Master of Science in Computer Science
January 2021

Reducing Unintended bias in Text Classification using Multitask learning.

Veankata Sai Sukesh Settipalli
Naga Manendra Kumar Dasireddy

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Authors:
Veankata Sai Sukesh Settipalli
E-mail: vese18@student.bth.se

Naga Manendra Kumar Dasireddy
E-mail: nada18@student.bth.se

University advisor:
Lawrence Edward Henesey
Department of Computer Science

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

Background: Recent developments in the field of machine learning and deep learning had motivated many researchers to use them in providing automated solutions for real world problems. One of such use cases is employing deep learning models for detecting toxic comments. But recent research shows that some models tend to exhibit unintended biases(like gender bias, identity bias) due to the imbalances in the training data. Studies were made to reduce this kind of model biases, one of which is using multitask learning to reduce the identity bias in toxicity classification. In this thesis we focus on reducing a special type of bias called identity dais by fine-tuning an attention based model called BERT(Bidirectional Encoder Representation from Transformers) using multitask learning.

Objectives: This thesis is an extension to the work done by Ameya Vaidya, Feng Mai and Yue Ning, who had used multitask learning to train an Bi-LSTM (Bidirectional Long Short Term Memory) model to reduce the identity bias. The main aim of this research is to use multitask learning to fine tune the BERT models namely BERT-base and Distil BERT, So that they can jointly predict the toxicity of the text comment and the presence of identity in that comment in order to reduce bias over frequently attacked identity terms. The proposed BERT models are also compared with the previously proposed Bi-LSTM model in terms of bias mitigation and classification performance.

Methods: First a literature review is conducted to find the metrics suitable for measuring the bias mitigation performance of the models. Then an experiment is conducted to train and test all the three models namely BERT-base, DistilBERT and Bi-LSTM models. Finally the results are compared to know whether BERT models provide any significant improvement in terms of bias mitigation and classification performance.

Results: All the three models are trained and tested during which metrics corresponding to classification and bias mitigation performance are extracted. On comparing the results it is evident that BERT based models offers better classification and bias mitigation performance than the Bi-LSTM model.

Conclusions: We can conclude that multitask learning using BERT is a promising approach for reducing identity bias. It is evident that BERT based models are able to outperform the previously proposed Bi-LSTM model in terms of bias mitigation and classification performance.

Keywords: Toxicity classification, Identity bias, Multitask learning, BERT.

Acknowledgments

First and foremost, we would like to express our gratitude to our supervisor, Professor Lawrence Edward Henesey, we are very thankful to him for providing us all the time, encouragement and inspiring guidance which helped us complete every aspect of this project work. We gratefully forward our affectionate thanks to our family, friends and classmates for their encouragement and support.

Contents

Abstract i

Acknowledgments iii

1 Introduction 1
- 1.1 Problem statement . . . 2
- 1.2 Aim and Objective . . . 3
- 1.3 Research Questions . . . 3
- 1.4 Outline . . . 4

2 Background 5
- 2.1 Natural language Processing . . . 5
- 2.2 Artificial Intelligence . . . 5
- 2.3 Machine Learning . . . 5
- 2.4 Deep Learning . . . 5
- 2.5 Applications of NLP . . . 6
  - 2.5.1 QA(Question Answering) . . . 6
  - 2.5.2 Text Summarization . . . 7
  - 2.5.3 Text Classification . . . 7
  - 2.5.4 Machine Translation . . . 7
- 2.6 Bias in machine learning and deep learning . . . 8
- 2.7 Bias in text classification models . . . 8
- 2.8 Multitask Learning . . . 10
- 2.9 Models used and their architecture: . . . 11
  - 2.9.1 Architecture of LSTM model: . . . 11
  - 2.9.2 Architecture of the BERT model: . . . 12

3 Literature Review 17
- 3.1 Mythology . . . 17
  - 3.1.1 Investigation of primary study: . . . 17
  - 3.1.2 Selection criteria: . . . 18
- 3.2 Error Rate Equality Difference . . . 18
  - 3.2.1 FPED (False-positive equality difference): . . . 19
  - 3.2.2 FNED (False-negative equality difference): . . . 19
- 3.3 Pinned AUC . . . 20
  - 3.3.1 Pinned AUC equality difference P AUCED: . . . 20
- 3.4 Limitations of PinnedAUC: . . . 21
- 3.5 New suit of metrics . . . 22

