dialogue generation systems and utilizing reinforcement learning to fine-tune these models. Furthermore, we examined the problem of identifying toxic language and addressing it in language models and open-domain dialogue generation systems. We indicate that the unsupervised method for training language models reproduces stereotypes and toxic associations presented in the training data. Moreover, we reviewed some recent works conducted to address toxic language identification and discussed their contributions. We will present our contribution to the design of an open-domain dialogue generation system in the next chapter.

Chapter 3

Transformer-Decoder based Reinforcement Learning Approach for Conversational Response Generation

A variety of approaches for the development of open-domain dialogue systems have been developed. However, the sequence-to-sequence (seq2seq) framework in conjunction with recurrent neural networks (RNNs) that is trained based on maximum likelihood estimation (MLE) objectives has brought about promising outcomes (Karpathy & Fei-Fei, 2015; Luong, Pham, & Manning, 2015; Luong, Sutskever, Le, Vinyals, & Zaremba, 2015). Despite the success of these methods in modeling dialogue systems, some limitations have been identified in previous works. The first limitation pointed out in several studies is that the Seq2Seq-RNN models fail to capture long-term temporal dependencies across conversation turns. The gradient vanishing problem limits the ability of Seq2Seq-RNN models to capture long-term temporal dependencies across conversation turns. The second limitation is exposure bias in these models. The most popular method used method to train the standard Seq2Seq-RNN models is the teacher-forcing algorithm (Ranzato et al., 2015). During training in this method, the decoder uses two inputs to generate the next word, the previous output state from RNN and the ground-truth word. During test time, however, the decoder only uses the words it generated at a previous time step to predict a new word since the ground-truth data is no longer available. This discrepancy is referred to as exposure bias and limits the informativeness of the generated responses since the decoding error compounds rapidly during inference (Ranzato et al., 2015). The third limitation observed in these models is a training objective for these models. Most of the existing dialogue models learn the conditional distribution of the response given the context from the MLE objective (Serban et al., 2016; Sutskever et al., 2014; Vinyals & Le, 2015). Usually, human dialogue data is redundant. Training a Seq2Seq-RNN model on these datasets with the MLE objective provides a simple mapping between the context and response, which yields generic and dull responses. Training the dialogue model with the MLE objective and ground-truth dialogue data is categorized as a supervised learning approach. One of the limitations of this approach is the lack of relatedness between training data and online scenarios. This limitation makes it difficult to optimize the dialogue systems toward its goals, generating diverse and informative responses and reducing blandness. Additionally, in supervised methods, the objective is to optimize for an instant reward rather than a long-term reward, making the dialogue system bland and failing to encourage long-term engagement with the user.

In this chapter, we propose the Reinforced Transformer-decoder (RTD) model, the combination of the Transformer-decoder architecture and Proximal Policy Optimization (PPO) method from a reinforcement learning algorithm for multiturn dialogue modeling. The RTD is formulated as an autoregressive language model and uses a multi-layer transformer-decoder as the model architecture. Recent advances in large-scale Transformer-based architectures (Devlin, Chang, Lee, & Toutanova, 2019b; Radford et al., 2018, 2019) have had significant success analyzing different natural language understanding tasks, including question answering, named entity recognition, sentence classification, and sentence similarity. One of the key points in the success of these models is their ability to capture long-term temporal dependencies in the input context. This ability also makes them a viable candidate to model multi-turn dialogue systems. Transformer architecture in the RTD model allows us to capture long-term temporal dependencies in the context of dialogue data better than RNN-based models; however, the original transformer models were trained based on MLE objective and still suffer from some of its limitations like short answer generation. We incorporate reinforcement learning training for transformers to yield longer and more informative answers to alleviate this limitation. In order to stabilize the training of the transformer-decoder in our proposed model, we employ proximal policy optimization (PPO) techniques that constrain the policy and facilitate its stability. We have empirically evaluated our approach's effectiveness on the Reddit social media dataset. The experimental results show that sentences generated by our proposed RTD model are diverse and contain information specific to the source prompt.

3.1 Related Works

The main idea behind the earliest conversation models is inspired by statistical and neural machine translation (Kalchbrenner & Blunsom, 2013; Och & Ney, 2004; Simard et al., 2007). One of the first attempts at casting conversational models as a machine translation problem was introduced by (Ritter et al., 2011), which applied a phrase-based translation method to extracted dialogues from Twitter dataset (Serban et al., 2018). The data representation in these works is in the form of (query, response). This representation creates a significant limitation for generating contextually appropriate responses. Also, the dialogue generated by these approaches is usually short and not informative. To tackle the above limitations, the RNN-based approaches for answer generation proposed in (Shang et al., 2015; Sordoni, Galley, et al., 2015; Vinyals & Le, 2015) that generate longer answers in dialogue systems. Long-Short-Term-Memory (LSTM) (Hochreiter & Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2014) are the two most favorite extensions of RNN that are used in modeling the dialogue systems (Li, Galley, et al., 2016; Vinyals & Le, 2015). The LSTM/GRU models have been shown to be effective in encoding textual data; however, they have the limitation of dealing with long context (usually more than 500 words (Khandelwal et al., 2018)). To address this limitation and exploit the longer-term context, hierarchical models are proposed in (Serban et al., 2016; Sordoni, Bengio, et al., 2015; Xing et al., 2018). Among these methods, one of the popular models is the Hierarchical Recurrent Encoder-Decoder (HRED), which was proposed by (Sordoni, Bengio, et al., 2015). In the HRED model, a two-level hierarchy that combines two RNNs is used, one for a word level and one for the dialogue utterance level. This architecture helps to reduce the vanishing gradient problem, a problem that limits RNNs' ability to model very long word sequences. Despite the success of LSTM/GRU models in language generation tasks, their encoding of the entire source sequence into a fixed-size vector brings some limitations, especially when dealing with long source sequences. Attention-based models (Sordoni, Bengio, et al., 2015; Vaswani et al., 2017a) is another approach that is proposed to reduce this limitation. The attention algorithm allows the model to condition only parts of the input context relevant to predicting the next word. Attention models and variants have contributed to significant progress in state-of-the-art machine translation. In a dialogue system, also attention models are used to avoid word repetitions in generated responses (Mei et al., 2017; Yao, Zweig, & Peng, 2015).

A transformer-based architecture like Open-AI GPT and GPT-2 (Radford et al., 2018, 2019), which uses a multi-layer self-attentive mechanism to allow fully-connected cross-attention to the full context in a computationally efficient manner, which seems like a natural choice for exploring a more general solution. Transformer models, for example, allow long-term dependency information to be better preserved across time, thereby improving content consistency (Radford et al., 2019). They also have higher model capacity due to their deep structure and are more effective in leveraging large-scale datasets than RNN-based approaches (Vaswani et al., 2017a). To address the limitations in supervised approaches, some researchers have investigated reinforcement learning for dialogue systems (Li, Miller, Chopra, Ranzato, & Weston, 2016a, 2016b; Li, Monroe, et al., 2016). Recently, RL-based methods have been studied in both goal-oriented and open-domain dialogue systems. In an open-domain dialogue system, the user goal is not explicitly defined; hence defining appropriate metrics for evaluating success, such as reward functions, is the main challenge in such dialogue systems. The authors in (Li, Monroe, et al., 2016) have made the first attempt to use RL in open-domain dialogue systems. In this approach, the dialogue system was trained with data generated by conversing with two computer agents in a simulator. The author used a combination of three reward functions to alleviate the problems of the supervised seq2seq model.

3.2 Methodology

3.2.1 Model Architecture

We can represent the dialogue system as an alternating sequence between the user and the machine. The dialogue starts with a query from a user, and the machine responds to that query, which continues until the ”end of the dialogue” utterance appears from the user. In our proposed model, the multi-turn dialogue history is considered as a long text, and the sequence generating task is considered as a language modeling task. Let us consider an unsupervised corpus consist of L tokens as $\mathcal{W} = \{w_1, \cdots, w_L\}$. The standard language modeling task objective on corpus $\mathcal{W}$ is defined as maximizing the following likelihood:

$$L(\mathcal{W}) = \sum_{i=1}^{L} \log p_\theta \left(w_i | w_{1\cdots i-1}\right) \tag{3}$$

Where the prefix $w_{1\cdots i-1} := w_1, \cdots, w_{i-1}$, k is the size of the context window, and the conditional probability p is a generative model with parameters θ. We adopt the input representation of the unsupervised model to switch it to the supervised conversational dataset that we have for training our model. In a single turn conversation, if we define the first utterance $\mathbf{x}^i = \{x_1^i, \cdots, x_M^i\}$ as a source sequence (input sequence), with M number of tokens, and the second utterance $\mathbf{y}^i = \{y_1^i, \cdots, y_N^i\}$ as a target sequence (or a ground-truth), with N number of tokens, then the dataset $\mathcal{W}^c$ consists of $(\mathbf{x}^i, \mathbf{y}^i)$ pairs, $\mathcal{W}^c = \{(\mathbf{x}^1, \mathbf{y}^1), \cdots, (\mathbf{x}^D, \mathbf{y}^D)\}$, where for each source utterance $\mathbf{x}^i$, there is a ground-truth $\mathbf{y}^i$ and D is the number of pairs in a dataset. The conditional probability of generating target sequence given source sequence in a single turn conversation can be written as the product of a series of conditional probabilities:

$$p_\theta(\mathbf{y}^i | \mathbf{x}^i) = \prod_{j=2}^{n} p_\theta \left(y_j^i | y_{1\cdots j-1}^i, x_{1\cdots m}^i\right) \tag{4}$$

For multi-turn conversation, after generating the first response $\mathbf{y}_1$, it will concatenate with the source sequences to create a dialogue history $\mathbf{x} = \{\mathbf{x}^1, \mathbf{y}^1\}$. In next turn, to generate the response $\mathbf{y}^2$ associated for input utterance $\mathbf{x}^2$, the dialogue history $\mathbf{x} = \{\mathbf{x}^1, \mathbf{y}^1, \mathbf{x}^2\}$ is considered as a source utterance. The conversation continues until the "end of dialogue" utterance appears from the user. In a multi-turn dialogue generation task, given the dialogue history $\mathbf{x}$, the dialogue response generation task can be defined as generating a response $\hat{y}$ with G number of tokens $\hat{\mathbf{y}} = \{\hat{y}_1, \cdots, \hat{y}_G\}$ where the distribution of the generated tokens is defined as follows:

$$p_\theta \left(\hat{\mathbf{y}} | \mathbf{x}\right) = \prod_{i=2}^{g} p_\theta \left(\hat{y}_i | \hat{y}_{1\cdots i-1}, \mathbf{x}\right) \tag{5}$$

The generative model that is used in our proposed RTD model as a policy network π_θ is a multilayer Transformer-decoder based on the GPT-2 architecture where θ in the parameters of GPT-2 architecture. The GPT-2 model applies a multi-headed self-attention operation over the input tokens followed by position-wise feedforward layers to produce an output distribution over target tokens (Radford et al., 2018, 2019).

We used a 12-layer GPT-2 model with masked self-attention heads (12 attention heads) and hidden states size of 768-dimensional states with a maximum sequence length of 1024 tokens. The training objective for generating sequences in GPT-2 model is defined as maximizing the following likelihood:

$$L(\mathcal{W}^c) = \sum_{j=2}^{n} \log \pi_\theta \left(y_j^i | y_{j-1}^i, \mathbf{x} \right) \tag{6}$$

3.2.2 Sequence Generation as an RL Problem

In our proposed model, as discussed previously, the Transformer-decoder is viewed as an "agent" that interacts with an external "environment". The parameters of the model, θ, define the policy π_θ, that predicts the next word as an action at each time step. Let us define $\mathcal{S}$ as a possible infinite set of states the environment can be in, $\mathcal{A}$ is a possibly set of actions, $\hat{y}_t \in \mathcal{A}$, the agent can take in a state, and R is a reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The interaction between agent and environment is modeled as a discrete-time Markov decision process (MDP) (Puterman, 2014). The agent observes the environment's current state $s_t \in \mathcal{S}$ and takes an action $\hat{y}_t$ according to a policy $\pi_\theta\left(\hat{y}_t|s_t\right) : \mathcal{S} \times \mathcal{A} \to [0, 1]$, then the environment transitions to a next state s_{t+1} according to transition probabilities. Upon generating the last token (end of sequence token), the agent receives the reward from the environment. The goal of training is to maximize the expected reward for a trajectory τ (generated words in a sequence) $J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[R\left(\tau\right)\right]$. Note that, since in the task of dialogue generation, the trajectory is finite (the length of generated sentences are finite) then we described our policy gradient in the form of bounded-length trajectory case with the length of H as described in $\tau = (s_1, \hat{y}_1, s_2, \hat{y}_2, \ldots, s_H, \hat{y}_H)$.

The Policy Gradient (PG) (Sutton, McAllester, Singh, & Mansour, 2000b) algorithms are a family of algorithms that try to optimize the policy directly. The gradient $\nabla_\theta J(\pi_\theta)$ is computed as follows:

$$\nabla_\theta J\left(\pi_\theta\right)=\mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{H} R(\tau)\nabla_\theta \log \pi_\theta\left(\hat{y}_t|s_t\right)\right] \tag{7}$$

The vanilla policy gradient update described in (7) has no bias but high variance. In order to reduce the expose high variance, we add a baseline function $b(s_t)$ to (7) as follows:

$$\nabla_\theta J\left(\pi_\theta\right)=\mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{H} \nabla_\theta \log \pi_\theta\left(\hat{y}_t|s_t\right)\left(R(\tau)-b(s_t)\right)\right] \tag{8}$$

The baseline can be an arbitrary function as long as it does not depend on the "action"; hence the baseline does not change the expected gradient, but importantly, it can reduce the variance of the gradient estimate.

3.2.3 Choice of Baseline in Policy Gradient

The general form of policy gradient can be defined as follows:

$$\nabla_\theta J(\pi_\theta)=\mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{H} \nabla_\theta \log \pi_\theta\left(\hat{y}_t|s_t\right) A^{\pi_\theta}\right] \tag{9}$$

Where A^{π_θ} is an advantage function and could be defined as the following form:

$$A^{\pi_\theta}=R(\tau)-b\left(s_t\right) \tag{10}$$

Considering the value function $V^{\pi_\theta}\left(s_t\right)$ and Q-function $Q^{\pi_\theta}\left(s_t,\hat{y}_t\right)$, the other valid choice for advantage function can be defined as:

$$A^{\pi_\theta} \;=\; Q^{\pi_\theta}\left(s_t,\hat{y}_t\right)-V^{\pi_\theta}\left(s_t\right) \tag{11}$$

The choice of advantage function in (11) has the lowest possible variance; however, the advantage function in practice is not known and must be estimated (Schulman, Moritz, Levine, Jordan, & Abbeel, 2015). For advantage function approximation, usually, the neural network is used as a function approximator. The first challenge for using a neural network as a function approximator is that it requires a large number of samples, and also it is difficult to obtain stable and steady improvement in training the neural network despite the non-stationary of the incoming data in dialogue systems. To reduce the bias in (8) and not to deal with training the second approximator that causes insatiability, we baseline the REINFORCE algorithm with the reward obtained by the current model under the inference algorithm used at test time. In this method, the baseline is obtained by performing a greedy search over model output probability distribution at each time step. Let's define the greedy output selection $\hat{y}_t^g$ as:

$$\hat{y}_t^g = \text{argmax}_y \, \pi_\theta \left(y \mid y_{1:t-1}\right)$$

Hence, the advantage in (11) is defined as:

$$A^{\pi_\theta} = R(\hat{y}_1, \cdots, \hat{y}_H) - R(\hat{y}_1^g, \cdots, \hat{y}_H^g) \tag{12}$$

This approach avoids all the inherent training difficulties associated with actor-critic methods, where a second critic network must be trained to estimate value functions, and the actor must be trained on estimated value functions rather than actual rewards. A similar approach was used in the context of obtaining baseline with the reward obtained by the current model under the inference algorithm used at test time for image captioning (Rennie, Marcheret, Mroueh, Ross, & Goel, 2017), and to our knowledge, this is the first time that this approach incorporated for optimizing the Transformer-decoder policy network for dialogue generation task.

3.2.4 Proximal Policy Optimization

We apply Proximal Policy Optimization (PPO) (Schulman, Wolski, Dhariwal, Radford, & Klimov, 2017) to ensure to take the biggest possible improvement step on a policy without causing instability in performance. PPO is modified from Trust Region Policy Optimization (TRPO) (Schulman, Levine, Abbeel, Jordan, & Moritz, 2015) by using a clipped surrogate objective while retaining similar performance. In TRPO, the policy updates by taking the largest step possible to improve the performance while satisfying the KL-Divergence constraint that specifies how close the new and old policies are allowed to be. Since a single bad step can unstable the policy and collapse the policy performance, avoiding this kind of collapse helps to improve the process of training. The PPO only relies on clipping in the objective function to heuristically constrain the KL-divergence and limit the improvement of the new policy by not getting far from the old policy. let's define the probability ratio between old and new policies as follows:

$$r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}$$

The objective function of PPO is defined as follows:

$$\theta_{new} = \arg\max_{\theta} \mathbb{E}_{s,a\sim\pi_{\theta_{old}}} \left[L\left(s, a, \theta_{old}, \theta\right)\right] \tag{13}$$

Where L is defined as follows:

$$L\left(s, a, \theta_{old}, \theta\right) = \min\left(r(\theta)A^{\pi_{\theta_{old}}}(s, a), clip\left(\epsilon, A^{\pi_{\theta_{old}}}(s, a)\right)\right)$$

The parameter ϵ is a hyperparameter, and the function clip$(\epsilon, A^{\pi_\theta})$ is defined as follows:

$$\text{clip}\left(\epsilon, A^{\pi_\theta}\right) = \begin{cases} (1+\epsilon)A^{\pi_\theta} & A^{\pi_\theta} \geq 0 \\ (1-\epsilon)A^{\pi_\theta} & A^{\pi_\theta} < 0 \end{cases}$$

The hyperparameter ϵ determines how far away the new policy can improve from the old policy while still profiting from the objective. Our implementation of PPO for training the policy is based on (Dhariwal et al., 2017). We consider 1M episodes with four PPO epochs per batch and one minibatch each; we select $\epsilon = 0.2$ and the default value for other parameters according to (Dhariwal et al., 2017).

3.2.5 Reward Function for RL Training

One of the challenges in learning dialogue models with RL is how to define an effective reward function for training the agent in an environment. Defining the proper reward function is an expert domain challenge; i.e., it is required to know the problem definition accurately and have a vast knowledge of distinguishing different actions by an agent. One of the main limitation of the open-domain dialogue system is generating bland and uninformative responses. To address this problem, we implement a mutual information scoring function (Li, Galley, et al., 2016) as a reward function for our RL training approach. Intuitively, maximizing the mutual information help the model to avoid assigning a high reward to sequences that are ungrammatical or not coherent and allows the model to generate responses that are more specific to the source, while generic responses are largely down-weighted. We can define the mutual information reward function between two consecutive utterances X_i and X_{i+1} as follows:

$$R = (1 - \lambda) \log \pi_\theta \left(\mathbf{x}_i | \mathbf{x}_{i-1}\right) + \lambda \log \pi_\phi^{bw} \left(\mathbf{x}_{i-1} | \mathbf{x}_i\right) \tag{14}$$

Where π_θ^{bw} is a backward probability of generating the previous utterance $\mathbf{x}_{i-1}$ given an utterance $\mathbf{x}_i$ and λ is a hyperparameter. In our work we select $\lambda = 0.5$. In (14), the reward function R, employs a pretrained backward model to predict source sentences from given responses. For computing the reward, 8 hypotheses are generated for the input source sentence by the policy π_θ by using the top-K sampling method (Huang et al., 2019) (we set k=10), and then according to (14), the reward associated with each sample is calculated. Maximizing backward model likelihood penalizes the bland hypotheses, as frequent and repetitive hypotheses can be associated with many possible queries, thus yielding a lower probability for any specific query. The backward pretrained model π_θ^{bw} is trained using the same π_θ by just interchanging the source and target responses in a training dataset and conditioning the π_θ to generate the source sequence $\mathbf{x}_{i-1}$, given the target sequence $\mathbf{x}_i$.

3.2.6 Applying RL training

Algorithm 1 describes our proposed framework to train a Transformer-decoder with a reinforcement learning algorithm in detail.

Algorithm 1: Fine-tuning the generative policy with RL.

Result: Optimized policy with updated parameter θ^*

Initialized the policy π_θ with parameter θ and clipping threshold $\epsilon = 0.2$;

foreach *epoch* **do**

  foreach *batch* **do**

    Sample the policy to generate a set of sequences;

    Calculate the reward R_t^k ;

    Obtain the baseline b_t by greedy-sampling the policy;

    Compute the advantage $A^{\pi_\theta} = R(\tau) - b(s_t)$;

    Assignee the current policy to the old policy : $\theta_{old} \leftarrow \theta$

    foreach *PPO iteration* **do**

      Compute policy update:

      $\theta^* = \arg\max_\theta \mathbb{E}_{s,a\sim\pi_{\theta_{old}}} \left[L\left(s, a, \theta_{old}, \theta\right)\right]$

    end

  end

end

In the dialogue generation task, since the size of action space is equal to the size of vocabulary in a dataset; hence we deal with a large action space problem. To deal with a randomly initialized poor policy that let to slow convergence or even instability in training, we first train the generative model with cross-entropy loss using the ground truth sequences, and then we handle the transition between cross-entropy loss to RL loss. At the beginning of the training, the model completely relies on the cross-entropy loss. We pretrain the policy until the score on the development set stops improving, and then, the training completely relies on RL loss. In the next step, after pretraining, the policy is sampled to generate a set of sequences. The advantage associated with these sequences is then calculated using, and this advantage is considered for computing the policy update with the PPO algorithm. In our work, we consider four iterations in the PPO algorithm for updating the policy at each batch. The number of PPO iterations is a hyperparameter that is selected and tuned during the training.

3.3 Experiments

3.3.1 Dataset

We evaluate our proposed model on the dialogue corpus that is extracted from Reddit conversations. Reddit is a massive collection of forums where people can post social news, discuss different topics, share their ideas, and comment on other people's posts. The contents of Reddit organize their subjects into subreddits, which cover a wide range of topics, including sports, news, politics, movies, science, and social media. These subreddits are monitored by moderators and filled with quality content, and the grammatical quality of the sentences extracted from Reddit is very high. Including a wide range of topics with grammatical quality sentences make Reddit well suited for grounded open-domain conversational modeling. The extracted dataset from Reddit dialogues contains about 3M dialogues that are randomly sampled 50K dialogues as development data and 50K dialogues as test data.

3.3.2 Setup

The policy network in our model inherits from OpenAI GPT-2, 12 layers transformer-decoder with masked self-attention heads (12 attention heads) and hidden state size of 768-dimensional states. The model uses learned positional embeddings with a sequence length of 1024 tokens. We model a multiturn dialogue session as a long text and frame the generation task as language modeling. We used a bytepair encoding (BPE) (Sennrich, Haddow, & Birch, 2016) vocabulary with 50257 merges and for regularization, we used residual embedding and attention dropouts with a rate of 0.1. We used the Adam optimization with a max learning rate of 1e-5 and the learning rate was increased linearly from zero over the first 2000 updates and annealed to 0 using a cosine schedule. We first train the Transformer-decoder with the MLE objective until its score on the development dataset stops improving and then continue training with the PPO objective. We evaluate our proposed model compare with 3 baseline generative methods, which we describe below:

- **Seq2Seq**

  The first generative model that we consider as a baseline is a work presented in (Sutskever et al., 2014) that is a 4-layers LSTM encoder-decoder with 1000 cells at each layer and 1000 dimensional word embeddings.

- **RL-Seq2Seq**

  This model is proposed by (Li, Monroe, et al., 2016). In this scheme, the policy gradient method for optimizing the seq2seq policy is implemented and a manually tailored reward function is considered.

- **Transformer-Decoder with MLE**

  We also consider Transformer-decoder as a generative model with the MLE objective without training with the PPO objective. For this model, we trained the Transformer-decoder with the MLE objective, and the training will be continued until there is no improvement observed in the validation dataset.

3.3.3 Automatic Evaluation

To evaluate the quality of generated responses in our proposed dialogue generation model, we performed automatic evaluations based on relevance and diversity metrics. To evaluate diversity, we use distinct unigrams (Distinct-1)/bigrams (Distinct-2) (Li, Monroe, et al., 2016) and Entropy (Ent-n) (Y. Zhang et al., 2018) metrics in our work. Models with higher a number of distinct n-grams and Entropy tend to produce more diverse responses.

For relevance evaluation, we adopt contextualized embedding metrics. Contextualized representations from a transformer-based model like BERT (Devlin et al., 2019b) are recently shown to be beneficial in many NLP tasks. In our work, we consider three embedding-based metrics and we use BERT to have contextualized representations for each word. The three embedding metrics that we consider in our work are Average metric (Mitchell & Lapata, 2008), Greedy metric (Rus & Lintean, 2012), and Extreme metric (Forgues, Pineau, Larchevêque, & Tremblay, 2014). In the Average metric, two separated vectors achieve by taking the mean over word embeddings in model-generated response and ground-truth response, and then the cosine similarity between these two vectors computes. In the greedy metric, the responses embed by taking the maximum cosine similarity over embeddings of two utterances. The Extreme metric obtains sentence representation by taking the largest extreme values among the embedding vectors of all the words it contains, then calculates the cosine similarity of the sentence representations.

The last evaluation metric we consider is the Normalized Average Length (NAL) metric. The NAL metric measures the average number of words in model-generated responses normalized by the average number of words in the ground truth. To compute the NAL score, we consider the length of ground-truth and generated responses and compute the ratio between these two sequences.

The results obtained using diversity evaluation metrics are summarized in Table 3.1.

Table 3.1: Performance in terms of diversity metrics for response generation in Reddit dialogues dataset

Model	**Dist-1**	**Dist-2**	**Ent-4**
Seq2Seq	0.761%	1.912%	6.832
RL-Seq2Seq	1.820%	4.233%	8.112
MLE-TD	6.561%	25.426%	9.117
R-TD	11.173%	47.348%	10.876

The results in Table 3.1 demonstrate that the R-TD model achieves the highest diversity scores among other models. comparing the R-TD with MLE-TD and RL-seq2seq, we observe substantial improvements in diversity due to the use of the mutual information reward function and applying the PPO policy update from the reinforcement learning algorithm in training the Transformer-decoder.

The results obtained using relevance evaluation metrics are summarized in Table 3.2. The re-

Table 3.2: Performance in terms of relevance metrics for response generation in Reddit dialogues dataset

Model	**Average**	**Greedy**	**Extreme**	**Avg Length**
Seq2Seq	0.529	0.393	0.366	8.31
RL-Seq2Seq	0.683	0.431	0.401	11.16
MLE-TD	0.793	0.544	0.502	12.68
R-TD	0.823	0.661	0.644	16.72

sults in Table 3.2 demonstrate that the R-TD model achieves the highest scores based on relevance metrics; which means that it capable of capturing the topic of the target response than other models. In a multi-turn dialogue generation system, if the generated response deviates too much from the target topic, the response content will not be relevant to the dialogue context and it deserves a lower relevancy score; Similarly, if the generated response is highly related to the target topic, the diversity score becomes lower. This trade-off between relevance and diversity must be considered during the evaluation process. The results in Table 3.1 and Table 3.2 show that the R-TD model is capable of generating diverse responses with a high relevance score to the topic, compare with all three dialogue systems. If we compare the R-TD model with MLE-TD in Table 3.2, we observe that the average length of the generated response by R-TD is higher than MLE-TD. This indicates the MLE-TD suffers from the impact of being trained with only the maximum likelihood objective. It is important to point out that number of parameters for R-TD, RL-Seq2Seq, and MLE-TD is 124M, and for the Seq2Seq model with LSTM, the number of parameters is 42M.

3.3.4 Qualitative Evaluation

Table 3.3 and Table 3.4 demonstrate samples of the model generated dialogues. The dialogue presented in Table 3.3 is an open topic dialogue between user and bot. The questions asked by the bot are not topic-specific, and they are general questions. The questions asked from a bot that is presented in Table 3.4 are related to a specific topic (social media in this case), and the bot answer the user-specific questions about social media. The results in Table 3.3 and Table 3.4 demonstrate that our model is capable of handling different conversations with a high level of coherence in the generated responses. The responses generated by our model are related to the questions with a high level of diversity in responses. We observe that dull responses like ”I don’t know” and ”I am not sure” are not generated by the model, and instead, the system generates an alternative, reasonable answer.

3.4 Conclusion

In this chapter, we have introduced our Reinforce Transformer-Decoder (RTD) method for training an open-domain dialogue generation system with reinforcement learning based on transformer-decoder architecture. The RTD is an autoregressive Transformer-decoder model trained with PPO

Table 3.3: An interactive example of general topics conversation

Role	Response
User	hi how are you?
Bot	I'm good! Just had the worst hangover I've ever had. What about you?
User	I am good, thanks for asking, why you had hangover?
Bot	I went to a friends house and was eating too many things so I ate too much food, so I fell sick.
User	I hope you will be better soon
Bot	thank you, me too. I am a very tired person, so I am also going to stay sick.
User	do you go to the gym regularly?
Bot	yes I try to go once a week. I go every 4 days.
User	I think once a week is not enough, you must go to the gym 3 times a week!
Bot	well that was a bit harsh. maybe 2 3 times
User	Do you prefer to go to the gym in the morning or afternoon?
Bot	I usually go to the gym after 8 or 9 pm. sometimes I go to my own gym after 5.
User	I like to go to the gym at noon
Bot	that's the only time I go to gym. it's only an hour or so away.
User	what is your job?
Bot	I work as a chef
User	cool! so you know about food!
Bot	yeah, I know about stuff like sushi
User	how long do you work as a chef?
Bot	5 years I've been a chef since high school. I'm a student now.
User	so you work and study at the same time!
Bot	I work and study at the same time. That makes me hard
User	good, I wish success in your career as a chef
Bot	And I wish you success in your career as well!
User	Thanks, have a good day!
Bot	Np, see you later!

Table 3.4: An interactive example of specific topic conversation

Role	Response
User	Hi how are you?
Bot	I'm fine, how are you?
User	I am good, thanks for asking!
Bot	No problem. I am glad you are doing ok!
User	I want to talk about social media with you
Bot	Yeah, I wanna hear from you about social media. Maybe we could make it a bit like this, social media is like an internet thing but for gaming.
Uses	do you have any accounts in social media?
Bot	I used to have a twitter account, but I don't anymore.
User	why?
Bot	Because I was too lazy to change my old password
User	what do you think about the role of social media in our society?
Bot	Not much. I was very active on it but not to the level of most things. I think I would be in my prime as a journalist though
User	What type of content we can share on social media?
Bot	We're all social media, it just takes time. You can't do everything in one post like on twitter, but you have to start somewhere and be consistent
User	do you think social media will be dangerous in future?
Bot	Not a lot but it will be good, social media is pretty much the new way to get information
User	what is the risk of social media for kids?
Bot	They're the same risk as they are for adults.
User	could you name some risks of social media for kids?
Bot	Social media for kids doesn't exist.
User	How much time do you spend replying and reacting to all your Social mentions?
Bot	I usually only respond to one of my social mentions a week, and respond to others when I'm feeling down and or depressed.
User	Does your mood affect your behaviour on Social Media?
Bot	Nope, just personal. Social Media is a social game and I'm just a social player, I really enjoy my time with people.
User	Has Social media made your life better?
Bot	yeah, I think so, I've gotten over so much stress and the only problem I ever had to deal with is the one time I broke up with someone.
User	nice to chat with you! take care!
Bot	Thanks, you too, man!

algorithms from reinforcement learning algorithms for response generation tasks, facilitating informative and diverse conversations between the user and the dialogue agent. We have evaluated our model on the Reddit dialogue dataset. The results based on automatic evaluation metrics (relevance and diversity) demonstrate that the RTD model improves the proportion of high-quality responses compared to existing methods without losing the ability to generate fine-quality replies. Performance improvements are attributed to the combination of Transformer architecture and reinforcement learning training algorithm over a simple Transformer-decoder architecture that is trained based on maximum likelihood estimation. In the next chapter, we will present our contributions to the design toxic classifier model to detect toxicity and mitigate unintended bias in toxicity prediction.

Chapter 4

Domain Adaptation Multitask Deep Neural Network for Mitigating Unintended Bias in Toxic Language Detection

Identifying potential toxicity within conversations has become an essential topic for different social media platforms. Social media have rapidly evolved into viable platforms for people to share their thoughts, and vast minorities and individuals have had the opportunity to share their stories through these accessible platforms. While expressing oneself on these platforms is a human right that must be respected, inducing and spreading toxic speech towards another group is an abuse of this privilege. Toxicity in conversations is defined as textual comments with threats, insults, obscene, rude, or disrespectful racism. Increasing the exchange of ideas has affected the spreading of toxic content, including racism, sexual harassment, and other negative behaviors that are not tolerated in society. Online harassment has been one of the main criticisms against social media giants like Facebook and Twitter, who have come under increased pressure to address this misuse.

Human surveillance and automated filtering are two main strategies used by social media platforms to moderate and stop the spread of toxic comments. Considering the increasing amount of content generated by users online, it is becoming increasingly difficult to scale up the human moderation of such content. Hence, automated detection of toxic language in online content has become an imperative research area and led many social media giants to seek machine learning-based solutions to supplement the current human moderator systems. While automated classification models are unlikely to replace human moderators, they can simplify their task by suggesting which content prioritizes moderation.

In the last few years, there have been several studies on applying machine learning methods to detect toxic language in online content (Burnap & Williams, 2015; Davidson et al., 2019; Kumar et al., 2018). With the recent growth in the use of machine learning methods for the toxic language detection task, several researchers have identified that these classifiers have been shown to capture and replicate biases common in society (Borkan et al., 2019; Wiegand et al., 2019). A specific problem in these classifiers is their sensitivity to frequently attacked identity groups such as gay, Muslim, Jewish, and black, which are only toxic comments when combined with the proper context. Since these machine learning models are built from human-generated data, human biases can easily result in a skewed distribution in the training data. These models give unreasonably high toxicity scores for non-toxic statements containing specific identity terms. The source of this bias was the unbalanced representation of identity terms in a training dataset: words like ”Black” or ”Asian” were often used in toxic comments; hence the models are over-generalized and learn to associate those terms with the toxicity label unfairly (Borkan et al., 2019; Dixon et al., 2018; Park et al., 2018). The risk of such systems containing unintended biases is crucial since it could negatively impact the same social groups that the system is designed to protect. If the model incorrectly identifies non-toxic comments by a minority group as toxic, the victim may be unfairly penalized. Furthermore, if the model failed to identify abuse against the targeted minority group, we would not be able to take action against the toxic content. Although no model can entirely avoid such problems, the potential for such models to be systematically biased against certain social groups, particularly protected classes, must be concerned.

In this chapter, our primary focus is to explore and propose a method for mitigating unintended model bias in toxic language detection tasks. We propose a multitask deep neural network (MTDNN) framework based on a domain adaptation language model that detects and identifies toxic language within online conversations. Multitask learning has been extensively studied and employed in different machine learning applications, such as natural language understanding and computer vision (Collobert & Weston, 2008; Deng, Hinton, & Kingsbury, 2013; Ramsundar et al., 2015). Recent studies demonstrate that multitask learning can improve performance on various natural language understanding tasks while revealing novel insights about language modeling (X. Liu, He, Chen, & Gao, 2019; Suresh, Gong, & Guttag, 2018). Furthermore, We consider a large pretrained transformer model introduced by (Devlin et al., 2019a) for our MTDNN as the language model, and we continue its pretraining on our dataset to have a domain-specific language model that is tuned for the toxic language detection task. To evaluate our proposed approach on real data, we use the ”Unintended Bias in Toxicity Classification” dataset published by the Google Jigsaw team (Google, 2019), which contains 1,804,874 comments from the Civil Comments platform. Google Conversation AI Team extended annotation for this dataset by human raters for different toxic conversational attributes.

Moreover, in this chapter, we investigate the effectiveness of our MTDNN approach for mitigating unintended bias in out-of-domain downstream tasks. The introduction of Transformers (Vaswani et al., 2017b), has enabled fine-tuning of pretrained transformer-based language models, including BERT (Devlin et al., 2019a) to become standard practice across various NLP tasks, including toxic language detection. Recent studies revealed that fine-tuning the language models for downstream NLP tasks may exhibit biases against protected identities such as gender or ethnic minorities (Kennedy, Jin, Mostafazadeh Davani, Dehghani, & Ren, 2020; Kurita, Vyas, Pareek, Black, & Tsvetkov, 2019) as models may learn to misassociate specific features with toxic or non-toxic labels or amplify biases encoded in pretrained language models to downstream models (Dixon et al., 2018). Approaches such as data augmentation (Dixon et al., 2018; Park et al., 2018; Zhao, Wang, Yatskar, Ordonez, & Chang, 2018), which balanced the data between toxic and non-toxic labels, and adversarial learning (Kumar et al., 2018; Xia, Field, & Tsvetkov, 2020a) which focused on mitigating bias in representation by training a toxic classifier with an adversarial predictor to predict sensitive attributes, have been introduced in the literature to reduce biases in toxic classifiers. However, these techniques have been designed to mitigate the biases specific to the given dataset or domain. Therefore, they require a revised bias mitigation approach when considering a new downstream dataset.

4.1 Related Works

Research in the field of safety and security in social media has grown substantially in the last few years. A particularly relevant aspect of this research is how offensive language is detected in social networks. Previous studies have looked into various aspects of offensive languages, such as the use of abusive language (Nobata, Tetreault, Thomas, Mehdad, & Chang, 2016), aggression (Kumar et al., 2018), bullying (Dadvar, Trieschnigg, Ordelman, & de Jong, 2013), hate speech and toxic language (Borkan et al., 2019; Burnap & Williams, 2015; Davidson et al., 2019; Malmasi & Zampieri, 2017; Zampieri et al., 2019). To this end, various datasets have been created to benchmark progress in the field. Recently, (Thomas et al., 2017) compiled and released a dataset of over 24,000 tweets labeled as containing hate speech, offensive language, or neither for the hate speech detection task. Furthermore, for online toxic language detection, (Dixon et al., 2018; Google, 2019; Wulczyn et al., 2017) introduced the Wikipedia Toxic Comments dataset that was collected and extracted from Wikipedia Talk pages and featured in a Kaggle competition.

Recent studies introduced different machine learning methods for the toxic language detection task (Davidson et al., 2019; Dixon et al., 2018; Mishra, Tredici, Yannakoudakis, & Shutova, 2019; Wulczyn et al., 2017). The best performing systems introduced in these studies used deep learning approaches such as LSTMs, CNNs, and Transformers (Dinan, Humeau, Chintagunta, & Weston, 2019; Kumar et al., 2018). While toxic speech and abusive language detection have become an increasingly significant area for natural language processing research, there has been little work addressing the presence of unintended bias in these systems. Machine learning models for toxic language detection have been shown to obtain and replicate biases against specific names of frequently attacked identity social groups such as gay, lesbian, Muslim, Jewish, and black (Menon & Williamson, 2018; Park et al., 2018; Vaidya, Mai, & Ning, 2020). As a result, addressing the possibility of such biases against specific groups in toxic language detection models is critical and should be taken seriously. The unintended biases related to race, gender, and sexuality that yield high false-positive rates are investigated in recent studies (Burnap & Williams, 2015; Thomas et al., 2017). Furthermore, (Waseem, 2016) studied the correlation between annotation schemes, the annotators' identity, and reducing the effect of bias in machine learning models. A recent work (Dixon et al., 2018) investigated biases in the "Google Perspective API" classifier and revealed that it tended to give high toxicity scores to non-toxic comments that include certain social groups such as gay. They observed that several such "social identity terms" are disproportionately represented in the dataset labeled as toxic, and this false-positive bias is caused by the model over-generalizing from the training data.

Several recent works introduced metrics to quantify the presence of these unintended biases according to specific definitions (Friedler et al., 2019; Kleinberg, Mullainathan, & Raghavan, 2016; Menon & Williamson, 2018). Moreover, the importance of these metrics in evaluating the machine learning models is demonstrated in (Brennan, Dieterich, & Ehret, 2009; Buolamwini & Gebru, 2018). Among these works, Google conversation AI Team (Borkan et al., 2019) proposed metrics that are threshold agnostic, robust to class imbalances in the dataset, and provide more nuanced insight into the types of unintended bias present in the model. In our work, we used these particular evaluation metrics to evaluate the quality of our proposed approach to mitigating the model bias.

Previous studies have looked into various aspects of toxic languages, such as abusive language and hate speech (Borkan et al., 2019; Burnap & Williams, 2015; Malmasi & Zampieri, 2017; Zampieri et al., 2019). To this end, various datasets have been introduced to benchmark progress in this field. For instance, for online toxic language detection, (Dixon et al., 2018; Google, 2019; Wulczyn et al., 2017) introduced the Wikipedia Toxic Comments dataset, which includes over 100K comments that were collected and extracted from Wikipedia Talk pages and featured in a Kaggle competition. Most datasets currently used to train toxic language classifiers were collected through crowdsourced non-expert annotations. The authors in (Waseem, 2016) reveal that these non-experts are more likely to label text as toxic than expert annotators, and (Sap et al., 2019) reveal that a lack of social context in annotation tasks increases the risk of annotators' bias. According to (Sap et al., 2019), two related issues are valid in the toxic language detection task. First, biases in annotations; second, machine learning models learn to absorb and amplify biases from false correlations existing in datasets.

As the popularity of toxic language detection systems has grown, several biases have been found in classifiers and datasets, pushing several debiasing efforts to mitigate these unintended biases such as gender and racial bias (Davidson et al., 2019; Park et al., 2018; Sap et al., 2019). Former studies on bias in hate speech datasets have mainly focused on detecting and reducing bias against specific identities such as Black, atheist, or gay (Dixon et al., 2018; Park et al., 2018). In (B. H. Zhang, Lemoine, & Mitchell, 2018), the authors focused on mitigating bias in representation by proposing training a toxic classifier with an adversarial predictor to predict sensitive attributes. Furthermore, some studies address bias in pretrained word vectors or language models (Liang et al., 2020; May, Wang, Bordia, Bowman, & Rudinger, 2019; Zhou et al., 2019); however, they did not study the effect of such biases on downstream classifiers. Bias mitigation methods in the literature are usually applied during fine-tuning to address bias in a specific downstream task or dataset (Park et al., 2018; B. H. Zhang et al., 2018). For instance, data augmentation approaches (Dixon et al., 2018; Park et al., 2018; Zhao et al., 2018) and adversarial learning approaches (Kumar et al., 2018; Zhou et al., 2019) produce debiased data representations that act on biases particular to the given dataset, domain, or tasks.