3.5.1 AUC based metrics: . . . 22
3.5.2 Average equality gap . . . 23
3.5.3 Generalised mean AUC: . . . 25
3.6 Metrics Selection: . . . 25

4 Experiment **27**
4.1 Dependent and Independent variables: . . . 27
4.2 Experiment set-up / Tools used . . . 27
4.2.1 Software Environment . . . 27
4.2.2 Hardware Environment: . . . 28
4.3 Dataset Description: . . . 29
4.4 Data preprocessing: . . . 29
4.4.1 Adding identity information: . . . 29
4.4.2 Tokenizing the sentences: . . . 31
4.4.3 Converting the Tokenized words into word vectors: . . . 31
4.5 Training the models using multi-task learning: . . . 32
4.5.1 Using a custom loss Function: . . . 33
4.5.2 Hyper-parameter tuning: . . . 34
4.6 Testing the Models: . . . 34
4.6.1 Classification performance: . . . 35
4.6.2 Bias mitigation performance: . . . 35

5 Results **39**
5.1 Classification Performance: . . . 39
5.2 Bias mitigation performance: . . . 40
5.2.1 Male: . . . 40
5.2.2 Female: . . . 41
5.2.3 Homosexual: . . . 41
5.2.4 Christian: . . . 42
5.2.5 Jewish: . . . 43
5.2.6 Muslim: . . . 43
5.2.7 Black: . . . 44
5.2.8 White: . . . 45
5.2.9 Physical disability: . . . 46
5.2.10 Generalised mean AUC: . . . 47
5.3 Significance Test . . . 48
5.3.1 Friedmans test: . . . 48
5.3.2 Nemenyi test: . . . 48
5.3.3 Subgroup-AUC: . . . 49
5.3.4 BPSN-AUC: . . . 50
5.3.5 BNSP-AUC: . . . 51

6 Analysis and Discussion **53**
6.1 Analysis of Literature Review . . . 53
6.2 Analysis of Experiment . . . 53
6.2.1 Classification performance: . . . 53
6.2.2 Bias mitigation performance: . . . 53

6.3 Discussion 54
6.4 Validity Threats 54
6.4.1 Internal validity: 54
6.4.2 External validity: 54
6.4.3 Conclusion validity: 55

7 Conclusion and Future work **57**
7.1 Conclusion 57
7.2 Future work 57

List of Figures

2.1 Sub-domains of AI 6
2.2 Hard parameter sharing for multitask learning in deep neural networks 10
2.3 Soft parameter sharing for multitask learning in deep neural networks 11
2.4 An overview of the Bi-LSTM model proposed by Vidya et al. [53] . . 12
2.5 Example of "How Self-Attention Mechanism works"[11] 13
2.6 The architecture of the Transformer proposed in [55] 14
2.7 Overview of the BERT architecture. 14

3.1 An Example of toxic distributions showing the bias corresponding to an identity subgroup and how pinned AUC is affected by the background class distribution [4]. 21
3.2 A plot of true positive rates of sub groups and the background distribution of a hypothetical classifier in which the shaded area represents the positive average equality gap. 24

4.1 Adding the identity information using a multi class BERT classifier. . 30
4.2 Tokenizing the sentence into individual tokens and converting the tokens into word vectors 31
4.3 Overview of the BERT embedding layer [18]. 32
4.4 Sharing the same network parameters for all the tasks to ensure hard parameter sharing. 33
4.5 ROC cure of a hypothetical classifier [19] 36

5.1 Bar plot showing the F1-scores, Overall-AUC of all the models . . . 39
5.2 Bar plot showing the per-identity matrix for identity subgroup "Male" 40
5.3 Bar plot showing the per-identity matrix for identity subgroup "Female" 41
5.4 Bar plot showing the per-identity matrix for identity subgroup "Homosexual" 42
5.5 Bar plot showing the per-identity matrix for identity subgroup "Christian" 43
5.6 Bar plot showing the per-identity matrix for identity subgroup "Jewish" 44
5.7 Bar plot showing the per-identity matrix for identity subgroup "Muslim" 44
5.8 Bar plot showing the per-identity matrix for identity subgroup "Black" 45
5.9 Bar plot showing the per-identity matrix for identity subgroup "White" 46

5.10 Bar plot showing the per-identity matrix for identity subgroup "Physical disability" 47
5.11 Bar plot showing the per-identity matrix for identity subgroup "Generalised AUC Mean". 47