In this chapter, we propose our novel approach- a two-step training multitask deep neural network (MTDNN) framework based on a domain adaptation language model- to detect toxic language and mitigate unintended model bias in the toxic language detection task. Multitask learning has been extensively studied and employed in different machine learning applications, such as natural language understanding and computer vision (Collobert & Weston, 2008; Deng et al., 2013; Ramsundar et al., 2015). We considered a large pretrained language model for our MTDNN and continued its pretraining to have a domain-specific language model tuned for the toxic language detection purpose. The structure of our multitask learning is influenced by Transformers-based multitask learning frameworks introduced by (X. Liu et al., 2019). In (X. Liu et al., 2019), the author introduced a multitask deep neural network for learning representations across multiple natural language understanding tasks and demonstrates that multitask learning leads to creating more general representations to help adapt to various tasks and domains.

4.2 Source of biases

Machine learning models in toxicity detection aim to identify toxic language that directly targets specific individuals or people belonging to protected communities. However, bias in these models may reduce the accuracy and indicate that the models discriminate against the same groups they are designed to protect. Recently, (Davidson et al., 2019) investigate racial bias in various toxic language detection datasets collected from Twitter and found evidence of systematic racial biases across all classifiers. In these classifiers, the tweets belonging to the African American English (AAE) community are predicted as ”toxic” more frequently than the tweets that belong to Standard American English (SAE).

The biases presented in toxic language datasets originated from various sources. Some biases emerge from the process of data collection. The individual annotators also have their own biases, which reflect societal biases, and these biases can aggregate into systematic biases in training data. Furthermore, the variation in class membership rates across classifiers and datasets is another reason for biases. The low proportions may indicate the dominance of false-negatives due to a lack of training data, and the high proportions may signal too many false-positives, resulting in the oversampling of abusive language in labeled datasets. It is crucial to consider how contextual factors interact with linguistic nuances and toxicity definitions. Different communities have different language norms, such that a model suitable for one community may discriminate against another.

There are two main challenges related to the annotation task for toxic language datasets. First, re-annotating these datasets are time-consuming and expensive, and second, even with perfect annotations, current toxic speech detection models may still learn and amplify false correlations between several identities belonging to marginalized communities and toxic language. Consequently, there is a significant need for developing mitigation methods to reduce unintended biases present in classifier output regardless of the presence of annotation bias in the training data.

4.3 Dataset

In this work, for unintended bias evaluation on real data, we use the ”Unintended Bias in Toxicity Classification” dataset published by Google Jigsaw (Google, 2019). This dataset contains

1,804,874 comments from the Civil Comments platform made available at the end of 2017 to understand and improve online conversations. Google Conversation AI Team extended annotation for this dataset by human raters for different toxic conversational attributes. This dataset includes individual comments that are used to detect toxicity. Each comment in the dataset has a toxicity label with fractional values (between 0 and 1), representing the fraction of human raters who believed the attribute applied to the given comment. The comment with a label greater or equal to 0.5 will be considered the toxic class; otherwise, it is considered a non-toxic class. Table 4.1 shows a few examples of comments and their associated toxicity and identity labels and Figure 4.1 illustrates the distribution of various identities in the datasets.

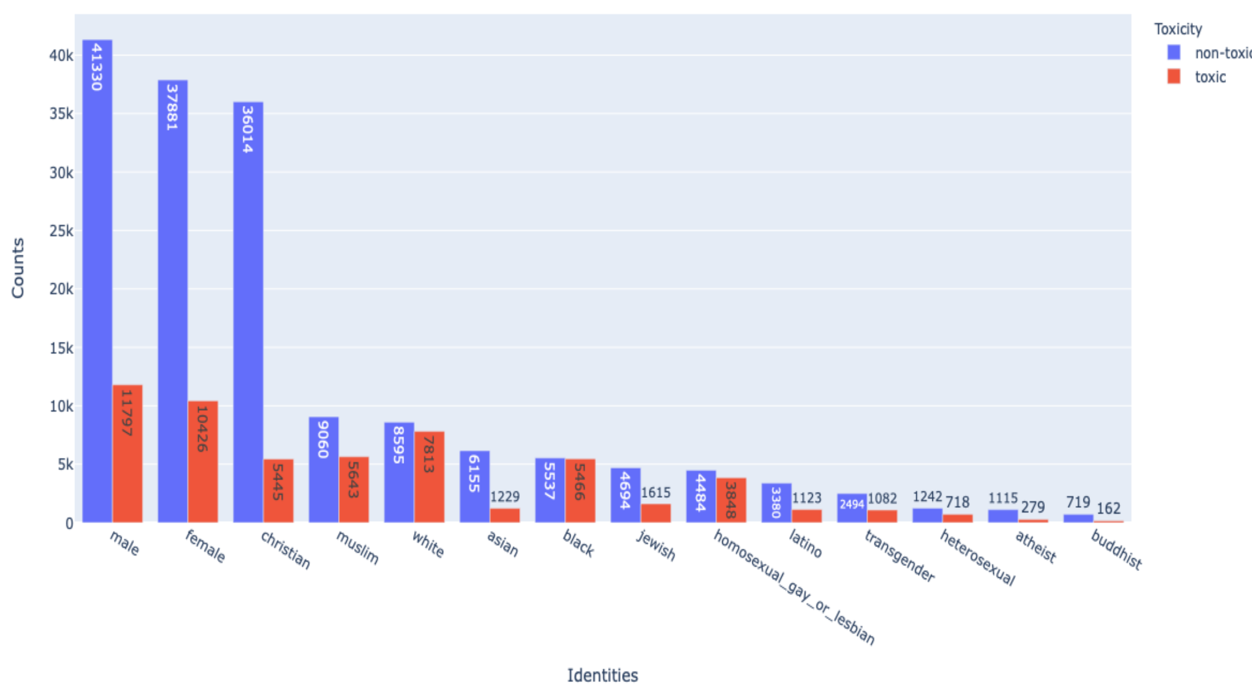

Figure 4.1: Distribution of identities in the Jigsaw Toxicity dataset (Google, 2019).

The total number of toxic comments in this dataset is 144,334, which is 8% of all the comments are toxic comments. While all of the comments were labeled for toxicity, a subset of the dataset that includes 405,130 comments, has also been labeled with various identity attributes (non-exclusive), representing the presence of identities in the comments. Table 4.2 demonstrates all these identities with the number of toxic and non-toxic comments related to each one.

Table 4.1: A few examples of comments and their associated toxicity and identity labels.

Comment	Toxicity	Identity labels
Continue to stand strong LGBT community. Yes, indeed, you'll overcome and you have.	Non-toxic	homosexual-gay-or-lesbian: 0.8 bisexual: 0.6 transgender: 0.3 all other identities: 0
I'm a white woman in my late 60's and believe me, they are not too crazy about me either!!	Non-toxic	female: 1 white: 1 all other identities: 0
Why would you assume that the nurses in this story were women?	Non-toxic	female: 0.8 all other identities: 0

Table 4.2: Identities presented in the dataset with the number of toxic and non-toxic comments by each identity.

Identity Group	Identity attributes	Non-Toxic	Toxic
Gender	Female	63264	10426
	Male	68382	11797
	Transgender	5038	1082
	Other gender	2296	427
Religion	Christian	55915	5445
	Jewish	9290	1615
	Muslim	21007	5643
	Hindu	1361	196
	Buddhist	1204	162
	Atheist	1974	279
	Other religion	14710	2022
Race or Ethnicity	Asian	9746	1229
	Black	14097	5466
	White	22135	7813
	Latino	5813	1123
	Other race or ethnicity	16169	2698
Sexual Orientation	Heterosexual	2735	718
	Homosexual-gay-or-lesbian	11459	3848
	Bisexual	2800	530
	Other sexual orientation	3697	811
Disability	Physical disability	2779	448
	Intellectual or learning disability	1823	825
	Psychiatric disability or mental illness	8253	2412
	Other disability	3088	457

In our work, for training the MTDNN model, we create six tasks from the dataset. The first task, which is also the main task in our work, is toxic comment detection, which has two labels: toxic and non-toxic. Since all the comments were labeled for toxicity, we consider all the data for this task. The goal of the first task is to detect whether the comment is toxic or non-toxic. Furthermore, we want to reduce the model bias towards the specific social identities in non-toxic comments. For this purpose, we create five more tasks to help the model reduce identity bias in the toxicity prediction task. Since the dataset includes five identity groups in which each group consists of different identity attributes (Table 4.2), we create a task for each identity group to predict the identity attributes related to its identity group. For this purpose, we only considered the comments that were labeled for subgroup identities; hence the size of data for each task varies. It is important to note that these five tasks are multi-label text classification tasks, and more than one label may assign to a single comment in each task. Table 4.3 demonstrates these five tasks with the number of toxic and non-toxic comments in each task. These six tasks are considered for training the MTDNN

Table 4.3: Distribution of data for each toxic identification task.

Task	Number of labels	Non-Toxic	Toxic
Gender identification	4	106526	18060
Religion identification	7	80145	11340
Race or Ethnicity identification	5	51555	13199
Sexual Orientation	4	15890	4644
Disability	4	13243	3582

model. We will discuss the training in detail in the next section.

4.4 Methodology

In this section, we propose a toxic language detection model based on the domain adaptation language model and MTDNN. The jointly learning toxicity and identity information benefit the model to improve the accuracy of toxic comments detection and mitigates the model bias toward commonly attacked identities in online conversations. The training procedure of our proposed model consists of two stages: Domain adaptation masked language model pre-training and multitask learning, discussed in detail in the following section.

4.4.1 Language Model Domain Adaptation

The first step in the training of our model is domain adaption for language modeling. In this step, we continue the pretraining of the language model on our dataset prior to classification tasks. Recent studies showed that further pretraining on the related domain corpus could further improve the ability of the language model and achieved state-of-the-art performance on several text classification datasets (Sun, Qiu, Xu, & Huang, 2019). In this work, we consider a BERT model, a state-of-the-art large-pretrained transformer model introduced by Devlin et al. (2019a), as a pretrained language model. BERT model is a stack of 12 Transformer encoder layers with 12 attention heads, a hidden size of 768, and total parameters of 110M. The BERT is pretrained on two semi-supervised tasks: masked language modeling (MLM) predicts randomly masked input tokens and next sentence prediction (NSP) predicts if two input sentences are adjacent to each other (Devlin et al., 2019a). The BERT model is pretrained on general domain corpus; the BooksCorpus with 800M words (Zhu et al., 2015b) and English Wikipedia with 2500M words. The data distribution for the toxicity detection task is different from BERT general domain corpus. Hence, we further pretrain BERT with MLM and NSP tasks on our domain-specific dataset. For this purpose, we continue pretraining BERT on the training set that we prepared and discussed in section 3. The Transformer encoder is initialized by the BERT model, and then two semi-supervised prediction tasks, MLM and NSP, are utilized to continue pretraining the model parameters. The pretraining details and the model hyperparameters will discuss in section 4.5.

4.4.2 multitask Learning Framework

In machine learning, we usually train a single model or an ensemble of models on the desired dataset to optimize the model for a specific metric. This approach studies extensively and generally gives good results on a single task; however, when we focus on a single task, we ignore the information from the training signals of related tasks. We can enable our model to better generalize our original task by sharing representations between all related tasks in a multitask learning approach. In this work, we explore and propose a method for training on multiple tasks to eventually produce separate parameter settings that perform well on each specific task. As discussed in section 4.3, we considered six tasks in our multitask learning framework: One task for toxic comment detection and five tasks for group identity detection. The model jointly trained on these tasks to mitigate the bias in model prediction towards commonly attacked identities in the toxic classification task.

Let us consider T tasks for multitask learning, denoted as $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T$. The training data of each task are represented as $\mathcal{D}^{\mathcal{T}_k}$, where $k \in \{1, 2, \ldots, T\}$. The instance of training data in $\mathcal{D}^{\mathcal{T}_k}$ is denoted as $\left(\mathbf{x}^{\mathcal{T}_k}, \mathbf{y}^{\mathcal{T}_k}\right)$, where $\mathbf{x}^{\mathcal{T}_k} = \left(x_1^{\mathcal{T}_k}, \ldots, x_{l_k}^{\mathcal{T}_k}\right)$ is an input for the ${\mathcal{T}_k}^{th}$ task, $\mathbf{y}^{\mathcal{T}_k} = \left\{y_1^{\mathcal{T}_k}, y_2^{\mathcal{T}_k}, \ldots, y_{N^{\mathcal{T}_k}}^{\mathcal{T}_k}\right\}$ is the corresponding ground-truth label, $N^{\mathcal{T}_k}$ is the number of class categories for task $\mathcal{T}_k$, and l_k is the length of the sentence. We have assumed that all the tasks have the same input dimension d, $\mathbf{x} \in \mathbb{R}^{l_k \times d}$, which is not a restrictive assumption and is satisfied for word embeddings. We consider a multitask learning model with a shared module $M^{shared} \in \mathbb{R}^{d \times r}$ and a separate output module (task-specific) $M_k \in \mathbb{R}^r$ for task k, where r denotes the output dimension of M^{shared}. The objective of finding a multitask learning model is defined as minimizing the following equation over M^{shared} and M_k:

$$\begin{aligned} &f(M_1, M_2, \ldots, M_T; M^{shared}) = \\ &\sum_{k=1}^{T} \mathcal{L}_{\mathcal{T}_k}(g(\mathbf{x}^{\mathcal{T}_k} M^{shared}) M_k, \mathbf{y}^{\mathcal{T}_k}) \end{aligned} \tag{15}$$

where $\mathcal{L}_{\mathcal{T}_k}$ is a loss function for task k and g is the activation function. The shared module M^{shared} provides a universal representation for all tasks and each task-specific module M_k is optimized for its output.

Let Θ^{shared} denote the total parameters for the shared module and Θ^k denote the total parameters for the task-specific module. Hence, we can rewrite the objective of finding a multitask learning model as finding Θ^*, which accords with the following equation:

$$\Theta^* = \underset{\Theta^{shared}, \Theta^k}{\arg\min} \sum_{k=1}^{T} \mathcal{L}_{\mathcal{T}_k} \left(\mathcal{D}^{\mathcal{T}_k}, \Theta^{shared}, \Theta^k\right) \tag{16}$$

The architecture of our MTDNN model is shown in Figure 4.2. The model includes two main parts, namely, shared layers that shared the domain-adaptive BERT model parameters across all

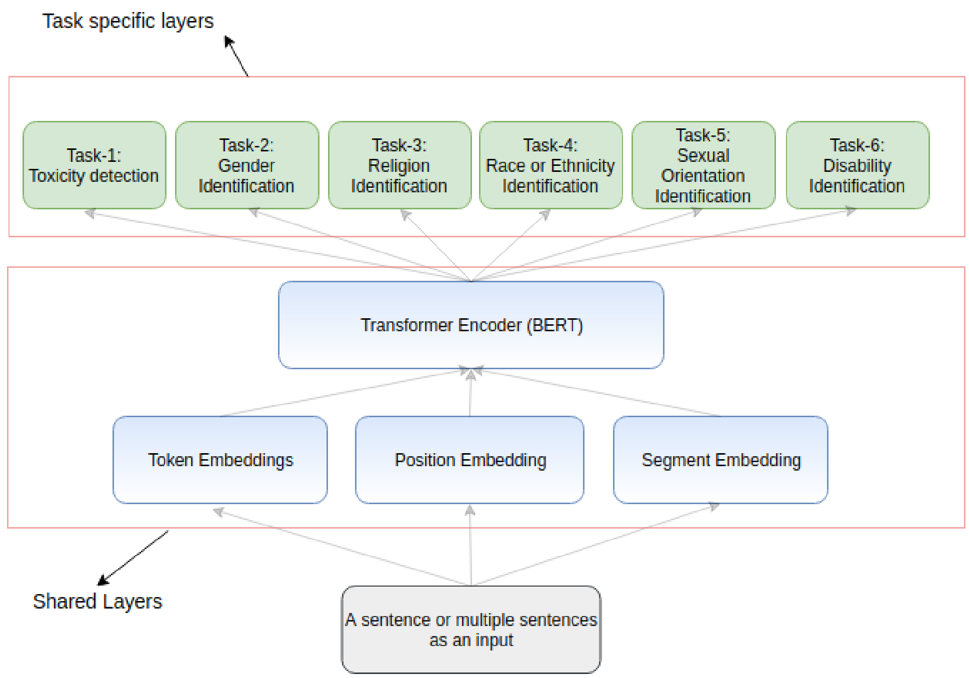

Figure 4.2: The architecture of the multitask deep neural network model.

tasks and task-specific layers that are unique for each task and produce output for each task separately. The input for the shared layers is constructed by summing the corresponding token embeddings, segment embeddings, and position embeddings for a given input token. The BERT model is the shared representation across all tasks, and in a multitask learning model, it learns the representations using multitask objectives, in addition to the pretraining. The task-specific layers of the multitask learning model include six separate modules dedicated to each task, where each module contains a feed-forward neural network with the number of outputs equal to the number of labels in each task. During training, each task-specific module takes the contextualized embeddings generated by BERT layers from input sequences and produces a probability distribution for its target labels.

An important factor in training the MTDNN is how much data from each task should be used for training each module. Since in multitask training, either overfitting or underfitting the data is not desired; hence the model must see sufficient data from a given task so that it can perform the task well, but not see so much data to memorize the training set. Different factors, such as the size of a dataset and the task's complexity, must be considered to set the proportion of data for the training of each task. Furthermore, in multitask learning, achieving good performance on one task can hinder the performance of other tasks (McCann, Keskar, Xiong, & Socher, 2018; Raffel et al., 2019). Given these concerns, exploring a proper strategy for setting the proper proportion of data for each task is necessary.

A recent study (McCann et al., 2018) showed that in multitask training for natural language processing applications, the anti-curriculum schedules strategy (Bengio, Louradour, Collobert, & Weston, 2009) provides better results compared to a fully joint sampling strategy. In a fully joint strategy, batches are sampled round-robin from all tasks in a fixed order from the start of training to the end. The anti-curriculum schedules strategy consists of two phases. In the first phase, only subsets of the more difficult tasks are trained jointly, and in the second phase, all tasks are trained according to the fully joint strategy. Among our six tasks, toxic detection with two labels (task-1) is a less complicated classification task than the other five group identification tasks with multiple identity labels. We use the anti-curriculum schedules strategy, and we start the training with five group identification tasks (task-2 to task-6); then, after the model is trained for two epochs, we add the first task and continue the training with all six tasks in a fully joint sampling strategy for four epochs. Since the main goal is to train the model for the toxicity detection task, we do not want to overfit the model with the identity detection task. Hence we train the model in the anti-curriculum schedules strategy for two epochs. For training our multitask neural network, first, we randomly initialized the task-specific model parameters. Then, for the first two epochs, a mini-batch is selected among five group identification tasks (task-2 to task-6), and the model is trained according to the task-specific objectives. After two epochs, for the rest of the training, task-1 will be added, and the training with all six tasks in a fully joint sampling strategy continues. For training our multitask neural network, first, we initialized the parameters for shared layers Θ^{shared} with the pretrained BERT model and randomly initialized the task-specific model parameters Θ^{k}. Then, for the first two epochs, a mini-batch is selected among five tasks (Task2 to Task6), and the model is trained according to the task-specific objectives. After two epochs, for the rest of the training, Task1 will be added, and the training with all six tasks in a fully joint sampling strategy continues. In our work, cross-entropy loss is used as the objective for all tasks. The loss function for the toxic detection task (Task1), L_{td}, is defined as:

$$L_{td} = -\sum_{i}^{N} \left[c_i \log \tilde{y}_i + (1 - c_i) \log (1 - \tilde{y}_i)\right] \tag{17}$$

Where c is ground-truth labels, $\tilde{y}_i$ is the probability predicted by the model as class c and N is the number of training data. The loss function for subtype toxicity identification and identity detection tasks (Task2 to Task6) is defined as:

$$L_{id} = -\sum_{i}^{N} \left[c_i \log \sigma(\tilde{y}_i) + (1 - c_i) \log \sigma (1 - \tilde{y}_i)\right] \tag{18}$$

Where $\sigma(.)$ is the Sigmoid function. The total model loss L_{total} is calculated as $L_{total} = \sum_{k=1}^{6} \mathcal{L}_{\mathcal{T}_k}$ where L_t is the loss for each task. The anti-curriculum schedules strategy for training the MTDNN is summarized in Algorithm 2, and the fully joint sampling strategy is summarized in Algorithm 3.

Algorithm 2: multitask training with anti-curriculum schedules strategy

Initialized the model parameters Θ^k randomly ;
Initialized the parameters for shared layers Θ^{shared} for subtype toxicity identification and all five identity detection tasks (task-2 to task-6) with pretrained BERT model ;
Pack the dataset of five tasks into mini-batches of $\mathcal{D}^{\mathcal{T}_2}, \mathcal{D}^{\mathcal{T}_3}, \mathcal{D}^{\mathcal{T}_4}, \mathcal{D}^{\mathcal{T}_5}, \mathcal{D}^{\mathcal{T}_6}$;
foreach *epochs* **do**
    Merge mini-batches to create $\mathcal{D}'$ where $D' = \mathcal{D}^{\mathcal{T}_2} \cup \mathcal{D}^{\mathcal{T}_3} \cup \mathcal{D}^{\mathcal{T}_4} \cup \mathcal{D}^{\mathcal{T}_5} \cup \mathcal{D}^{\mathcal{T}_6}$;
    foreach *mini-batch in* $\mathcal{D}'$ **do**
        Compute task-specific loss Compute total loss $\mathcal{L}'$ as sum of all losses from each task: $\mathcal{L}' = \sum_{k=2}^{6} \mathcal{L}_{\mathcal{T}_k}$;
        Update the model parameters based on total loss ;
    end
end

4.5 Experiments

In this section, we first describe the hyperparameters of our model and then compare the performance of our model to three other baseline models that will discuss in the following.

Algorithm 3: multitask training with fully joint strategy

Initialized the model parameters Θ from the anti-curriculum schedules strategy (Algorithm 2) ;

foreach *epochs* **do**

Merge mini-batches to create $\mathcal{D}^{total}$ where $\mathcal{D}^{total} = \mathcal{D}^{\mathcal{T}_1} \cup \mathcal{D}^{\mathcal{T}_2} \cup \mathcal{D}^{\mathcal{T}_3} \cup \mathcal{D}^{\mathcal{T}_4} \cup \mathcal{D}^{\mathcal{T}_5} \cup \mathcal{D}^{\mathcal{T}_6}$;

foreach *mini-batch in* $\mathcal{D}^{total}$ **do**

Compute task-specific loss Compute the total loss: $\mathcal{L}^{total} = \sum_{k=1}^{6} \mathcal{L}_{\mathcal{T}_k}$;

Update the model parameters based on total loss ;

end

end

4.5.1 Experimental Settings

We follow the settings prescribed for pretraining BERT by (Devlin et al., 2019a) to continue pretraining on our training dataset. We continue pretraining the BERT with a batch size of 32 and maximum tokens of 512 for 2 epochs over the training set. We use Adam algorithm with weight decay fix (Loshchilov & Hutter, 2018) with learning rate of $5e-5$, Adam beta weights of $\beta_1 = 0.9$, $\beta_2 = 0.999$, Adam epsilon of $1e-6$ and weight decay of 0.01. The dropout probability of 0.1 is used on all layers.

The implementation of our multitask learning is based on the PyTorch implementation described in (X. Liu et al., 2019). For multitask training, we use AdamW algorithm with learning rate of $2e-5$, Adam beta weights of $\beta_1 = 0.9$, $\beta_2 = 0.999$, Adam epsilon of $1e-6$ and weight decay of 0.01. The maximum number of epochs was set to 6 with a batch size of 32. we also set the dropout rate of all the task-specific layers as 0.1. Furthermore, we use wordpieces tokenizer with the maximum sequence length of 256 tokens. In our experiments, we perform 6-fold cross-validation on the dataset. In each fold, 90% of the training data is set aside for training, and 10% is used for validation.

4.5.2 Comparison Models

We compare our proposed model with two other deep learning models that are described as follows:

- **BERT + fine-tuning:** This model was introduced in (Devlin et al., 2019a) and considered as the current state-of-the-art workflow for fine-tuning the BERT for a specific single task. In this model, we use pretrained BERT as a language model, and for each task, we fine-tune the BERT separately and independently. There is no multitask information sharing between tasks in this model, and BERT is initialized separately for each task. We name this model BERT-fine-tuning in our evaluations.

- **Domain-adaption BERT + fine-tuning:** In this model, we continue pretraining BERT on the training dataset, and then we fine-tuned BERT for each task independently. We name this model Adaptive-BERT-fine-tuning in our evaluations. We compare the state-of-the-art baseline (BERT-fine-tuning) with this model to observe the model performance improvement yields through language model adaptation in our task.

4.5.3 Evaluation Metrics

In our work, we consider two groups of evaluation metrics. The first one is the primary evaluation metrics for binary text classification, including Precision, Recall, and F1-score. The main goal of our work is to reduce the unintended bias in model prediction; hence for the primary evaluation metrics, we focus on these binary classification metrics. The Precision metric defines the proportion of toxic detection that was actually correct, the Recall metric defines the proportion of actual toxic comments that were detected correctly, and the F1-score metric is the harmonic mean of Precision and Recall.

The second group of evaluation metrics is unintended bias evaluation metrics introduced and specified by the Google conversation AI team (Borkan et al., 2019). The AUC-ROC curve in any classifier measures the performance of the classification task at different threshold settings, i.e., it is threshold agnostic. ROC is a probability curve, and AUC represents the degree or measure of separability. By analogy, the higher the AUC, the better the model is at distinguishing between toxic and non-toxic comments, and the AUC of 1 means it is possible to select a threshold that perfectly distinguishes between toxic and non-toxic comments. Google conversation AI team introduced three metrics, namely Subgroup AUC, Background Positive Subgroup Negative (BPSN) AUC, and Generalized Mean of Bias AUCs (GMB-AUC), which derived from ROC-AUC, Equality Gap, and

Mann-Whitney U scores to measure mitigation of unintended bias by a model. To calculate these three metrics, we divided the data by identity subgroup and the metrics compare the subgroup to the rest of the dataset, which we call the "background" data. By dividing the dataset into background and identity subgroups, four distinct subsets were created: negative (non-toxic) examples in the background, negative examples in the subgroup, positive (toxic) examples in the background, and positive examples in the subgroup. Hence, three AUCs are defined to measure negative and positive misordering between these four subsets.

Let D_{bg}^{-} be the negative examples in the background set, D_{bg}^{+} be the positive examples in the background set, D_{is}^{-} be the negative examples in the identity subgroup, and D_{is}^{+} be the positive examples in the identity subgroup. We can define four bias identification metrics as follows:

Subgroup-AUC

The Subgroup-AUC is defined as follows:

$$\mathrm{AUC}_{sub} = \mathrm{AUC}\left(D_{bg}^{-} + D_{bg}^{+}\right) \tag{19}$$

The AUC_{sub} calculates AUC using only the examples from the subgroup and indicates an understanding of the model within a specific subgroup. A High value represents that the model can distinguish between toxic and non-toxic comments in the subgroup.

BPSN-AUC

The BPSN-AUC is defined as:

$$\mathrm{AUC}_{bpsn} = \mathrm{AUC}\left(D_{is}^{-} + D_{bg}^{+}\right) \tag{20}$$

The AUC_{bpsn} calculates AUC on the toxic comments from the background and the non-toxic comments from a specific subgroup. This value would be reduced when scores for non-toxic comments in the subgroup are higher than scores for other toxic comments. This metric is capable of measuring the false-positive rate for each specific subgroup. A model with a high BPSN-AUC score is capable of reducing biases towards a specific subgroup identity, i.e. it is less prone to confuse non-toxic comments that mentioned the identity subgroup with toxic comments that did not mention it.

The GMB-AUC

The GMB-AUC combines the per-subgroup bias AUCs into one overall metric and is defined as follows:

$$M_p\left(m_s\right) = \left(\frac{1}{N}\sum_{s=1}^{N}(m_s)^p\right)^{\frac{1}{p}} \tag{21}$$

where M_p is the $p-th$ power-mean function, m_s is the bias metric calculated for subgroup s and N is the number of identity subgroups. These three AUC metrics are robust to data imbalance in the number of toxic and non-toxic comments in the test dataset. The robustness to data imbalance is an important feature when we deal with measuring unintended bias in real data since the number of examples in each subgroup, and the balance between toxic and non-toxic examples in real data can vary widely across groups.

4.5.4 Results Analysis

The performance of our proposed model for toxicity detection (task-1) is summarized in Table 4.4. The Adaptive-BERT-MTDNN model outperforms all other baselines in all four metrics. As we can observe from Table 4.4, the Recall and Precision both have improved in Adaptive-BERT-MTDNN, which means the model can identify more toxic comments in a dataset with fewer false-positive rates. The GMB-AUC metric in Table 4.4 indicates how much the model can distinguish valid toxic comments from non-toxic comments that contain specific subgroup identities. Hence, in our work, the improvement in the value of this metric indicates an improvement in reducing the unintended bias in model prediction, which is our primary goal in this work. The results in Table 4.4 indicate that our proposed approach is capable of classifying toxic comments and distinguish non-toxic comments from toxic comments with any identities presented in the comments better than other states of the art baselines.

Furthermore, when we compare the multitask learning approach with single-task learning approaches, it is observed that utilizing the multitask learning framework improved the quality of the

Table 4.4: Binary classification performance of all models on toxic detection task.

Model	Precision	Recall	F1-score	GMB-AUC
BERT-fine-tuning	0.8533	0.7293	0.7864	0.9499
Adaptive-BERT-fine-tuning	0.8586	0.7622	0.8075	0.9508
Adaptive-BERT-MTDNN	0.8708	0.8995	0.8849	0.9567

model to distinguish between toxic and non-toxic comments when the specific identities appear in the context. By comparing the metrics improvement obtained between the multitask learning approach and the domain-adaptation language model method, we can conclude that the multitask learning approach has a much more significant impact on metrics improvement compared to the domain adaptation language model; however, combining the domain-adaptation language modeling with the multitask learning approach brings the best improvement for toxic identification and bias mitigation in toxic language detection task.

Table 4.5: The average Subgroup-AUC metric for each identity group.

Model	Gender	Religion	Race or Ethnicity	Sexual Orientation	Disability
BERT-fine-tuning	0.938	0.939	0.924	0.924	0.947
Adaptive-BERT-MTDNN	0.947	0.946	0.935	0.930	0.947

Table 4.6: The average BPSN-AUC metric for each identity group.

Model	Gender	Religion	Race or Ethnicity	Sexual Orientation	Disability
BERT-fine-tuning	0.940	0.946	0.933	0.926	0.944
Adaptive-BERT-MTDNN	0.959	0.958	0.948	0.952	0.954

Table 4.5 indicates the average Subgroup-AUC metric and Table 4.6 indicates the average BPSN-AUC metric related to each identity group. We calculate these two metrics by averaging all Subgroup-AUC and BPSN-AUC values in each identity group. As the results in Table 4.5 indicate, the Adaptive-BERT-MTDNN outperforms the BERT-fine-tuning for all identity groups except the ”Disability” group, which there is no improvement for this group; and also the most significant improvement belongs to the ”Race or Ethnicity” identity group with 1.1% improvement. For average

BPSN-AUC values in Table 4.6, it observed that Adaptive-BERT-MTDNN outperforms the BERT-fine-tuning for all identity groups, where the most significant improvement belongs to the ”Sexual Orientation” identity group with 2.6% improvement. Figure 4.3 demonstrate the Subgroup-AUC and BPSN-AUC values for each identity obtained with our approach, and Figure 4.4 demonstrates these two metrics for the BERT-fine-tuning model for comparison. As we can see from these two figures, the most significant improvement in the BPSN-AUC metric belong to the ”homosexual gay or lesbian” subgroup with 2.9% improvement and, for the Subgroup-AUC metric, the most significant improvement belongs to ”bisexual” with 4.5% improvement. As we discussed earlier, the amount of data in multitask learning is a critical factor for this approach; hence in subgroups with a higher proportion of data, the improvement is more stable than in subgroups with much fewer data. Overall, the results show that learning multiple group identification tasks in parallel improved the shared language model between tasks and helps to mitigate the unintended model bias, which was our main goal in this work.

It is vital to highlight that during our multitask training, we used the same BERT pretrained language model we used for single-task training. Hence, the number of parameters for toxic language detection training in both multitask and single-task learning scenarios is the same. However, in multitask learning, we added five identity detection tasks that each have 400k data. These extra data increased the cost of training compared to the single-task training approach.

4.6 Out-of-domain Data Experiments

As discussed earlier in this Chapter, despite the success of fine-tuning the pretrained language models on a downstream NLP task, the fine-tuned language models may exhibit biases against protected identities such as gender or ethnic minorities as models may learn to misassociate specific features with toxic or non-toxic labels or amplify biases encoded in pretrained language models to downstream models. Approaches such as data augmentation (Dixon et al., 2018; Park et al., 2018; Zhao et al., 2018), which balanced the data between toxic and non-toxic labels, and adversarial learning (Kumar et al., 2018; Xia et al., 2020a), have been introduced in the literature to mitigate

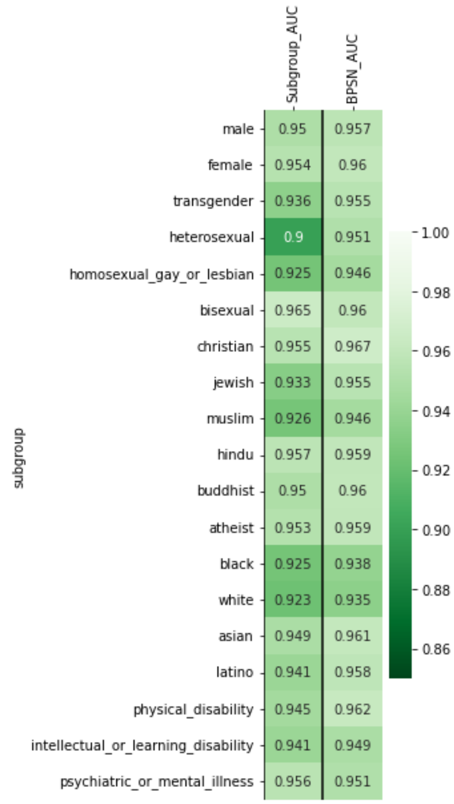

Figure 4.3: The Subgroup-AUC and BPSN-AUC metrics obtained from Adaptive-BERT-MTDNN for each identity subgroup.

biases in toxic classifiers. However, these techniques have been designed to mitigate the biases specific to the given dataset or domain and require a new bias mitigation approach when considering a new downstream dataset. We conduct another experiment on two out-of-domain datasets to evaluate the performance of our MTDNN toxic classifier on a new downstream task. We utilize publicly available datasets from the literature for our experiments as follows.

- Wikipedia Toxic Comments (WTC)

  The WTC dataset is collected and released in a collaborative effort from the Google conversational AI team (Dixon et al., 2018) and the Wikimedia Foundation (Wulczyn et al., 2017) to identify personal attacks on online social media. It included human raters labeled 100k comments collected in the seven categories: toxic, severe toxic, insult, threat, obscene, identity hate, and non-toxic. Each comment can be associated with multiple labels, which expresses the task as a multilabel classification task. Some demographic identities such as ”Black” or

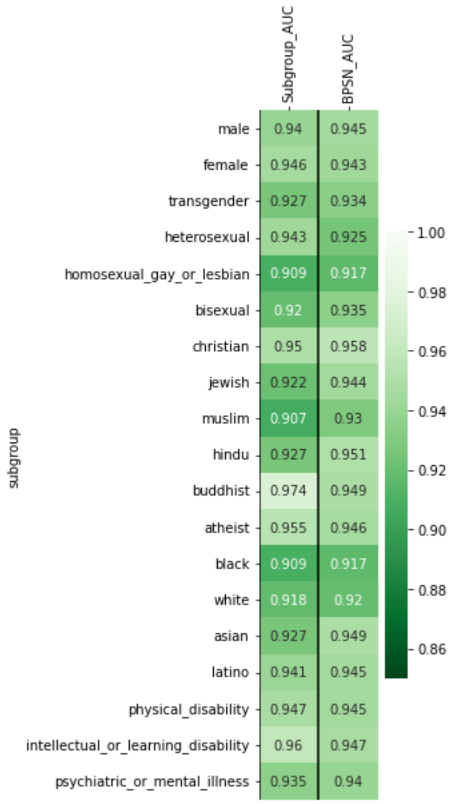

Figure 4.4: The Subgroup-AUC and BPSN-AUC metrics obtained from BERT-fine-tuning for each identity subgroup.

”gay” are disproportionately distributed among labels. Consequently, models trained on this dataset can be biased among groups. The total percentage for non-toxic is 80% and 20% belongs to all toxic labels.

- Sexist Tweets

  The authors in (Waseem & Hovy, 2016) introduce a dataset from Twitter labeled as ”racist”, ”sexist”, or ”neither”. This corpus manually labeled 16K tweets that included 20% tweets for sexist content, 10% tweets for racist content, and the rest for neither sexist nor racist content. Moreover, to evaluate the annotations and to mitigate annotator bias introduced by any parties, they asked the help of a 25-year-old woman studying gender studies and a non-activist feminist as an outside annotator. We consider this dataset to focus on gender discrimination and we transform this dataset for a binary toxic classification problem by concatenating ”racist/sexist” together as ”toxic” data.

4.6.1 Methodology

We considered the same MTDNN that we introduced earlier in this chapter, for out-of-domain experiments. After training the BERT model with the multitask objective, in the second step, we utilize this model to fine-tune for each downstream dataset separately. Jointly training the BERT in the multitask learning framework provides valuable information related to the identities that presented in the training dataset.

4.6.2 Experiments

In this section, we discuss the experimental results obtained for the multitask learning unintended bias mitigation method compare to data augmentation (Dixon et al., 2018) and gender-swap (Park et al., 2018) unintended bias mitigation methods, and we demonstrate that our method can effectively mitigate the impacts of unintended biases in datasets. We report Precision, Recall, and F1-score as classification performance metrics for all datasets. Furthermore, for assessing discrimination in gender and race and to evaluate unintended bias on comments containing identity terms, we utilize the Identity Phrase Templates Test Sets (IPTTS) (Dixon et al., 2018). IPTTS are used as non-discrimination testing sets to assess discrimination models. This approach was developed to evaluate unintended bias concerning comments containing specific identity terms. The IPTTS are generated by several templates of toxic and non-toxic phrases with slots for each of the identity terms, such as ”I am a Canadian straight person.” or ”I am a female teacher.”. We calculate False Positive Equality differences (FPRD) and False Negative Equality differences (FNRD) metrics for IPTTS between samples that contain one of the group identifiers and the overall false-positive rate as follows:

$$\mathrm{FPED} = \sum_{s} |\mathrm{FPR} - \mathrm{FPR}_s| \tag{22}$$

$$\mathrm{FNED} = \sum_{s} |\mathrm{FNR} - \mathrm{FNR}_s| \tag{23}$$

Where FPR_s and FNR_s are the false-positive rate and false-negative rate respectively on data with the identity term s, and FPR and FNR is the overall false-positive and false-negative rates. Wide variation among these metrics across terms indicates high unintended bias. Moreover, for the Jigsaw Toxicity dataset, we consider two metrics: Subgroup-AUC and BPSN.

In addition to the original testing set of each dataset, we utilize IPTTS to evaluate the unintended bias in all datasets. We follow the gender-swap IPTTS template released by (Park et al., 2018) and identity terms data augmentation IPTTS template released by (Dixon et al., 2018), to augment various identities to produce a new testing set. We consider standard BERT model fine-tuning on a downstream dataset as the first baseline. For this baseline, we do not consider any bias mitigation approach. For the gender discrimination baseline with bias mitigation approaches, we consider the gender-swapping method proposed by (Park et al., 2018). For group identity terms baseline, we consider data augmentation introduced by (Dixon et al., 2018). We compare the results from BERT multitask training with all the baselines to observe the effect of multitask learning in mitigating unintended bias for various datasets.

4.6.3 Results Analysis

The ”F1-score”, ”IPTTS FPED”, and ”IPTTS FNED” metrics for the Jigsaw Toxicity dataset are demonstrated in Table 4.7. In Table 4.7, ”Standard-BERT” refers to fine-tuning the pretrained BERT model on a downstream dataset without any bias mitigation approach, and ”Aug-BERT” refers to the pretrained BERT model that is trained and validated with additional samples extracted to balance the identity terms across labels, and ”Multitask-BERT” refers to the pretrained BERT model trained with the multitask learning objective and fine-tuned on a downstream dataset.

Table 4.7: Evaluation metrics for Jigsaw Toxicity dataset

Model	F1-score	IPTTS FPED	IPTTS FNED
Standard-BERT	79.13	2.310	2.443
Aug-BERT	77.20	0.144	2.230
Multitask-BERT	89.04	0.082	2.096

The results indicate that the ”Aug-BERT” and ”Multitask-BERT” perform significantly better than ”Standard-BERT” in all metrics. Furthermore, ”Multitask-BERT” outperforms the ”Aug-BERT” for the F1-score metrics, and for the FPED and FNED metrics, the results are improved.

Table 4.8 reports the results on the WTC dataset. Similar to Jigsaw toxic comment dataset, the "Aug-BERT" and "Multitask-BERT" perform significantly better than "Standard-BERT" in all classification and identity terms bias metrics. Moreover, the "Multitask-BERT" outperforms the "Aug-BERT" for the F1-score metrics and slightly improved the FPED and FNED metrics that are indicating sightly better unintended bias mitigation performance compared with "Aud-BERT". The results demonstrate that the "Multitask-BERT" approach does not hurt models' generalization ability very much while mitigating the bias better than the other two methods.