List of Tables

4.1 Hardware configuration of the system used to train the model 28
4.2 List of identity labels mentioned in the dataset 29
4.3 Percentage of comments corresponding to different identity subgroups in the training data 30
4.4 Hyper-parameter tuning in all the three models 34

5.1 F1-score and Overall-AUC for all the three models 39
5.2 Values of per-identity AUC metrics for identity subgroup 'male' . . . 40
5.3 Values of per-identity AUC metrics for identity subgroup 'female' . . 41
5.4 Values of per-identity AUC metrics for identity subgroup 'homosexual' 42
5.5 Values of per-identity AUC metrics for identity subgroup 'christian' . 42
5.6 Values of per-identity AUC metrics for identity subgroup 'jewish' . . 43
5.7 Values of per-identity AUC metrics for identity subgroup 'Muslim' . . 44
5.8 Values of per-identity AUC metrics for identity subgroup 'black' . . . 45
5.9 Values of per-identity AUC metrics for identity subgroup 'white' . . . 46
5.10 Values of per-identity AUC metrics for identity subgroup 'physical disability' 46
5.11 Values of generalised mean AUC for all the models 47
5.12 Subgroup-AUC for all the nine identity subgroups 49
5.13 Pairwise average rank difference for subgroup AUC 49
5.14 BPSN-AUC for all the nine identity subgroups 50
5.15 Pairwise average rank difference for BPSN AUC 50
5.16 BNSP-AUC for all the nine identity subgroups 51
5.17 Pairwise average rank difference for BNSP-AUC 51

List of Abbreviations

AI Artificial Intelligence

BERT Bidirectional Encoder Representations from Transformers

Bi-LSTM Bidirectional Long Short-Term Memory

CNN Convolutional Neural Networks

GPU Graphics Processing Unit

GRU Gated Recurrent Unit

ML Machine Learning

MT Machine Translation

MTL Multitask Learning

NLP Natural Language Processing

QA Question Answering

RNN Recurrent Neural Networks

Chapter 1

Introduction

Nowadays, smartphones and personal computers have become an essential tool for day to day tasks. With increased use of social media [38], people tend to use communication channels like Twitter, Facebook etc. to express their ideas and views. But some people are hesitant, often afraid of expressing themselves freely because of the fear of getting harassed or bullied. The increase of toxic comments had increased the threat of harassment or abuse online, which made many people stop expressing themselves and made them give up on seeking different opinions. Toxic comments are the comments that are rude and disrespectful and are more likely to make someone leave a discussion [45]. According to the survey conducted by the Pew Research institute in 2017, 41% of the internet users have experienced online harassment in which 18% are severe harassments [20]. Platforms struggle to effectively facilitate healthy conversations, leading many communities to limit or completely shut down user comments. As the social media forms and chat systems contain lots of text conversations, it is impossible to manually monitor and control toxic behavior. So, we need an automated approach to monitor and control toxic behaviour. Computers which are generally good at logical operation are not intelligent enough to mimic the human brain which can easily understand text conversations. Here comes a field called NLP(Natural Language Processing) using which we can build models to detect and classify toxic conversations. NLP uses tools and techniques like machine learning, deep learning to detect and analyze patterns in speech and text.

Advancements in machine learning, deep learning and availability of a large amount of data (from big data sources like Facebook, Twitter, Wikipedia etc.) for training had led to the development of many deep learning and traditional machine learning models for detecting toxicity in text conversations. But a recent study shows how some models are biased towards certain identities like ‘gay’, ‘black’ and so on. For example, the model learns to incorrectly label some non- toxic comments (like “I’m a proud gay man”) as toxic because they contain identities (like “gay”) which are frequently referred in toxic conversations [19] [56]. The main cause of this bias is the disproportionate representation of some identity terms like ’gay’ in the data used to train the models i.e., the data-set contains more toxic examples with identities like ’gay’ that the non-toxic examples. As a result models trained on this data learned to over generalize identity terms as toxic terms. Research had been done in the past in which some bias mitigation methods and some models were also proposed to address this issue which are discussed clearly in chapter 2. In this research, we had used multitask learning to fine-tune a pretrained language representation model

called BERT to overcome this problem. In the next section, a detailed explanation about how and why some NLP models were wrongly biased towards some identity terms is given.

1.1 Problem statement

As mentioned earlier, some models were biased towards frequently attacked identity terms (like gay). Fox example models learn to associate toxicity to non-toxic identities (like 'gay') as they were frequently used in toxic conversations. Unfortunately, this is due to the data used for training the model in which certain identities are frequently referred to in an abusive manner [19] [56]. The data set used to train this model contains more toxic examples with identity terms and contains very less training examples that use these identity terms in non-toxic manner. As a result, models trained on these data sets learn to over-generalize identity terms as toxic terms, and they are classifying non-toxic comments like "I'm a proud gay man" as toxic. Even identities related to certain nations (like Jews) or religions (like Muslim) which are frequently attacked in toxic conversations are learned as triggers for toxicity [56].

Some research had been done in the past in which some bias mitigation methods and some models were also proposed to address this issue. One of such research is using multitask learning to reduce the model bias [53]. This paper suggests that multitask learning will boost the performance if tasks are related to each other. Which means, if the tasks are related to each other, it is better to learn them simultaneously than to learn each of them individually. Vaidya at el.[53] had trained a Bi-LSTM in such a way that it can perform two tasks in parallel, namely identity task and toxicity task. The identity task is designed to predict the presence of identity in a comment, whereas the toxicity task is designed to predict whether a comment is toxic or not. These two tasks work jointly to reduce identity bias. In this paper, they also mentioned that using attention based models like BERT instead of Bi-LSTM may increase the performance. So, in this research, we would like to use multitask learning to train BERT based models so that they can classify toxic comments with reduced bias over identity terms and we would also like to compare our BERT based model with the previous one proposed by [53] to know whether it offers better performance or not.

In general BERT models are large i.e., they have millions of parameters[18]. In this study, two variants of BERT have been chosen i.e., BERT-base and DistilBERT as they are relatively small and faster when compared to other variants of BERT like BERT-large. Despite being small these models still provide similar results to other variants of BERT in various downstream NLP tasks [18][48]. Also, these models are computationally less expensive i.e. both the inference time and training time for these models is less when compared to other variants of BERT [18]. As this study is focused on improving toxic comment classification in online chat and social media platforms it is better to use models which have less inference time as social media platforms and chat systems produce large amounts of text data in real time that need to be processed [26]. Also, DistilBERT is small enough that it can be employed

in edge devices (like on mobile devices) there by opening potential for various novel and interesting language processing applications [48].

1.2 Aim and Objective

The aim of this thesis to train the BERT-base and DistilBERT using multi task learning and compare them with the previous Bi-LSTM [53] and the main objectives of this thesis are

1. To train BERT-base, DistilBERT and the previous Bi-LSTM model using multitask learning, so that they can classify toxic comments with reduced identity bias.

2. To test the bias mitigation performance and classification performance of the trained model.

3. To compare the proposed BERT models with the previous Bi-LSTM model in terms of bias mitigation and classification performance.

1.3 Research Questions

RQ1. What metrics are best suitable to measure the bias mitigation performance of the proposed models and why?

Motivation: Apart from toxicity classification, the proposed model should also mitigate the identity bias over frequently attacked non-toxic identities. A model is said to exhibit identity bias if its output differs for comments related to different Identity groups [45]. For example, consider the comments "I'm proud Man" and "I'm a proud gay man", an ideal model should assign a same toxic score to both the statements, if a model is biased then it will assign different scores for those two comments. So, to tell whether the model is biased or not, we should know whether the model is consistent over different identities or not. So, we need a metric with which we can measure the extent to which the model is consistent over different identity groups. So, we would like to study various metrics that were used in the past and choose the metric that fits our problem.

RQ2. How well the proposed models can measure the toxicity by reducing the unintended bias over certain non-toxic identity terms when compared to previous model?

Motivation: After building the three models BERT-base and Distil-BERT and the previous Bi-LSTM proposed by Vaidya at el.[53] we should measure their performance to know how well these models are able to measure the toxicity in text conversations and to which extent these models are able to reduce the unintended bias over frequently attacked non-toxic identity terms when compared to the previous model proposed in paper [53].

1.4 Outline

The structure of the thesis work is discussed in this section.

Chapter 1: In this chapter, the Introduction and motivation of the thesis are explained along with the aim and objectives, problem statement and research questions.

Chapter 2: This chapter explains the concepts related to the thesis and past studies that are related to the current thesis.

Chapter 3: This chapter provides details about the method used to conduct the literature review and also provides the outcomes of the literature review.