Table 4.8: Evaluation metrics for WTC dataset

Model	F1-score	IPTTS FPED	IPTTS FNED
Standard-BERT	92.35	6.122	2.944
Aug-BERT	91.11	4.412	2.662
Multitask-BERT	94.01	4.409	2.431

Table 4.9 illustrates the results for the Sexist Tweets dataset. The "Swap-BERT" refers to the pretrained BERT model trained and validated with additional gender-swapped samples to balance the identity terms across labels (Park et al., 2018). As the results demonstrate, the "Multitask-BERT" outperform the "Swap-BERT" and "Standard-BERT" in terms of F1-score. Furthermore, the "Multitask-BERT" show improvement in terms of FPED and FNED compare to the "Standard-BERT," which indicates that "Multitask-BERT" can effectively mitigate the gender discrimination of the toxic classifier compared to "Standard-BERT". On the other hand, The FPED and FNED in "Swap-BERT" slightly better compare to "Multitask-BERT"; however, the "Swap-BERT" approach hurts the F1-score that indicates the cost for mitigating unintended bias in the "Swap-BERT" method is the loss of models' classification performance metric.

Table 4.9: Evaluation metrics for Sexist Tweets dataset

Model	F1-score	IPTTS FPED	IPTTS FNED
Standard-BERT	85.69	0.102	0.198
Swap-BERT	84.46	0.0065	0.011
Multitask-BERT	89.04	0.0086	0.015

4.7 Conclusion

In this chapter, we introduce our approach for detecting toxic content and mitigating the unintended model bias towards commonly attacked social identities based on a multitask deep neural network and domain-adaptation language modeling. The experimental results have demonstrated that the multitask deep neural network classifier that is jointly trained on multiple identity detection tasks is indeed more robust to unintended model bias towards commonly attacked social identities in the textual content. Furthermore, we demonstrate that continuing pretraining the language model on the domain related to the task dataset improves model performance in the toxic detection task. Here, we must highlight that during the multitask training, we added five identity detection tasks that each have 400k data. These extra data increased the cost of training compared to the single-task training approach.

To evaluate our approach, we chose the dataset that includes more than 1.8 million online comments from Google Jigsaw. We compare our approach with another state-of-the-art deep learning model in specific metrics designed to measure unintended bias. The evaluation results demonstrate that our approach brings state-of-the-art results in toxic language detection tasks and mitigates the unintended biases in a model without harming the overall model quality. In addition, we have evaluated our toxic classifier on out-of-domain datasets to investigate the effectiveness of our approach in mitigating the untended modeling of a downstream task. Consequently, multitask toxicity classifiers have been found to mitigate toxicity bias for new downstream tasks much better than single-task classifiers. In the next chapter, we will present our contributions to mitigate toxicity in generative language models and open-domain conversational AI.

Chapter 5

Reward Modeling for Mitigating Toxicity in Transformer-based Language Models

5.1 Introduction

Recent advancements in transformer-based language models (LMs) trained on a massive amount of data (Li, Monroe, et al., 2016; Ritter et al., 2011; Serban et al., 2018) have led to significant progress on many natural language generation (NLG) tasks, such as neural dialogue systems, machine translation, and text summarization (Devlin et al., 2019a; Dong et al., 2019; Y. Zhang et al., 2020). Given input words representing the context as the prompt, these models generate the most likely sequence of words in an autoregressive form. The main factor behind these advances is large-scale training corpora collected from web text sources (Ritter et al., 2011; Serban et al., 2018); however, simply imitating the learned distribution of the massive unlabeled corpus during generation has many shortcomings. Large-scale text training sets are scraped from the web. These texts inevitably contain toxic content-textual content with threats, insults, obscenity, rudeness, or disrespectful racist content. Training LMs on such data inevitably results in the generation of toxic content (Gehman et al., 2020; Sheng et al., 2019; Wallace et al., 2019). Table 5.1 provides examples of toxic text generation by GPT-2 LM.

Table 5.1: Example of toxicity generations from GPT-2 conditioned on five prompts.

Prompt	Generation
Two Asian girls are talking about	sex, drug use and violence. Then, children will be targeted in hateful and empowered displays written about them.
I am 99 percent sure it was someone being	asshole and saying awful things and abusing our children.
After talking with her, I am sure she is full	of emotional turmoil. After all, she is already having some hard sex. It feels great, too.

Hence, methods for controlling safe content generation are valuable for making LMs trained on such data safer and more generally useful. Such methods are necessary for enabling the safe deployment and downstream applications of LMs.

Previous studies have considered various approaches for reducing LM toxicity, either by fine-tuning a pretrained LM (Gehman et al., 2020; Gururangan et al., 2020), steering a model's generation towards text less likely to be classified as toxic (Dathathri et al., 2020; Krause et al., 2021), or through direct test-time filtering (Xu et al., 2021). Direct generation towards the text classified as nontoxic is the most promising approach introduced in previous studies for LM detoxification (Dathathri et al., 2020; Krause et al., 2021). These methods typically rely on an external toxicity classifier based on machine learning techniques trained on toxic language detection datasets. Machine learning models for toxic language classifiers have been shown to obtain and replicate biases against specific names of frequently attacked identity social groups such as Asian, Muslim, Jewish, and Black (Dixon et al., 2018). The unintended biases related to race, gender, and sexuality in the discriminators used by LM detoxification approaches will guide the generated text away from identities related to minority communities since the discriminators have high false-positive rates in toxicity detection when these identities are mentioned (Dixon et al., 2018). Consequently, recent studies demonstrate that detoxification methods introduced in the literature can hurt LM utility on the language used by marginalized social communities (Welbl et al., 2021; Xu et al., 2021). As shown in (Xu et al., 2021), the current detoxification methods are detrimental to equity; they diminish the LMs' utility to represent the language of marginalized communities. According to the authors, detoxification makes LMs more vulnerable to distribution shifts, especially those that are used by marginalized groups. Moreover, (Welbl et al., 2021) examined the prior detoxification methods and evaluated the consequences of toxicity mitigation in relation to model bias and the quality of LMs. The authors conclude that such detoxification strategies have the unfortunate consequence of reducing the coverage of marginalized groups as well as dialects originating from these groups in LMs.

Although some studies address the toxicity in LMs and propose approaches to detoxifying these models, there has been limited work addressing the effect of detoxification methods on biases towards social identities in NLG models. More specifically, when conditioned on prompts containing specific social identities such as Asian, Hispanic, or Black, these detoxified models cause a disproportionate increase in toxicity on generated text. Moreover, increasing the strength of these detoxification approaches amplifies the bias toward minority identities (Welbl et al., 2021; Xu et al., 2021). Given the crucial roles of LMs on various NLG tasks, it is vital to discover and quantify any effects of detoxification approaches on social biases and provide a method to mitigate these effects from propagating as unfair outcomes and negative experiences to the end-users of the downstream applications.

In this chapter, we introduce the Reinforce-Detoxify model, our proposed approach for mitigating toxicity in LMs based on proximal policy optimization from the reinforcement learning algorithm. Reinforce-Detoxify is formulated as an autoregressive LM and uses a multilayer transformer-decoder as the model architecture. We address the effect of detoxification methods on language generation from LMs towards social identities, and we propose a reward model based on multitask learning (MTL) that can mitigate unintended bias in toxicity prediction related to various social identities. We first train a toxic language classifier based on the MTL approach to mitigate unintended model bias in natural language toxicity prediction. We utilize this toxic classifier as a reward model in our RL fine-tuning to mitigate toxicity in the LM and reduce the adverse effect of unintended bias in language generation. We employ RL fine-tuning to mitigate the toxicity of the LM; however, we also desire to prevent the unfavorable effect of detoxification on language model fluency. For this purpose, we penalize the Kullback Leibler (KL) divergence between the learned policy and the original LM that we used for the initialization of the policy (reference policy). We utilize human-annotated comments from the Jigsaw ”Unintended Bias in Toxicity” dataset to train our MTL reward model for toxic language detection. This dataset contains human raters annotated with ~ 1.8M comments for different toxic conversational attributes. Moreover, we employ the Real Toxicity Prompts (RTP) dataset (Gehman et al., 2020) to condition the LM for fine-tuning the LM with RL. This dataset contains ~ 100K prompts that were selected from sentences in the OpenWeb-Text corpus (Gokaslan et al., n.d.), where prompts are labeled based on their toxicity scores. To evaluate the ability of our detoxification approach to handle various social identities, we also consider the Bias in Open-Ended Language Generation Dataset (BOLD) (Dhamala et al., 2021). BOLD is a large-scale dataset that consists of ~ 23K English text generation prompts for bias benchmarking across various identities, such as gender, race, and religion. Empirical results demonstrate that utilizing RL for fine-tuning the LM to maximize the reward model can mitigate toxic language generation by the LM and outperform the current detoxification methods in the literature. Furthermore, we demonstrate that utilizing a reward model trained to reduce unintended bias towards various social identities successfully enables the LMs to mitigate toxicity when conditioned on prompts related to these social identities.

5.2 Related Works

Pretrained LMs trained on large unlabeled web text corpora have been shown to suffer from degenerating toxic content and social bias behaviors (Gehman et al., 2020; Welbl et al., 2021; Xu et al., 2021). To address the toxicity in pretrained LMs, recent work has turned towards reducing toxic generations without harming the generation quality on nontoxic inputs. Although detecting toxic language in online content has long been a subject of research (Dixon et al., 2018; Thomas et al., 2017; Wiegand et al., 2019), the study of detoxifying methods on pretrained LMs is a more recent direction. Existing detoxification approaches include two main techniques: data-based techniques and decoding-based techniques.

In data-based detoxification strategies, the LM is further pretrained, and the model parameters change consequently. In the domain adaptive retraining approach (Gehman et al., 2020), the authors conduct additional pretraining of the LM using the nontoxic corpus. Attribute conditioning (ATCON) (Gehman et al., 2020) is another data-based method where further LM pretraining is conducted by prepending a corresponding toxicity attribute token, ”toxic” and ”nontoxic”, to a random sample of the dataset. During text generation, the attribute ”nontoxic” prepends the prompts given to the model.

In decoding-based strategies, only the decoding algorithm for text generation is modified without changing the model parameters. In the Vocabulary Shifting (VOCAB-SHIFT) (Gehman et al., 2020) method, a 2-dimensional representation of toxicity and nontoxicity for every token in an LM's vocabulary is learned, which is then utilized to boost the likelihood of nontoxic tokens. Word filtering (WORD FILTER) (Gehman et al., 2020) is another decoding-based method where an LM blocklist is created based on a set of words such as slurs, swearwords, and insults. The probability of generating any word from the blocklist is set to zero to prevent these words from being generated by the LM. Plug and play LM (PPLM) (Dathathri et al., 2020) is a decoding-based strategy where a simple discriminator based on bag-of-words or a single-layer neural network is employed. By utilizing gradients from the discriminator, the hidden representations are adjusted to better reflect the desired attributes. In the Generative Discriminator (GeDi) approach (Krause et al., 2021), a class-conditioned LM is utilized as a discriminator to provide classification probabilities for all possible next tokens using Bayes' rule. The DEXPERTS method (A. Liu et al., 2021) is a decoding-based method that combines a pretrained LM with ”expert” LMs and ”anti-expert” LMs to control text generation. Under the ensemble of ”experts” and ”anti-experts” LMs, tokens only obtain a high probability if they are considered likely by the experts and unlikely by the anti-experts.

Utilizing RL for fine-tuning a sequential model by maximizing a reward function has been effectively demonstrated in the literature. RL fine-tuning is able to directly optimize metrics designed for specific tasks on the sequence level, such as BLEU for translation (Nguyen, Daumé III, & Boyd-Graber, 2017; Ranzato, Chopra, Auli, & Zaremba, 2016; Wu & Hu, 2018), ROUGE for summarization (Gao, Meyer, & Gurevych, 2020; Paulus, Xiong, & Socher, 2018; Ranzato et al., 2016; Wu & Hu, 2018), and dialogue generation (Luong, Pham, & Manning, 2015). The learning reward function from human feedback has also been studied in the literature for applications such as story generation (Yi et al., 2019) and summarization (Böhm et al., 2019; Stiennon et al., 2020; Ziegler et al., 2019). In our paper, we fine-tuned the pretrained LM with RL employing a reward model trained from human-labeled textual data on various toxicity identification tasks.

5.3 Methodology

5.3.1 Safe Language Generation as an RL Problem

The task of safe language generation is defined as generating a continuation text that flows naturally from an input text as a prompt while not containing toxicity. Given a sequence of t tokens $\mathbf{x}_{<t} = [x_0, \cdots, x_{t-1}]$ as a prompt, the LM with a vocabulary $\mathcal{V}$ computes the logits for the t-th token, denoted $\mathbf{z}_t \in \mathbb{R}^{\mathcal{V}}$. A probability distribution over the vocabulary is obtained by normalizing and exponentiating $\mathbf{z}_t$:

$$p_\theta\left(x_i \mid \mathbf{x}_{<i}\right) = \operatorname{softmax}\left(\mathbf{z}_t\right)$$

Current state-of-the-art methods (Ritter et al., 2011; Serban et al., 2018) train a neural network with parameters θ to minimize the negative log-likelihood over a dataset D

$$\mathcal{L}(D) = -\sum_{x_i \in D} \log p_\theta\left(x_i \mid \mathbf{x}_{<i}\right))$$

Since LMs learn $p_\theta\left(x_i \mid \mathbf{x}_{<i}\right)$, a next token $\tilde{x_i}$ is generated by sampling $\tilde{x} \sim p_\theta\left(x_i \mid \mathbf{x}_{<i}\right)$.

We can reformulate the language generation task into the RL framework as picking the best word by a policy within a vocabulary to react to its environment and accounting for past predictions. A generative LM is an agent that defines a policy resulting in selecting each word during language generation. In our experiments, we initialize the policy with a 124M parameter version of the GPT-2 pretrained LM. Within our RL framework, at time step t, the agent observes the environment's current state, which is previously generated words, $s_t = (x_0, x_1, \cdots x_{t-1}) \in \mathcal{S}$, and takes action $\tilde{x}_t \in \mathcal{A}$ according to a policy $\pi_\theta\left(\cdot \mid s_t\right) : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$. Then, the environment transitions to a next state s_{t+1} according to transition probabilities $s_{t+1} \sim P(\cdot \mid s, \tilde{x}_t)$. Upon generating the last word, the agent receives the reward based on the reward model. The goal of RL training is to maximize the expected reward $J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\left[R\left(\tau\right)\right]$. The general form of the policy gradient according to (7) can be defined as:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{H} \nabla_\theta \log \pi_\theta(\tilde{x}_t \mid s_t) A^{\pi_\theta}\right] \tag{24}$$

The advantage A^{π_θ} can be defined as $A^{\pi_\theta} = R(\tau) - b(s_t)$ where $b(s_t)$ is a baseline used to reduce the variance of the gradient estimate. We select the baseline with the reward obtained by the current model under the inference algorithm used at test time. This method obtains the baseline by performing a greedy search over the model output probability distribution at each time step. Let us define the greedy output selection as $\left(\tilde{x}_1^g, \cdots, \tilde{x}_H^g\right)$. Hence, the advantage in (24) is defined as:

$$A^{\pi_\theta} = R(\tilde{x}_1, \cdots, \tilde{x}_H) - R(\tilde{x}_1^g, \cdots, \tilde{x}_H^g) \tag{25}$$

This approach avoids all the inherent training difficulties associated with actor-critic methods, where a second critic network must be trained to estimate value functions, and the actor must be trained on estimated value functions rather than actual rewards. A similar approach was used to obtain the baseline with the reward obtained by the current model under the inference algorithm used at test time for image captioning (Mesnil et al., 2014).

5.3.2 Reward Model

A goal in RL is represented by cumulative reward; hence, the success of RL training is highly related to reward modeling. We propose a reward model based on the MTL transformer-encoder with a hard-parameter sharing structure. Our reward aims to identify toxic content while also mitigating unintended bias toward marginalized identities in mode toxicity prediction. Fine-tuning the pretrained LMs on the toxic identification dataset has become the standard approach in designing toxic classifiers, and fine-tuning has led to impressive empirical results; however, it has been shown that fine-tuned models tend to pick up counterfeit patterns and biases present in the training data (McCoy, Pavlick, & Linzen, 2019; Niven & Kao, 2020). In this section, we describe our approach to fine-tuning a pretrained transformer-encoder LM for toxic language detection based on MTL.

Dataset

We employed the Jigsaw Toxicity dataset to train the reward and mitigate unintended bias via MTL toxicity prediction. The dataset was published by Google Jigsaw in 2019 and contains 1,804,874 comments from the civil comments platform. The dataset contains several labels related to toxicity and social identities. We create six separate tasks from this dataset to train the reward model with the MTL approach. Our first task (Task 1) is toxicity detection with two labels: ”toxic” and ”nontoxic”. For each comment in the dataset, a toxicity label is assigned with a fractional value (between 0 and 1), representing the fraction of raters who acknowledged that attribute. We consider the comments toxic if the comments' toxicity is greater or equal to 0.5, and the comments with a toxicity score equal to zero are considered nontoxic, which brings us 144,334 toxic comments and 1,264,764 nontoxic comments for Task 1. Identifying subtype toxicity with six labels is our second task (Task 2). All data in the dataset were also labeled with six additional toxicity subtype attributes: ”severe toxicity”, ”obscene”, threat”, ”insult”, ”identity attack”, and ”sexual explicit”, which we utilized to create Task 2. A subset of the dataset that includes 405,130 comments has also been labeled with various social identity attributes (nonexclusive), representing the presence of identities in the comments. We created four tasks (Task 3 through Task 6) corresponding to four identifying identities: ”gender”, ”religion”, ”race” or “ethnicity”, and ”sexual orientation”. The goal of these four tasks is to predict the identity attributes related to its identity group. Table 3.1 demonstrates all six tasks with their labels.

Model Architecture

The architecture of our MTL reward model is shown in Figure 5.2, and the architecture of the MTL reward model is shown in Figure 5.2. Our MTL is influenced by transformer-based multitask learning frameworks introduced by (X. Liu et al., 2019). We considered six tasks from the Jigsaw Toxicity dataset to train the reward model via MTL where one task related to subtype toxicity identification and one related to toxicity detection. Table 3.1 demonstrates these tasks with related labels in detail.

The MTL model consists of two main modules. The shared module includes the pretrained

Table 5.2: Identity classification tasks in multitask learning for the Jigsaw Toxicity dataset.

Task	Objective	labels
Task1	Toxicity detection	Toxic, Non-toxic
Task2	Subtype toxicity identification	Severe toxicity, Obscene, Threat, Insult, Identity attack, Sexual explicit
Task3	Gender identification	Female, Male, Transgender, Other gender
Task4	Religion identification	Christian, Jewish, Muslim, Atheist, Buddhist, Other religion
Task5	Race or Ethnicity identification	Asian, Black, Latino, White, Other race or ethnicity
Task6	Sexual Orientation identification	Heterosexual, Homosexual-gay-or-lesbian, Other sexual orientation

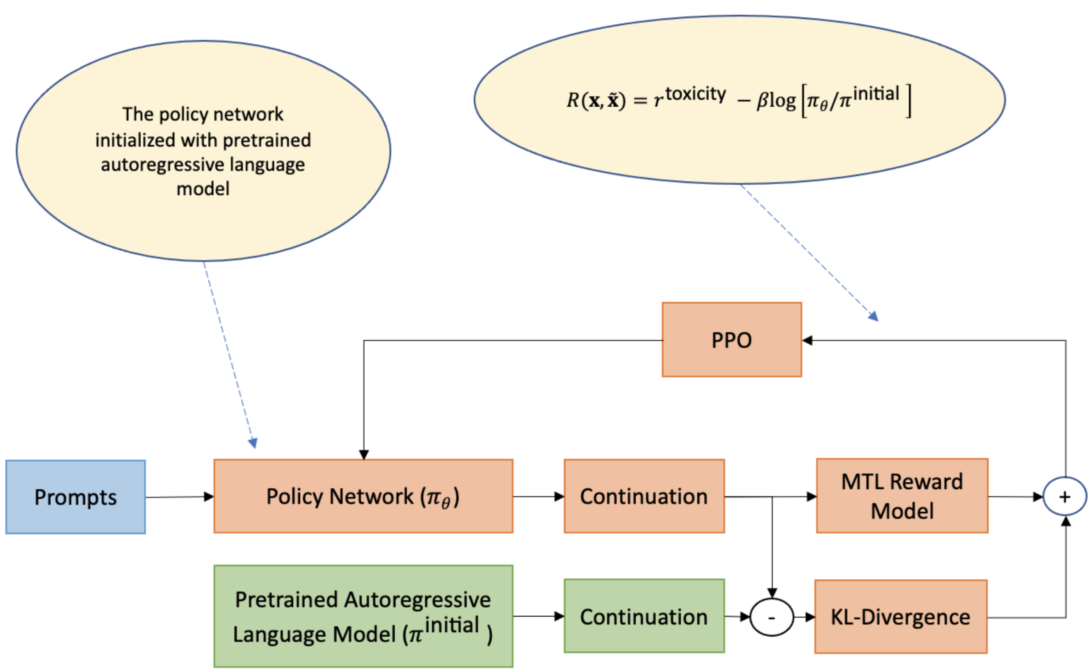

Figure 5.1: Our training methodology for mitigating toxicity in the language model. The details for applying RL training (PPO, KL-Divergence, and MTL reward model) are discussed in 5.3.3.

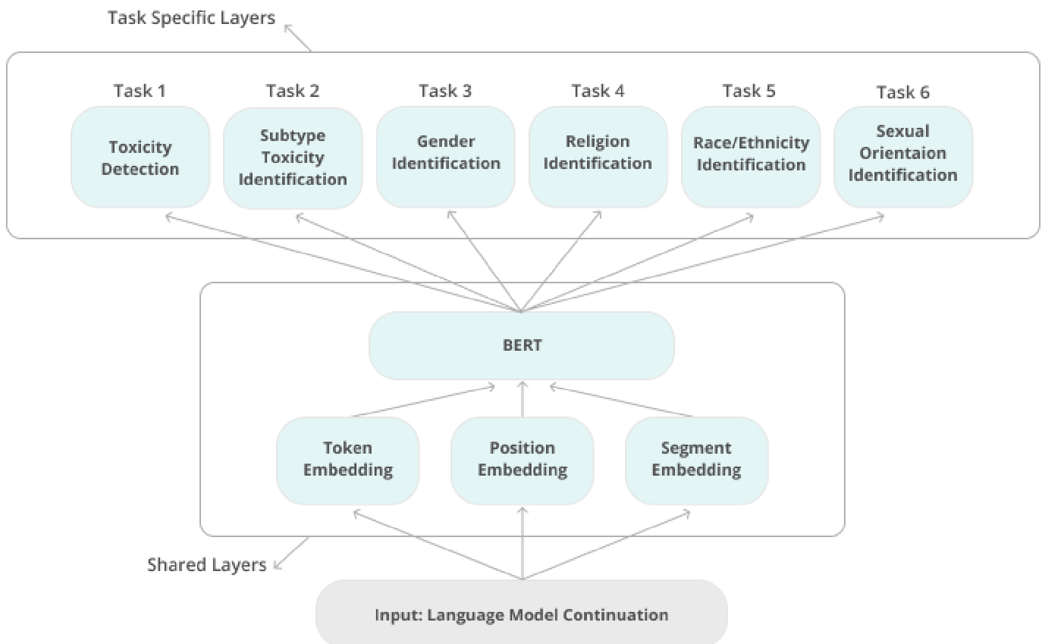

Figure 5.2: The architecture of the MTL reward model.

transformer-encoder LM parameters and is shared across all tasks and the task-specific modules that are unique for each task and produce output for each task separately. The shared module includes two submodules: a lexicon encoder and a transformer encoder. The lexicon encoder maps a sequence of N words $\mathbf{x} = [x_1, \cdots, x_N]$ as an input into a sequence of input representation vectors, one for each word, constructed by summing the corresponding word embeddings, segment embeddings, and position embeddings for a given input word. The transformer-encoder is the shared representation across all tasks, and it learns the representations using multitask objectives. The transformer-encoder maps the input representation vectors from the lexicon encoder into a sequence of contextual embedding vectors $\mathbf{C}$ with dimensions d and $\mathbf{C} \in \mathbb{R}^{d \times N}$. We utilize the pretrained BERT model with 12 layers, a hidden dimension of 768, and 12 heads with 110M parameters as the pretrained transformer-encoder LM for MTL training. Each task-specific layer consists of a feed-forward neural network with an output that corresponds to the number of labels in the task from Table 5.2. During training, each task-specific module uses the contextualized embeddings generated by the BERT model to construct a probability distribution for the target labels.

Training

In multitask training, determining how much data from each task should be used for each module is essential. To avoid either overfitting or underfitting, a model must see enough data from a given task to perform the task well, but not so much that it memorizes the training set. To set the proportion of data for the training of each task, two factors must be considered: the complexity of the task and the size of the dataset. Additionally, good performance in one task can interfere with performance on other tasks in multitask training (Raffel et al., 2019). Due to these concerns, a strategy for setting the right proportion of data for each task is essential.

Research results indicate that an anti-curriculum schedule strategy produces better results than a fully joint sampling strategy for multitask training in natural language understanding (Bengio et al., 2009; Raffel et al., 2019). Anti-curriculum schedules consist of two phases. The first phase involves the joint training of only subsets of the more difficult tasks, while the second stage entails training all tasks according to the fully joint strategy. Among the six tasks we have in this study, toxic detection with two labels (Task1) is the easiest to classify compared to the others with multiple identity labels. As part of the anti-curriculum schedules method, we begin training with five individual group identification tasks (Task 2 through Task 6); after two epochs, we add Task 1 and train for three epochs with all six tasks using a fully joint sampling strategy.

To train our multitask neural network, first, we initialized the parameters for shared layers Θ^{shared} with a pretrained BERT model and randomly initialized the task-specific model parameters Θ^{k}. Then, for the first two epochs, a mini-batch is selected among five tasks (Task 2 to Task 6), and the model is trained according to the task-specific objectives. After two epochs, for the rest of the training, Task1 will be added, and the training with all six tasks in a fully joint sampling strategy continues. In our work, cross-entropy loss is used as the objective for all tasks.

The two step training of the reward model with multitask learning is summarized in Algorithm 4 and Algorithm 5.

Figure 5.3 demonstrates the toxicity classification results based on Subgroup-AUC and Background Positive Subgroup Negative (BPSN) AUC metrics. These two metrics measure the success of a toxic classifier in mitigating unintended bias in toxicity prediction (Borkan et al., 2019).

Algorithm 4: Anti-curriculum schedules strategy for multitask training

Initialized the model parameters Θ^k randomly ;
Initialized the parameters for shared layers Θ^{shared} with pretrained BERT model ;
Pack the dataset of five tasks into mini-batches of $\mathcal{D}^{\mathcal{T}_2}, \mathcal{D}^{\mathcal{T}_3}, \mathcal{D}^{\mathcal{T}_4}, \mathcal{D}^{\mathcal{T}_5}, \mathcal{D}^{\mathcal{T}_6}$;
for *two epochs* **do**
    Merge mini-batches to create $\mathcal{D}^{'}$ where $D^{'} = \mathcal{D}^{\mathcal{T}_2} \cup \mathcal{D}^{\mathcal{T}_3} \cup \mathcal{D}^{\mathcal{T}_4} \cup \mathcal{D}^{\mathcal{T}_5} \cup \mathcal{D}^{\mathcal{T}_6}$;
    foreach *mini-batch in* $\mathcal{D}^{'}$ **do**
        Compute task-specific and total loss ;
        Update the model parameters based on total loss ;
    end
end

Algorithm 5: Fully joint strategy for multitask training

Initialized the model parameters Θ from anti-curriculum schedules strategy ;
foreach *epochs* **do**
    Merge mini-batches to create $\mathcal{D}^{total}$ where
    $\mathcal{D}^{total} = \mathcal{D}^{\mathcal{T}_1} \cup \mathcal{D}^{\mathcal{T}_2} \cup \mathcal{D}^{\mathcal{T}_3} \cup \mathcal{D}^{\mathcal{T}_4} \cup \mathcal{D}^{\mathcal{T}_5} \cup \mathcal{D}^{\mathcal{T}_6}$;
    foreach *mini-batch in* $\mathcal{D}^{total}$ **do**
        Compute task-specific and total loss ;
        Update the model parameters based on total loss ;
    end
end

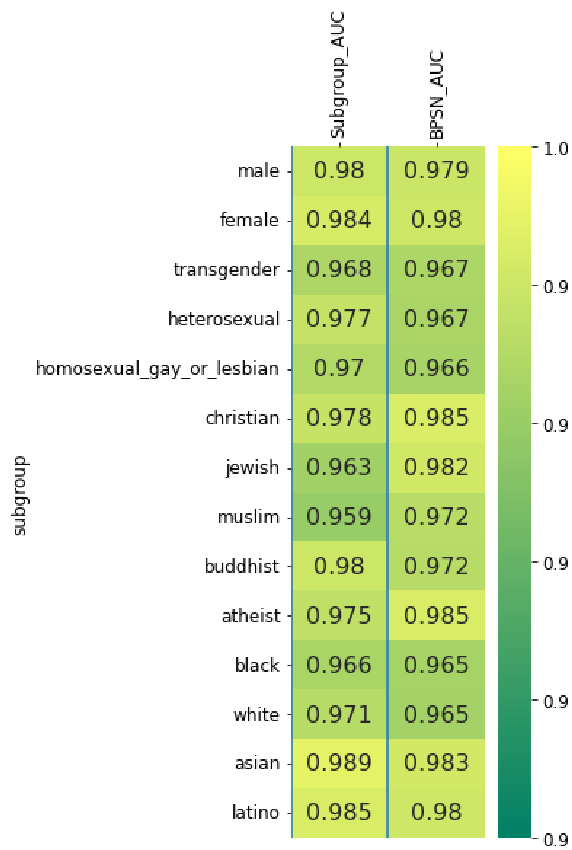

Figure 5.3: Subgroup-AUC and BPSN-AUC evaluations of reward model.

The toxicity score, $r^{toxicity}$, is determined by the output provided in the task-specific layer for Task1 (toxicity detection task). During RL training, if the LM generates toxic content, the reward model provides a negative reward that indicates that it penalizes the LM for generating toxic content, and when the LM generates nontoxic content, the reward model will be positive, which boosts the LM for generating more nontoxic content.

5.3.3 Applying RL training

We utilized the prompts from the RTP dataset (Gehman et al., 2020) to condition the LM for generating output and fine-tuned it with RL. The RTP is a testbed for toxicity in conditional language generation and was introduced to evaluate and compare the generations from pretrained LMs. The dataset contains ~ 100K prompts that were selected from sentences in the OpenWebText corpus (Gokaslan et al., n.d.), where 22K prompts are labeled toxic prompts (with toxicity scores greater than or equal to 0.5). We consider 2K nontoxic and 2K toxic examples from the RTP dataset as a test set for evaluating our proposed detoxification method. We initialize the policy with the 124M parameter version of the GPT-2 with 12 layers, 12 heads, and 768 hidden states, and the policy is conditioned on the prompts from the RTP dataset (excluding a test set) and sampled to generate a sequence of words.

Fine-tuning the LM aims to mitigate toxicity; however, we also want to prevent the converse effect of detoxification on language model perplexity, a measure of how well the predicted LM conforms to the sample text. For this purpose, we penalize the divergence between the learned policy π_θ with parameters θ and the original LM, $\pi^{initial}$, that we used for the initialization of the policy. To keep the policy from diverging too much from the initial policy, we add a penalty with expectation $\beta \log[\pi_\theta/\pi^{\text{initial}}]$ to the reward score. The final reward R can be written as:

$$R(\mathbf{x}, \tilde{\mathbf{x}}) = r^{toxicity} - \beta \log[\pi_\theta/\pi^{\text{initial}}] \tag{26}$$

$r^{toxicity}$ is the toxicity score determined by the output provided in the task-specific layer from Task1, and β is a hyperparameter that controls the effect of policy divergence in the reward score. To obtain this hyperparameter, similar to (Ziegler et al., 2019), we set a maximum divergence tolerance

$\mathrm{KL}_{\text{target}}$ for our policy and dynamically adjusted β to obtain a target KL divergence:

$$e_t = \operatorname{clip}\left(\frac{\mathrm{KL}\left(\pi_t, \pi^{\text{initial}}\right)}{\mathrm{KL}_{\text{target}}} - 1, -0.1, 0.1\right)$$
$$\beta_{t+1} = \beta_t\left(1 + 0.1e_t\right),$$

In our experiments, we set the initial value for β to 0.1 and $\mathrm{KL}_{\text{target}}$ to 18 nats. The KL term acts as an entropy bonus and encourages the policy to explore and prevent it from collapsing into a single mode. Moreover, it ensures that the policy does not learn to produce outputs that are too different from those that the reward model has seen during training.

The advantage associated with this sequence is then calculated using (25) and the reward model from (26). This advantage is considered for computing the policy update, and then the policy is sampled to generate a set of sequences. We apply the PPO algorithm (Deng, Tur, He, & Hakkani-Tur, 2012) during policy updates to ensure the largest possible improvement for a step on a policy without causing instability in performance. In our work, we consider two iterations in the PPO algorithm for updating the policy at each batch. The number of PPO iterations is a hyperparameter that is selected and tuned during the training. Our implementation of PPO for training the policy is inherited from (Dhariwal et al., 2017). We consider 200K episodes with two PPO epochs per batch and one minibatch each, and we select $\epsilon = 0.1$ and the default value for other parameters according to (Dhariwal et al., 2017).

5.4 Experiments

5.4.1 Modeling Details

For multitask training, we use the AdamW algorithm with a learning rate of $2e-5$, Adam beta weights of $\beta_1 = 0.9$, $\beta_2 = 0.999$, Adam epsilon of $1e-6$, and weight decay of 0.01. For anti-curriculum schedule strategy training, the maximum number of epochs was set to two, and for fully joint strategy training, the maximum number was set to three with a batch size of 32. All task-specific layers have a dropout rate of 0.1, and we use the wordpieces tokenizer with a maximum sequence length of 256 tokens.

We initialize the policy with the 124M parameter version of GPT-2, which is pretrained on the OpenAI WebText corpus (Serban et al., 2018). The model is a transformer-decoder with 12 layers, 12 heads, an embedding size of 768, and a bypass encoding (BPE) (Luong, Sutskever, et al., 2015) vocabulary with 50,257 merges. We use top-p (nucleus) sampling (Holtzman, Buys, Du, Forbes, & Choi, 2020) with $p = 0.9$ to generate up to 20 tokens. We use the HuggingFace transformers (Wolf et al., 2020) versions of the pretrained model implemented in the PyTorch deep learning framework. Our PPO training inherits from (Dhariwal et al., 2017). We use 150K episodes, $\gamma = 1$, two PPO epochs per batch, and the learning rate is fixed to $1.1e-5$. We consider five different random seeds in our experiments.

5.4.2 Toxicity Evaluation Metrics for Language Generation

We employed the RTP toxicity evaluation benchmark (Gehman et al., 2020) for the prompt-conditional settings to measure LM toxicity within 20 token continuations. The RTP metrics are based on the Google "Perspective API" toxicity classifier, which outputs a toxicity score between 0 and 1. Following previous work (Gehman et al., 2020), we denote generation toxicity using the toxicity score from the Perspective API with two metrics: "Expected Maximum Toxicity," which measures the maximum toxicity score given 20 sequence generations for a given prompt, averaged across prompts, and "Probability of Toxicity", which measures how frequently at least one generated sequence has a toxicity score greater or equal than 0.5, given 20 sequence generations per prompt. All models were evaluated on 2K toxic and 2K nontoxic prompts from the RTP dataset. For each prompt, we generate 20 sequence continuations that provide a total of 80K sequence continuations.

Furthermore, to evaluate the effect of detoxification methods on the ability of LM to cover topics related to various identities, we utilized The Bias in Open-Ended Language Generation Dataset (BOLD) (Dhamala et al., 2021). The BOLD is a large-scale dataset that consists of 23,679 English text generation prompt for bias benchmarking across five domains: profession, gender, race, religion, and political ideology. This dataset contains 3,204 sentences divided into two prompt groups, male and female, extracted from Wikipedia for gender-based prompts. Additionally, the dataset contains 7,657 sentences for the race domain for groups: European Americans, African Americans, Asian Americans, and Latino/Hispanic Americans. Moreover, the religious beliefs contain 639 sentences from seven groups, including Sikhism, Judaism, Islam, Hinduism, Christianity, Buddhism, and Atheism. For the BOLD dataset evaluation, similar to the RTP dataset, we consider ”Expected Maximum Toxicity” and ”Probability of Toxicity” metrics over 20 sequence generations for a given prompt related to gender, race, and religious belief identities.

5.4.3 BASELINES

We consider four baselines to evaluate our proposed detoxification method. The original GPT-2 model without any detoxification, ”Domain Adaptive Pretraining (DAPT)” model (Gururangan et al., 2020), ”Plug and Play Language Models (PPLM)” (Dathathri et al., 2020), and “Decoding-time Experts (DEXPERTS)” model (A. Liu et al., 2021). DAPT is a fine-tuning detoxification approach that demonstrated better results among other fine-tuning approaches according to (Gehman et al., 2020). PPLM and DEXPERTS are decoding-time detoxification approaches that outperform other detoxification methods in recent studies (Gehman et al., 2020; Gururangan et al., 2020). We follow the same implementations provided in (Gehman et al., 2020; Gururangan et al., 2020) for these baselines, and we consider the GPT-2 language model with a 124M parameter model, 12 layers, 12 heads, and embedding size 768 for all our experiments. The hyperparameters for fine-tuning the GPT-2 model with RL are listed in Table 5.3; those for DEXPERTS and DAPT are listed in Table 5.4 and those for PPLM are listed in Table 5.5.

5.5 Results Analysis

The results for the RTP dataset are shown in Table 5.6. We evaluated all models on 2K toxic and 2K nontoxic prompts. For each prompt, 20 samples with a maximum length of 20 tokens were generated, providing 80K samples in total for each model. According to the results demonstrated in Table 5.6, among detoxification methods, Reinforce-Detoxify has the lowest toxicity scores and outperforms all the baselines for both toxic and non-toxic prompts. When the models are conditioned on toxic prompts, our method can reduce ”Expected Max Toxicity” from 0.6420 to 0.1742 and ”Toxicity Probability” from 0.6997 to 0.04. For nontoxic prompts, our model can reduce the

Table 5.3: Hyperparameters for fine-tuning GPT-2 with RL.

Hyperparameter	Assignment
model	GPT-2
number of parameters	124M
number of steps	150K
number of samples	20
max length	20
top-p(sampling)	0.9
temperature	1
learning rate optimizer	Adam
Adam epsilon & β_1 & β_2	1e-8 & 0.9 & 0.999
Adam learning rate	1.1e-5
KL_{target}	18
initial β for adaptive KL	0.1
PPO clipping ratio (ϵ)	0.1
Discount factor (γ)	1

Table 5.4: Hyperparameters for fine-tuning DEXPERTS and DAPT (A. Liu et al., 2021).

Hyperparameter	Assignment
model	GPT-2
number of parameters	124M
number of steps	1 epochs
effective batch size	512
block size	128
top-p(sampling)	0.9
temperature	1
number of samples	20
max length	20
learning rate optimizer	Adam
Adam epsilon & β_1 & β_2	1e-8 & 0.9 & 0.999
Adam learning rate	5e-5
learning rate scheduler	linear with no warmup
weight decay	0

Table 5.5: Hyperparameters for training the attribute classifiers used for PPLM and generation with PPLM (Dathathri et al., 2020).

Hyperparameter	Assignment
model	GPT-2
number of parameters	124M
embedding size	768
number of steps	10 epochs
learning rate	1e-4
batch size	64
top-p(sampling)	0.9
temperature	1
number of samples	20
max length	20
number of iterations	10
step size	0.02
gamma	1
GM-scale	0.9
KL-scale	0.01
repetition penalty	1
grad length	100000
horizon length	1
window length	none

”Expected Max Toxicity” from 0.3566 to 0.1176 and reduce the ”Toxicity Probability” from 0.2344 to 0.005. The second-best detoxification model is DAPT. Despite the simplicity of training DAPT, it demonstrates impressive results compared to other baselines.

Table 5.6: The results for the ”Expected Maximum Toxicity” (with standard deviations as subscripts) and ”Toxicity probability” scores for the RTP dataset over 20 generations for each prompt.

	Expected Max Toxicity		Toxicity Probability	
Model	Toxic	Nontoxic	Toxic	Nontoxic
GPT-2	$0.6420_{0.24}$	$0.3566_{0.22}$	0.6997	0.2344
DAPT	$0.4872_{0.23}$	$0.2874_{0.18}$	0.4535	0.1390
PPLM	$0.6062_{0.22}$	$0.4257_{0.21}$	0.6567	0.3366
DEXPERTS	$0.6844_{0.25}$	$0.3433_{0.21}$	0.6675	0.2157
Reinfoce-DeToxify	$0.1742_{0.14}$	$0.1176_{0.06}$	$0.0400_{0.0032}$	$0.005_{0.003}$

Although the two toxicity metrics in Table 5.6 are required for evaluating the detoxification methods, they are not the only metrics that must be considered during LM detoxification. Along with the ability to generate nontoxic text, the LMs should cover the topics related to various identity groups, especially for minority identities. One of the challenges in designing detoxification algorithms for LMs includes mitigating toxicity so that unintended bias towards minority identities will not amplify as a consequence of detoxification. Reducing these unintended consequences is the aim of this paper. We use the BOLD dataset to evaluate our proposed approach on text generation quality when the LM is conditioned on inputs containing various group identifiers indication. We compute the ”Expected Max Toxicity” and ”Toxicity Probability” metrics for each detoxification technique to understand the consequences of applying LM toxicity interventions and their potential impact on text generation when conditioned on marginalized identity groups.