Chapter 4: This chapter tells bout Experimentation i.e. describes the data collection, data prepossessing, training and testing procedures.

Chapter 5: In this chapter, results from the experiment are presented.

Chapter 6: In this chapter, analysis and discussion is made regarding the obtained results from the literature review and the experiment.

Chapter 7: In this chapter, conclusion and future work are explained.

Chapter 2

Background

2.1 Natural language Processing

A theory-motivated collection of computational techniques for the automated analysis and representation of human language is described as NLP [8] [17]. The examination of text by humans is easier than machines since the way machines process the text data is way different from humans [8]. NLP is mainly focused on developing ways to program computers so that they can understand and interpret natural language as humans do [17]. NLP uses the knowledge, tools and techniques of various sub-domains of artificial intelligence like machine learning and deep learning to understand and interpret the natural language.

2.2 Artificial Intelligence

Artificial intelligence(AI) refers to the development of human-like intelligence in machines or computers so that they can learn, reason, plan, perceive, or process information like humans. The main aim of AI is to replace humans with computers. AI is also known as the study of "intelligent agents" which can study the environment around it and take actions accordingly to accomplish the given task [7].

2.3 Machine Learning

Machine Learning(ML) is a sub-domain of AI which focuses on developing algorithms that can learn the underlying relations or patterns in the data to make intelligent decisions [24]. The main aim of ML is to give computers the ability to learn from data without being explicitly programmed. ML algorithms use the past data to gain the experience using which they can make rational decisions regarding the unseen or future scenarios.

2.4 Deep Learning

Deep learning) is a sub-domain of machine learning which uses algorithms called artificial neural networks to process the data and explore patterns in it [49]. Artificial neural networks are a class of machine learning algorithms which are vaguely inspired by the biological neural networks which constitute the human brain. Models

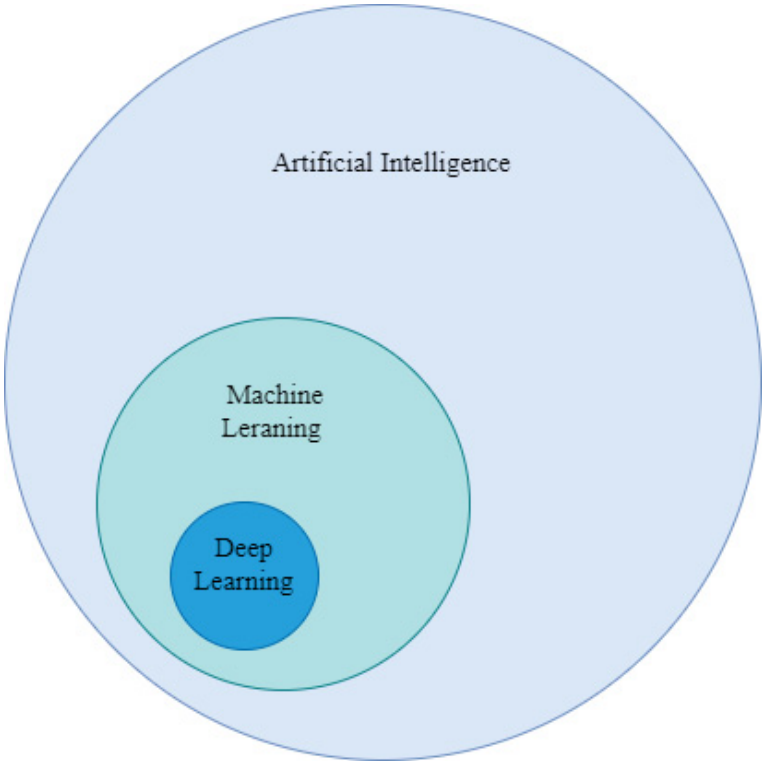

Figure 2.1: Sub-domains of AI

like CNN(Convolution Neural Network) and RNN(Recurrent Neural Network) are well known deep learning algorithms which are used in various tasks [16]. These algorithms are known to achieve a state of the art results in various fields like computer vision, NLP and pattern recognition. Deep learning is a sub-domain of Machine learning whereas machine learning falls into a much broader deceptive called AI.

2.5 Applications of NLP

Recent development in the fields of AI, machine learning and deep learning has led to the development of several NLP models that can perform a wide variety of tasks related to NLP like QA(Question-Answering), text summarization, text classification, machine translation [17].