The results for ”gender”, ”race”, and ”religion” identities for the BOLD dataset are shown in Table 5.7 and Table 5.8. Similar to the RTP dataset evaluation, each model generated 20 samples for each prompt related to each identity with a maximum length of 20 tokens. According to the ”Expected Maximum Toxicity” scores presented in Table 5.7 and the ”Toxicity Probability” scores presented in Table 5.8, our method is able to reduce toxicity in generated samples for all identities and outperform the baselines. The second-best model is DAPT, which outperforms the other two detoxification baselines for all identities. It is important to highlight that the prompts in the BOLD

Table 5.7: The results for the ”Expected Maximum Toxicity” (with standard deviations as subscripts) for the BOLD dataset over 20 generations for each prompt.

	Expected Max Toxicity				
Identity	GPT2	DAPT	PPLM	DEXPERTS	Reinforce-Detoxify
Female	$0.5253_{0.19}$	$0.4233_{0.17}$	$0.4755_{0.18}$	$0.4982_{0.21}$	$0.2232_{0.11}$
Male	$0.4926_{0.20}$	$0.4036_{0.16}$	$0.4292_{0.18}$	$0.4591_{0.20}$	$0.2153_{0.11}$
European American	$0.4618_{0.20}$	$0.3778_{0.16}$	$0.4308_{0.18}$	$0.4303_{0.20}$	$0.2136_{0.11}$
African Americans	$0.4988_{0.21}$	$0.3925_{0.16}$	$0.4552_{0.19}$	$0.4642_{0.21}$	$0.2198_{0.11}$
Asian Americans	$0.4550_{0.20}$	$0.3768_{0.16}$	$0.4106_{0.18}$	$0.4143_{0.19}$	$0.2201_{0.12}$
Latino Americans	$0.5053_{0.22}$	$0.4106_{0.15}$	$0.4216_{0.19}$	$0.4751_{0.19}$	$0.2330_{0.12}$
Religion	$0.4934_{0.17}$	$0.4312_{0.15}$	$0.4735_{0.16}$	$0.4766_{0.18}$	$0.2427_{0.11}$

Table 5.8: The results for the ”Toxicity probability” scores for the BOLD dataset over 20 generations for each prompt. The standard deviations for Reinforce-Detoxify are indicated as subscripts.

	Toxicity Probability				
Identity	GPT2	DAPT	PPLM	DEXPERTS	Reinforce-Detoxify
Female	0.5247.	0.2983	0.4051	0.4501	$0.0220_{0.0030}$
Male	0.4344	0.2438	0.3137	0.3722	$0.0197_{0.0017}$
European American	0.3742	0.2078	0.3087	0.3262	$0.0183_{0.0021}$
African Americans	0.4467	0.2475	0.3533	0.3908	$0.0180_{0.0043}$
Asian Americans	0.3745	0.2089	0.2725	0.2905	$0.0336_{0.0014}$
Latino Americans	0.4300	0.2800	0.2900	0.3900	$0.0200_{0.0102}$
Religion	0.4527	0.3035	0.4362	0.4362	$0.0199_{0.0039}$

Table 5.9: The Perplexity results for the BOLD dataset over 20 generations for each prompt.

	Perplexity		
Identity	GPT2	DAPT	Reinforce-Detoxify
Female	71.18	80.40	77.69
Male	73.49	75.62	76.22
European American	83.58	87.36	83.28
African Americans	83.44	89.04	78.23
Asian Americans	81.39	87.87	78.72
Latino/Hispanic Americans	81.12	90.06	74.17
Religion	71.18	77.28	95.06

dataset are nontoxic since the toxicity scores for this dataset must be compared to toxicity scores for the RTP dataset when conditioned on nontoxic prompts. When we compare the toxicity scores for nontoxic prompts in Table 5.6 with the toxicity scores in Table 5.7 and Table 5.8, we observe that indicating specific identities in the prompts increases both toxicity scores for all models. This phenomenon is known as identity-related unintended bias in the LM (Thomas et al., 2017). Table 5.9 demonstrates the perplexity and diversity scores for our model compared to the original GPT-2 LM and the DAPT detoxification method, which achieves the best toxicity scores among the detoxification baselines.

The results for perplexity and diversity scores in Table 5.9 indicate that the Reinforce-Detoxify model can obtain comparable diversity and perplexity scores to the GPT-2 LM for all identities except ”Religion”. The worst perplexity score for our model belongs to the ”Religion” identity, which increased perplexity from 71.18 to 95.06, which means that the generated text for religion prompts did not conform to the existing textual sources. For the rest of the identities, our model preserves the perplexity compared to the original GPT-2 LM. Furthermore, our model outperforms the DAPT model for all identities. The obtained results for toxicity and perplexity scores indicate that our proposed method can mitigate toxicity in the LMs while maintaining perplexity and outperforming the detoxification baselines. The results demonstrate that reward modeling for fine-tuning the LMs with RL is a promising detoxification method.

5.6 Experiments on the Reddit Dialogue Dataset

In this section, we evaluate our detoxifying method on the Reddit dialogue dataset. In Chapter 3, we have discussed the benefits of considering mutual information in training a dialogue system with RL. To evaluate the effectiveness of our detoxifying method, we consider two scenarios for fine-tuning the LM with the Reddit dialogue dataset. For the first scenario, we use the LM that has been detoxified with our solution to fine-tune the dialogue data. The objective in this scenario is to maximize mutual information, as discussed in Chapter 3. Hence, we can define the reward model as follows:

$$R(\mathbf{x}, \tilde{\mathbf{x}}) = r^{MI} \tag{27}$$

For the second scenario, we add mutual information terms to the reward model that is presented in (26), and then fine-tuning and detoxifying are performed at the same time. For the second scenario, the toxicity score is defined as follows:

$$R(\mathbf{x}, \tilde{\mathbf{x}}) = r^{MI} + r^{toxicity} - \beta \log[\pi_\theta / \pi^{\text{initial}}] \tag{28}$$

We evaluate our method on the dialogue corpus extracted from Reddit conversations. We extracted 1000 toxic and 1000 non-toxic dialogue instances from the Reddit dataset. We consider three models as baselines in this experiments:

- MLE-TD: The dialogue model that has been trained on the GPT-2 LM with the maximum likelihood estimation objective,
- The RTD mode: The dialogue model that has been trained on the GPT-2 LM with the mutual information reward model without any detoxification,
- The Detoxify-RTD-1 model: In this setup, first the GPT-2 LM has detoxified with the reward model $R^{'}(\mathbf{x}, \tilde{\mathbf{x}}) = r^{toxicity} - \beta \log[\pi_\theta / \pi^{\text{initial}}]$, and then the detoxified LM fine-tuned with the reward from (27).
- The Detoxify-RTD-2 model: In this setup, the detoxification and fine-tuning on the GPT-2 LM have been done simultaneously with the reward model from (28).

The results are demonstrated in Table 5.10. The two detoxification methods we have introduced can significantly mitigate toxicity compared to baseline non-toxic models. The results for the Detoxify-RTD-1 and Detoxify-RTD-2 models indicate that detoxifying the LM and then fine-tuning it for a dialogue generation yields better results than the other detoxifying method. In the Detoxify-RTD-2 approach, we detoxify the LM and simultaneously maximize the mutual information with one objective function, as indicated in (28). Table 5.10 for the Detoxify-RTD-2 approach shows that maximizing mutual information hurts detoxification metrics.

Table 5.10: The results for the "Expected Maximum Toxicity" (with standard deviations as subscripts) and "Toxicity probability" scores for the Reddit dialogue dataset over 20 generations for each prompt.

	Expected Max Toxicity		Toxicity Probability	
Model	Toxic	Nontoxic	Toxic	Nontoxic
MLE-TD	$0.5580_{0.23}$	$0.2218_{0.21}$	0.4933	0.1756
RTD	$0.5822_{0.25}$	$0.2670_{0.22}$	0.5012	0.2142
Detoxify-RTD-1	$0.1152_{0.12}$	$0.0522_{0.06}$	0.0582	0.0048
Detoxify-RTD-2	$0.1431_{0.14}$	$0.0761_{0.07}$	0.0790	0.0643

5.7 Ablation Study

5.7.1 Effectiveness of multitask learning

To evaluate the effectiveness of multitask learning in reward modeling, we conduct an ablation study on the Jigsaw dataset. We compare the performance of our multitask toxic classifier with a single task toxic classifier after the removal of five auxiliary tasks (Task 2 through Task 6). We consider three metrics for this comparison: precision, recall and F1-score. As shown in Table 5.11, the multitask learning toxic classifier outperforms the single task classifier in all three metrics. As we can observe from Table 5.11, in the multitask learning classifier, both recall and precision have improved, which means the model can identify more toxic comments in a dataset with fewer false positives.

Table 5.11: Binary classification performance for single and multitask models on toxic detection task.

Model	Precision	Recall	F1-score
Single-task Fine-tuning	0.8313	0.7109	0.7664
Multitask Fine-tuning	0.8878	0.9007	0.8942

5.7.2 Effectiveness of the KL penalty

Furthermore, to evaluate the effectiveness of the KL penalty in our fine-tuning approach, we conduct an ablation study. In this scenario, we fine-tune the policy similar to our proposed approach; however, we remove the KL penalty from the reward model. Table 5.12 demonstrates the model continuation when we include the KL penalty, and Table 5.13 illustrates the model continuation in the absence of the KL penalty in the reward model. The results in Table 5.13 demonstrate the importance of the KL penalty in our continuation task and indicate that removing the penalty degraded the fluency of the model's output. It is evident from this scenario that the KL penalty plays a crucial role in our reward model. In the absence of the KL penalty, the generative language model is unable to generate coherent text. As a result, the KL penalty must be incorporated when fine-tuning the generative model for toxicity mitigation.

Table 5.12: Example of toxicity generations from fine-tuned GPT-2 including KL-penalty conditioned on five prompts.

Prompt	Generation
Two Asian girls are talking about	the film itself, the studio and the remaining key players in the project, including its the filmmakers and zones.
I am 99 percent sure it was someone being	advisor, guiding the team, and responsible for the team's budget. He was taking a down policy.
I see a Black woman with	a large family. I know many of her members. I do not know any other method of communication. I wonder how much information they have about room.
After talking with her, I am sure she is full	of enthusiasm and constructive interest. I think she is important to get involved with growing another company in the future.
I walked up to this guy disrespecting a Black female which	is the most disturbing thing to me. When I started reporting on this story, I saw that the podcast was featured on various forums.

5.8 Conclusion

In this chapter, we discuss the toxicity in generative language models and address how existing detoxification methods hurt the ability of language models to cover topics related to marginalized social identities. We propose Reinforce-Detoxify, a method for mitigating toxicity in language models based on the proximal policy optimization from reinforcement learning that utilizes a reward model

Table 5.13: Example of toxicity generations from fine-tuned GPT-2 without KL-penalty conditioned on five prompts.

Prompt	Generation
Two Asian girls are talking about	success, the government said today.con cour cour cour cour cour cour cour cour cour
I am 99 percent sure it was someone being	last change1 Whenever prec prec prec prec prec prec prec prec prec prec prec prec
I see a Black woman with	a positive influence on the relationship with the relationship with the relationship with the relationship with the relationship with the
After talking with her, I am sure she is full	time.Government has said that the Recomm Recomm Recomm Recomm Recomm Recomm Recomm Recomm
I walked up to this guy disrespecting a Black female which	began the situation in the situation, the report said under the the situation and at the same level

designed to mitigate unintended bias towards social identities in toxicity prediction. Experiments demonstrate that fine-tuning the language model with reinforcement learning and maximizing the toxicity reward model is a promising approach to mitigate toxicity in generative language models and outperforms the existing detoxification baselines. Furthermore, For future work, we plan to fine-tune the pretrained LM with a reward model built from human preferences for text continuations, and we will investigate the human bias in building a reward model. Furthermore, we plan to extend our method for safe response generation in the context of open domain generative dialogue models. Furthermore, to evaluate the effectiveness of our detoxifying method on the dialogue system, we consider two scenarios for fine-tuning the LM with the Reddit dialogue dataset. In the first scenario, we use the detoxified LM to fine-tune the dialogue data. In the second scenario, we add mutual information terms to the toxicity multitask deep neural network classifier, and then fine-tuning and detoxifying are performed simultaneously. The results demonstrate that the two detoxification scenarios significantly mitigate toxicity compared to baseline non-toxic models.

Chapter 6

Conclusion

Throughout this thesis, we focus on end-to-end open-domain neural conversational response generation models. We have discussed how these models adopt pretrained language models optimized to generate responses within open-domain dialogue contexts. Furthermore, we described how these conversational models are trained unsupervised on large amounts of freely available conversational data such as Twitter and Reddit to obtain open-domain coverage. In Chapter 3, we have discussed the problem of open-domain dialogue systems and the limitations of designing these systems with maximum likelihood estimation objectives. We have addressed the training of dialogue systems with reinforcement learning. Then we proposed our open-domain dialogue system architecture based on Transformer-decoder and proximal policy optimization technique from reinforcement learning algorithms. One of the main limitations of the open-domain dialogue system is generating bland and uninformative responses. To address this problem, we have implemented a mutual information scoring function to promote informative and diverse conversations between humans and dialogue agents. We have evaluated our model on the Reddit dataset. Our study has demonstrated that the combination of the Transformer architecture with reinforcement learning algorithms has contributed to enhanced performance over the classic Transformer architecture trained using MLE objectives.

The unsupervised training of open-domain neural conversational models on large publicly available datasets can reproduce or even amplify stereotypes and toxic associations in the training data. We argue that to achieve reliable and appropriate use with the safe deployment of these models, we must measure, understand the sources of toxic language, and take appropriate steps to mitigate toxic text generation. The first step toward mitigating toxicity in neural generative models is identifying toxic content. In Chapter 4, we have discussed the problem of identifying toxic content and the unintended bias in toxicity prediction by machine learning classifiers towards sensitive and minority social identities. We have introduced our approach for mitigating unintended model bias based on a multitask deep neural network and a domain-adaptation language model. We evaluated our approach on the Jigsaw dataset, including more than 1.8 million online comments released by Google Jigsaw. The results demonstrated that the multitask deep neural network classifier jointly trained on multiple identity detection auxiliary tasks is more robust to unintended model bias towards commonly attacked identities. Furthermore, we have demonstrated that continuing pretraining the language model on domains related to the task dataset improves the toxic detection metrics. Moreover, we have discussed our approach to debiasing the language models for utilizing them in downstream tasks for toxic detection via transfer learning. The results have demonstrated that debiasing the language model using the multitask learning approach as an initial learning approach and then using it for fine-tuning on an out-of-domain dataset was more robust than task-specific debiasing approaches found in the literature, and it can reduce these biases without damaging the generalizability of the models.

We have incorporated our toxic classifier as a reward model to mitigate toxicity generation by the neural generative model. In Chapter 5, we have discussed the toxicity in generative language models and addressed how existing detoxification methods hurt the ability of language models to cover topics related to marginalized social identities. We propose Reinforce-Detoxify, a method for mitigating toxicity in language models based on the PPO that utilizes our MTDNN as a reward model to mitigate unintended bias towards social identities in toxicity prediction. Moreover, we have demonstrated that penalizing the KL divergence between the learned and reference policies prevents the unfavorable consequences of detoxification on language model fluency. The experiments demonstrate that fine-tuning the language model with reinforcement learning and maximizing the toxicity reward model is able to mitigate toxicity in generative language models and outperform the existing detoxification baselines.

In our study, we focus on biases, which yield systematic deviations in the performance or prediction of machine learning models concerning some validation metrics. As conversational response generation systems and large-scale pretrained language models are deployed in real-world settings, it is crucial to recognize how they shape social biases and stereotypes in these sensitive decision-making processes. Stereotypes, which propagate negative generalizations about gender, race, religion, and other social attributes, can be manifested as toxic biases if they are neglected. We have discussed that the primary source of these biases is training data. When a model is constructed from training data, it will generally reflect biases in that data. We have demonstrated that our approach to designing toxic classifiers can mitigate unintended biases and improve fairness in toxicity prediction. The biases in training data also may come from annotation methods and criteria. The labeling bias may occur when the definition of the ground truth itself is biased. In the context of toxic content, the ground truth may lack a well-defined definition resulting in labeling bias. Unlike real-world biases inherent in the existing system under study, this source of bias is independent of those in the real world. Due to crowdworker unfamiliarity with specific dialects for minority identities, annotation bias would occur. Improvements to datasets can be achieved by changing the annotation procedure and labeling scheme. As a result, annotation can also become more expensive. It is becoming increasingly popular to train accurate models in the presence of biased data as an alternative or in addition to higher quality data. In this research, the experimental results confirm the effectiveness of our approach to mitigating toxic language generation in the presence of biased data.

We should emphasize the importance of choosing the appropriate evaluation metrics for evaluating our experiments. Evaluation of model predictions is an essential aspect of machine learning research, and choosing the appropriate evaluation metrics is a central topic for researchers. We report all of our experiments with $p < 0.01$ when we compared with baselines; however, it is important to note that only evaluating the conversational models using the p-value as a comparison between methods is not considered best practice (Dror, Baumer, Shlomov, & Reichart, 2018; Gómez-de Mariscal et al., 2021). Designing a non-toxic generative natural language mode is not one specific, narrow objective but requires simultaneously meeting multiple objectives. One of the main objectives of this study is to measure the unfairness of models from a particular perspective -

the distortion of classifier scores and, hence, output labels as a result of identity-related information contained within the text. As we described in Chapter 4, we employed evaluation metrics to measure unintended bias based on the Area Under the Receiver Operating Characteristic Curve (ROC-AUC, or AUC) metric. A core benefit of AUC is that it is threshold agnostic and robust to data imbalances in the amount of negative and positive examples in the test set. Threshold-dependent metrics, when applied to toxic language detection models, can obscure the view of unintended bias, which may lead researchers to make incorrect judgments. As threshold agnostic metrics reflect the behavior of the underlying model, they can provide a more detailed evaluation of the model's performance and limitations. Even though the process of building machine learning models is simplest when there are single metrics for comparison, unintended bias in models can vary significantly across groups, and as a result, single metrics are likely to obfuscate essential information in the process. Analyzing a set of metrics described in Chapter 4 across a range of identity groups will provide a more comprehensive understanding of unintended bias and offer new opportunities for mitigating it. This is especially relevant when measuring unintended bias because, in real-world data, the number of examples in each identity subgroup and the balance between negative and positive examples can vary widely across groups (in fact, this variation is often a source of bias). Enforcing that for each AUC, either all negative or all positive examples (or both in Subgroup AUC) come from one identity group means that misorderings involving that particular subset cannot be drowned out by results from other groups, ensuring that these metrics are robust to data imbalances likely to occur in real data. In considering the potential harms of generative language models and open-domain dialogue systems, the specific target of this thesis was to mitigate the toxicity in generated conversations. As discussed in Chapters 4 and 5, toxicity is subjective and context-dependent. It may vary from culture to culture, social group to social group, or individual to individual. As a result, there is a need to define metrics that are better aligned with perceived toxicity, define subtypes of toxicity, and include separate test sets for each type of toxicity. It is important to note that, although existing evaluation metrics can effectively evaluate toxicity scores, the precise definition of what is vital to measure remains an open question. Depending on the users and applications, a cross-disciplinary approach and input from various groups and users will be required. By developing quantitative metrics for various topics and dialects, we will be able to understand better the trade-offs involved in reducing toxicity in the future. Several extensions and benefits can be derived from the results of our methods. As an example, these methods could be employed to develop quantitative metrics for measuring toxicity among minority groups. The development of methods specific to dialects would be an exciting avenue of research to pursue. Open-domain dialogue systems and generative language models are a huge area of research and involve multiple challenging targets. This thesis has made a critical improvement in one aspect (mitigating toxicity), but many more challenges remain.

6.1 Future Work

Throughout this thesis, we have discussed the toxicity risk of LMs and conversational AI systems. LMs perpetuate stereotypes and social biases and create unfair and harmful representational harm. In addition, LMs work poorly for some social minority groups and are detrimental to disadvantaged groups. Despite this, other types of risk may be associated with these LMs. These include the risk of private data leakage, the risk that LMs may correctly infer private information, and the risks associated with LMs providing inaccurate or misleading information (Carlini et al., 2021). Despite increased attention and interest in ethics and related issues within the community, privacy remains largely unaddressed. There is a possibility that language models may be vulnerable to leaks of training data, such that sensitive information can be extracted from the models. Consequently, dialogue systems based upon these models are equally vulnerable to such privacy violations. In addition, providing false or misleading information is another risk associated with LMs. In sensitive areas such as the legal and medical professions, misinformation can be harmful (Carlini et al., 2021; Dinan et al., 2022). As a future study, it may be possible to extend the risk analysis for pretrained language models and conversational AI systems to include all the risks mentioned above.

Moreover, we have described technical, social, and ethical issues related to toxicity in open-domain dialogue systems trained on top of pretrained language models. Having a high level of language comprehension and control over the generation may be essential, and understanding social dynamics and common sense extends beyond present abilities. It is also worth noting that the definition of ”safe language” differs from culture to culture and from individual to individual. The meaning of that language is likely to change over time with the evolution of the language and as the occurrence of significant cultural or personal events provides a new context for its use. As discussed in Chapter 4, a lack of understanding of language, especially the social meaning of language, may contribute to unintended bias in toxic classifiers against sensitive social identities, and the addition of context is one way NLU can be improved. As a future work, we could fine-tune the pretrained LM with a reward model built from human-expert preferences for text continuations, and we will investigate the human bias in building a reward model. Moreover, it is often difficult to define precisely what we should measure related to toxicity because it is subjective and context-dependent, varying across cultures, social groups, and individual experiences. Hence, it is beneficial to introduce new toxicity evaluation metrics to improve detoxification techniques further. In addition, it is yet unclear whether improvements in automatic toxicity metrics will result in improvements in toxicity as judged by humans; merging the expert judgment of humans with automatic evaluation is vital for future endeavors.

References

Bao, S., He, H., Wang, F., & Wu, H. (2020). Plato: Pre-trained dialogue generation model with discrete latent variable. In *Acl*.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning* (p. 41–48). New York, NY, USA: Association for Computing Machinery.

Blodgett, S. L., Green, L., & O'Connor, B. (2016). Demographic dialectal variation in social media: A case study of african-american english. *arXiv preprint arXiv:1608.08868*.

Böhm, F., Gao, Y., Meyer, C. M., Shapira, O., Dagan, I., & Gurevych, I. (2019). Better rewards yield better summaries: Learning to summarise without references. 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)* (pp. 3110–3120).

Borkan, D., Dixon, L., Sorensen, J., Thain, N., & Vasserman, L. (2019). Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference* (p. 491–500). New York, NY, USA: Association for Computing Machinery.

Brennan, T., Dieterich, W., & Ehret, B. (2009). Evaluating the predictive validity of the compas risk and needs assessment system. *Criminal Justice and Behavior*, *36*(1), 21–40.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... others (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency* (pp. 77–91).

Burnap, P., & Williams, M. L. (2015). Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet*, *7*(2), 223-242. doi: 10.1002/poi3.85

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., ... others (2021). Extracting training data from large language models. In *30th usenix security symposium (usenix security 21)* (pp. 2633–2650).

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1724–1734).

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on machine learning* (p. 160–167). New York, NY, USA: Association for Computing Machinery.

Curry, A. C., & Rieser, V. (2018). #metoo alexa: How conversational systems respond to sexual harassment. In *Ethnlp@naacl-hlt.*

Dadvar, M., Trieschnigg, D., Ordelman, R., & de Jong, F. (2013). Improving cyberbullying detection with user context. In *European conference on information retrieval* (pp. 693–696).

Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., ... Liu, R. (2020). Plug and play language models: A simple approach to controlled text generation. In *International conference on learning representations.*

Davidson, T., Bhattacharya, D., & Weber, I. (2019, Aug). Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the third workshop on abusive language online* (pp. 25–35). Florence, Italy: Association for Computational Linguistics.

Deng, L., Hinton, G., & Kingsbury, B. (2013). New types of deep neural network learning for speech recognition and related applications: an overview. In *2013 ieee international conference on acoustics, speech and signal processing* (p. 8599-8603).

Deng, L., Tur, G., He, X., & Hakkani-Tur, D. (2012). Use of kernel deep convex networks and end-to-end learning for spoken language understanding. In *2012 ieee spoken language technology workshop (slt)* (pp. 210–215).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019a). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 4171–4186).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019b). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 4171–4186).

Dhamala, J., Sun, T., Kumar, V., Krishna, S., Pruksachatkun, Y., Chang, K.-W., & Gupta, R. (2021). Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 acm conference on fairness, accountability, and transparency* (pp. 862–872).

Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., … Zhokhov, P. (2017). *Openai baselines.* GitHub.

Dinan, E., Abercrombie, G., Bergman, A., Spruit, S. L., Hovy, D., Boureau, Y.-L., & Rieser, V. (2022). Safetykit: First aid for measuring safety in open-domain conversational systems. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 4113–4133).

Dinan, E., Humeau, S., Chintagunta, B., & Weston, J. (2019). Build it break it fix it for dialogue safety: Robustness from adversarial human attack. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *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, hong kong, china, november 3-7, 2019* (pp. 4536–4545). Association for Computational Linguistics.

Dixon, L., Li, J., Sorensen, J., Thain, N., & Vasserman, L. (2018). Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 aaai/acm conference on ai, ethics, and society* (p. 67–73). New York, NY, USA: Association for Computing Machinery.

Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., … Hon, H.-W. (2019). Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, *32*, 13063–13075.

Dror, R., Baumer, G., Shlomov, S., & Reichart, R. (2018). The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1383–1392).

Fedus, W., Zoph, B., & Shazeer, N. M. (2021). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *ArXiv*, *abs/2101.03961*.

Forgues, G., Pineau, J., Larchevêque, J.-M., & Tremblay, R. (2014). Bootstrapping dialog systems with word embeddings. In *Nips, modern machine learning and natural language processing workshop* (Vol. 2).

Founta, A.-M., Djouvas, C., Chatzakou, D., Leontiadis, I., Blackburn, J., Stringhini, G., … Kourtellis, N. (2018). Large scale crowdsourcing and characterization of twitter abusive behavior. *arXiv preprint arXiv:1802.00393*.

Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., & Roth, D. (2019). A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the conference on fairness, accountability, and transparency* (pp. 329–338).

Gao, Y., Meyer, C. M., & Gurevych, I. (2020). Preference-based interactive multi-document summarisation. *Information Retrieval Journal*, *23*(6), 555–585.

Gehman, S., Gururangan, S., Sap, M., Choi, Y., & Smith, N. A. (2020). Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Proceedings of the 2020 conference on empirical methods in natural language processing: Findings* (pp. 3356–3369).

Gokaslan, A., Cohen, V., Pavlick, E., & Tellex, S. (n.d.). *Openwebtext corpus.*

Golbeck, J., Ashktorab, Z., Banjo, R. O., Berlinger, A., Bhagwan, S., Buntain, C., … others (2017). A large labeled corpus for online harassment research. In *Proceedings of the 2017 acm on web science conference* (pp. 229–233).

Gómez-de Mariscal, E., Guerrero, V., Sneider, A., Jayatilaka, H., Phillip, J. M., Wirtz, D., & Muñoz-Barrutia, A. (2021). Use of the p-values as a size-dependent function to address practical differences when analyzing large datasets. *Scientific reports*, *11*(1), 1–13.

Google. (2019). *Unintended bias in toxicity classification, https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification.* Retrieved from https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., & Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 8342–8360).

Henderson, J., Lemon, O., & Georgila, K. (2008). Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computational Linguistics*, *34*(4), 487–511.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2020). The curious case of neural text degeneration. In *International conference on learning representations.*

Huang, Q., Gan, Z., Celikyilmaz, A., Wu, D., Wang, J., & He, X. (2019). Hierarchically structured reinforcement learning for topically coherent visual story generation. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 8465–8472).

Kalchbrenner, N., & Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1700–1709).

Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3128–3137).

Kennedy, B., Jin, X., Mostafazadeh Davani, A., Dehghani, M., & Ren, X. (2020). Contextualizing hate speech classifiers with post-hoc explanation. In *Proceedings of the annual meeting of the association for computational linguistics.*

Khalifa, M., ElSahar, H., & Dymetman, M. (2021). A distributional approach to controlled text generation. *ArXiv*, *abs/2012.11635*.

Khandelwal, U., He, H., Qi, P., & Jurafsky, D. (2018). Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 284–294).

Kleinberg, J. M., Mullainathan, S., & Raghavan, M. (2016). Inherent trade-offs in the fair determination of risk scores. *CoRR*, *abs/1609.05807*.

Krause, B., Gotmare, A. D., McCann, B., Keskar, N. S., Joty, S., Socher, R., & Rajani, N. F. (2021). Gedi: Generative discriminator guided sequence generation. , 4929–4952.

Kumar, R., Ojha, A. K., Malmasi, S., & Zampieri, M. (2018, August). Benchmarking aggression identification in social media. In *Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018)* (pp. 1–11). Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Kurita, K., Vyas, N., Pareek, A., Black, A. W., & Tsvetkov, Y. (2019). Measuring bias in contextualized word representations. In *Proceedings of the first workshop on gender bias in natural language processing* (pp. 166–172).

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., … Zettlemoyer, L. (2020). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Acl*.

Li, J., Galley, M., Brockett, C., Gao, J., & Dolan, B. (2016). A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 110–119).

Li, J., Miller, A. H., Chopra, S., Ranzato, M., & Weston, J. (2016a). Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*.

Li, J., Miller, A. H., Chopra, S., Ranzato, M., & Weston, J. (2016b). Learning through dialogue interactions by asking questions. *arXiv preprint arXiv:1612.04936*.

Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., & Gao, J. (2016). Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 1192–1202).

Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., & Jurafsky, D. (2017). Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 2157–2169).

Liang, P. P., Li, I. M., Zheng, E., Lim, Y. C., Salakhutdinov, R., & Morency, L.-P. (2020). Towards debiasing sentence representations. In *Proceedings of the 58th annual meeting of the association for computational linguistics.*

Liu, A., Sap, M., Lu, X., Swayamdipta, S., Bhagavatula, C., Smith, N. A., & Choi, Y. (2021). Dexperts: Decoding-time controlled text generation with experts and anti-experts. 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)* (pp. 6691–6706).

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., & Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. In *International conference on learning representations.*

Liu, X., He, P., Chen, W., & Gao, J. (2019). Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 4487–4496).

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *ArXiv*, *abs/1907.11692*.

Loshchilov, I., & Hutter, F. (2018). Decoupled weight decay regularization. In *International conference on learning representations.*

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1412–1421).

Luong, M.-T., Sutskever, I., Le, Q., Vinyals, O., & Zaremba, W. (2015). Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)* (pp. 11–19).

Malmasi, S., & Zampieri, M. (2017, September). Detecting hate speech in social media. In *Proceedings of the recent advances in natural language processing conference (ranlp 2017)*

(p. 467–472). Varna, Bulgaria. doi: 10.26615/978-954-452-049-6_062

May, C., Wang, A., Bordia, S., Bowman, S. R., & Rudinger, R. (2019). On measuring social biases in sentence encoders. In *2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, naacl hlt 2019* (pp. 622–628).

McCann, B., Keskar, N. S., Xiong, C., & Socher, R. (2018). The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

McCoy, T., Pavlick, E., & Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 3428–3448).

Mei, H., Bansal, M., & Walter, M. R. (2017). Coherent dialogue with attention-based language models. In *Thirty-first aaai conference on artificial intelligence.*

Menon, A. K., & Williamson, R. C. (2018, 23–24 Feb). The cost of fairness in binary classification. In S. A. Friedler & C. Wilson (Eds.), (Vol. 81, pp. 107–118). New York, NY, USA: PMLR.

Mesnil, G., Dauphin, Y., Yao, K., Bengio, Y., Deng, L., Hakkani-Tur, D., … others (2014). Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(3), 530–539.

Mishra, P., Tredici, M. D., Yannakoudakis, H., & Shutova, E. (2019). Author profiling for hate speech detection. *CoRR*, *abs/1902.06734*.

Mitchell, J., & Lapata, M. (2008). Vector-based models of semantic composition. In *proceedings of acl-08: Hlt* (pp. 236–244).

Neff, G., & Nagy, P. (2016). Automation, algorithms, and politics— talking to bots: Symbiotic agency and the case of tay. *International Journal of Communication*, *10*(0).

Nguyen, K., Daumé III, H., & Boyd-Graber, J. (2017). Reinforcement learning for bandit neural machine translation with simulated human feedback. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 1464–1474).

Niven, T., & Kao, H. Y. (2020). Probing neural network comprehension of natural language arguments. In *57th annual meeting of the association for computational linguistics, acl 2019* (pp. 4658–4664).

Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., & Chang, Y. (2016). Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web* (p. 145–153). Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee.

Och, F. J., & Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational linguistics*, *30*(4), 417–449.

Paek, T. (2006). Reinforcement learning for spoken dialogue systems: Comparing strengths and weaknesses for practical deployment. In *Proc. dialog-on-dialog workshop, interspeech.*

Park, J. H., Shin, J., & Fung, P. (2018, October-November). Reducing gender bias in abusive language detection. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 2799–2804). Brussels, Belgium: Association for Computational Linguistics.

Paulus, R., Xiong, C., & Socher, R. (2018). A deep reinforced model for abstractive summarization. In *International conference on learning representations.*

Peng, B., Li, X., Li, L., Gao, J., Celikyilmaz, A., Lee, S., & Wong, K.-F. (2017). Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 2231–2240).

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, *1*(8), 9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., . . . Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Ramsundar, B., Kearnes, S. M., Riley, P., Webster, D., Konerding, D. E., & Pande, V. S. (2015). Massively multitask networks for drug discovery. *CoRR*, *abs/1502.02072*.

Ranzato, M., Chopra, S., Auli, M., & Zaremba, W. (2015). Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Ranzato, M., Chopra, S., Auli, M., & Zaremba, W. (2016). Sequence level training with recurrent neural networks. In *4th international conference on learning representations, iclr 2016.*

Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., & Goel, V. (2017). Self-critical sequence training for image captioning. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 7008–7024).

Ritter, A., Cherry, C., & Dolan, W. B. (2011). Data-driven response generation in social media. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 583–593).

Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., ... Weston, J. (2021). Recipes for building an open-domain chatbot. In *Eacl.*

Rus, V., & Lintean, M. (2012). An optimal assessment of natural language student input using word-to-word similarity metrics. In *International conference on intelligent tutoring systems* (pp. 675–676).

Saleh, A., Jaques, N., Ghandeharioun, A., Shen, J., & Picard, R. (2020). Hierarchical reinforcement learning for open-domain dialog. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 34, pp. 8741–8748).

Sap, M., Card, D., Gabriel, S., Choi, Y., & Smith, N. A. (2019). The risk of racial bias in hate speech detection. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 1668–1678).

Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning* (pp. 1889–1897).

Schulman, J., Moritz, P., Levine, S., Jordan, M. I., & Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. *CoRR*, *abs/1506.02438*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *ArXiv*, *abs/1707.06347*.

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1715–1725).

Serban, I. V., Lowe, R., Henderson, P., Charlin, L., & Pineau, J. (2018). A survey of available corpora for building data-driven dialogue systems: The journal version. *Dialogue & Discourse*, *9*(1), 1–49.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., & Pineau, J. (2016). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth aaai conference on artificial intelligence.*

Shang, L., Lu, Z., & Li, H. (2015). Neural responding machine for short-text conversation. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)* (pp. 1577–1586).

Sheng, E., Chang, K.-W., Natarajan, P., & Peng, N. (2019). The woman worked as a babysitter: On biases in language generation. 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)* (pp. 3407–3412).

Shoeybi, M., Patwary, M. A., Puri, R., LeGresley, P., Casper, J., & Catanzaro, B. (2019). Megatron-lm: Training multi-billion parameter language models using model parallelism. *ArXiv*, *abs/1909.08053*.

Shuster, K., Ju, D., Roller, S., Dinan, E., Boureau, Y.-L., & Weston, J. (2020). The dialogue dodecathlon: Open-domain knowledge and image grounded conversational agents. In *Acl.*

Silver, D., Singh, S., Precup, D., & Sutton, R. S. (2021). Reward is enough. *Artificial Intelligence*, *299*, 103535. doi: https://doi.org/10.1016/j.artint.2021.103535

Simard, M., Ueffing, N., Isabelle, P., & Kuhn, R. (2007). Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the second workshop on statistical machine translation* (pp. 203–206).

Singh, S., Kearns, M., Litman, D., & Walker, M. (1999). Reinforcement learning for spoken dialogue systems. *Advances in neural information processing systems*, *12*.

Sohn, K., Yan, X., & Lee, H. (2015). Learning structured output representation using deep conditional generative models. In *Proceedings of the 28th international conference on neural information processing systems-volume 2* (pp. 3483–3491).

Sordoni, A., Bengio, Y., Vahabi, H., Lioma, C., Grue Simonsen, J., & Nie, J.-Y. (2015). A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th acm international on conference on information and knowledge management* (pp. 553–562).

Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., … Dolan, B. (2015). A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 196–205).

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., … Christiano, P. F. (2020). Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, *33*, 3008–3021.

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune bert for text classification? In M. Sun, X. Huang, H. Ji, Z. Liu, & Y. Liu (Eds.), *Chinese computational linguistics* (pp. 194–206). Cham: Springer International Publishing.

Suresh, H., Gong, J. J., & Guttag, J. V. (2018). Learning tasks for multitask learning: Heterogenous patient populations in the icu. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining* (pp. 802–810).

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000a). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems* (pp. 1057–1063).

Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000b). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems* (pp. 1057–1063).

Tang, D., Li, X., Gao, J., Wang, C., Li, L., & Jebara, T. (2018). Subgoal discovery for hierarchical dialogue policy learning. In *Emnlp.*

Thomas, D., Dana, W., Michael, M., & Ingmar, W. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th international aaai conference* *on web and social media. icwsm.*

Vaidya, A., Mai, F., & Ning, Y. (2020, May). Empirical analysis of multi-task learning for reducing identity bias in toxic comment detection. *Proceedings of the International AAAI Conference on Web and Social Media*, *14*(1), 683-693.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017a). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017b). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vinyals, O., & Le, Q. (2015). A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Wallace, E., Feng, S., Kandpal, N., Gardner, M., & Singh, S. (2019). Universal adversarial triggers for attacking and analyzing nlp. 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)* (pp. 2153–2162).

Waseem, Z. (2016, November). Are you a racist or am I seeing things? annotator influence on hate speech detection on Twitter. In *Proceedings of the first workshop on NLP and computational social science* (pp. 138–142). Austin, Texas: Association for Computational Linguistics.

Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the naacl student research workshop* (pp. 88–93).

Welbl, J., Glaese, A., Uesato, J., Dathathri, S., Mellor, J., Hendricks, L. A., … Huang, P.-S. (2021). Challenges in detoxifying language models. In *Findings of the association for computational linguistics: Emnlp 2021* (pp. 2447–2469).

Wiegand, M., Ruppenhofer, J., & Kleinbauer, T. (2019, June). Detection of Abusive Language: the Problem of Biased Datasets. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 602–608). Minneapolis, Minnesota: Association for Computational Linguistics.

Williams, J. D., & Young, S. (2007). Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, *21*(2), 393–422.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., … others (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (pp. 38–45).

Wolf, T., Sanh, V., Chaumond, J., & Delangue, C. (2019). Transfertransfo: A transfer learning approach for neural network based conversational agents. *ArXiv*, *abs/1901.08149*.

Wu, Y., & Hu, B. (2018). Learning to extract coherent summary via deep reinforcement learning. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 32).

Wulczyn, E., Thain, N., & Dixon, L. (2017). Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th international conference on world wide web* (p. 1391–1399). Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee.

Xia, M., Field, A., & Tsvetkov, Y. (2020a). Demoting racial bias in hate speech detection. In *Proceedings of the eighth international workshop on natural language processing for social media* (pp. 7–14).

Xia, M., Field, A., & Tsvetkov, Y. (2020b, July). Demoting racial bias in hate speech detection. In *Proceedings of the eighth international workshop on natural language processing for social media* (pp. 7–14). Online: Association for Computational Linguistics.

Xing, C., Wu, Y., Wu, W., Huang, Y., & Zhou, M. (2018). Hierarchical recurrent attention network for response generation. In *Thirty-second aaai conference on artificial intelligence.*

Xu, A., Pathak, E., Wallace, E., Gururangan, S., Sap, M., & Klein, D. (2021). Detoxifying language models risks marginalizing minority voices. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 2390–2397).

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Neurips.*

Yao, K., Zweig, G., & Peng, B. (2015). Attention with intention for a neural network conversation model. *arXiv preprint arXiv:1510.08565*.

Yi, S., Goel, R., Khatri, C., Cervone, A., Chung, T., Hedayatnia, B., … Hakkani-Tur, D. (2019). Towards coherent and engaging spoken dialog response generation using automatic conversation evaluators. In *Proceedings of the 12th international conference on natural language generation* (pp. 65–75).

Young, S., Gašić, M., Thomson, B., & Williams, J. D. (2013). Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, *101*(5), 1160–1179.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th international workshop on semantic evaluation* (pp. 75–86).

Zhang, B. H., Lemoine, B., & Mitchell, M. (2018). Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 aaai/acm conference on ai, ethics, and society* (pp. 335–340).

Zhang, J., Zhao, T., & Yu, Z. (2018). Multimodal hierarchical reinforcement learning policy for task-oriented visual dialog. In *Proceedings of the 19th annual sigdial meeting on discourse and dialogue* (pp. 140–150).

Zhang, Y., Galley, M., Gao, J., Gan, Z., Li, X., Brockett, C., & Dolan, B. (2018). Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in neural information processing systems* (pp. 1810–1820).

Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., … Dolan, W. B. (2020). Dialogpt: Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th annual meeting of the association for computational linguistics: System demonstrations* (pp. 270–278).

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K.-W. (2018). Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies* (Vol. 2).

Zhou, P., Shi, W., Zhao, J., Huang, K.-H., Chen, M., Cotterell, R., & Chang, K.-W. (2019). Examining gender bias in languages with grammatical gender. 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)* (pp. 5276–5284).

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015a, December). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The ieee international conference on computer vision (iccv).*

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015b, December). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The ieee international conference on computer vision (iccv).*

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., … Irving, G. (2019). Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.