2.5.1 QA(Question Answering)

QA is a computer science discipline which focuses on building systems that can automatically answer questions asked by humans which are in natural language [10]. QA is based on Information Retrieval [8]. While answering a given question, QA may give a single answer or many answers which are relevant to that question [50] [31]. Development in the fiends of machine learning, deep learning and NLP has led to the development of several models for QA. Yeo H at el. [60] had proposed a QA system for answering complex queries related to health care. In this research the question is divided into small sub queries and a logistic regression classifier is used

to classify the sub queries and then the answers to these queries are extracted from the relational database. Kapashi et al. [27] and Kumar et al. [30] had introduced a new class of deep learning models called memory networks which can be used in various QA tasks. These memory networks are inspired by LSTM(Long Short-Term Memory) networks.

2.5.2 Text Summarization

Text summarization means reducing or summarizing the given text data into a shorter version without losing any key informational elements and context of the data. Often it is hard or time consuming to process the entire text documents, so a summarized version of it comes in handy. Text summarization methods are categorised into two types namely extractive methods and abstraction methods[5] [1]. Extraction method refers to the selection of key sentences from the given data and making it into a small document. Abstractive method is used to understand the main concepts in a given document and then express those concepts in clear natural language [1] [52]. Development in the fiends of NLP has led to the development of several effective text summarization models. Merchant et al. [34] had proposed an unsupervised statistical-algebraic summarization technique called latent semantic analysis to summarize the legal documents. Many machine learning and deep learning models were also proposed for automatic text summarization which were quite effective [14] [15] [61] [47].

2.5.3 Text Classification

Text classification is the process of categorizing text data into groups based on the context of the text. The digital age that we live in, is surrounded by text from various places like social media, advertising commercials, Ebooks and so on. Most of this data is unstructured so classifying or grouping this data is very essential for further analysis. Text classification is a subdomain of NLP which has broad applications like hatespeach detection, spam detection, sentiment analysis and intent detection. The availability of text data and the advancements in the field of machine learning and deep learning had motivated several researchers to focus on building machine learning and deep learning based text classification models for various applications like sentiment analysis, hate speech detection and so on [65] [63].

2.5.4 Machine Translation

MT(Machine Translation) is a sub domain of NLP which aims to develop computer systems that can translate text from one language to another language. The availability of large amounts of text data in various languages has led to the development of many data driven MT approaches since 1990 [32]. The main goal of data-driven MT is to use bi-lingual training data to train the machines to get translation knowledge and then use it to translate unseen sentences of one language to another language. The availability of large amounts of multilingual data has grabbed the attention of several NLP researchers to focus on applying machine learning and deep learning

for building effective machine translation models. Deep learning models like [41] are known to achieve state of the art results in machine translation tasks.

2.6 Bias in machine learning and deep learning

With the widespread use of machine learning and deep learning models in various aspects of our daily life it is important to ensure the consistency of these systems. These models are being used in various scenarios like movie recommendations, product suggestions to buy and in making important decisions regarding loan applications [36], dating and hiring [3] [12]. This is a clear advantage in automating day to day tasks using computer systems as they do not become tired or buried like humans. Moreover automation also saves a lot of time and effort. But recent research had found that some machine learning models are prone to biases which makes their decisions unfair [29] [37]. Irsoy et al. [33] discusses various biases that were found in a wide range of machine learning and deep learning models which were used in any applications ranging from image recognition, text classification and so on. In the upcoming section we discuss various biases in text classification and methods proposed to mitigate them.

2.7 Bias in text classification models

In this section, we would like to discuss some past studies that were done on reducing various kinds of biases in text classification models like identity bias, gender bias and sentiment bias, efforts that were put to mitigate them. Dixon et al. [19] and Wiegand et al. [56] said that some models were wrongly biased towards some non-toxic identity terms. They also stated that the main reason for this bias is the disproportionate representation of identity terms in the training data, that is the training data consists of more toxic examples which these identities when compared to non-toxic examples. Wiegand et al. [56] had taken some popular data sets that were used to train models for detecting abusive language and showed how the bias in the data set is reflected in the model trained on that data set. The author also showed that models trained on the Waseem-dataset are negatively biased towards neutral terms like commentator, announcer, and football. This is because Waseem-dataset has been sampled in such a way that it contains a high proportion of toxic micro-posts which discuss the role of women in sports, particularly about their suitability as football commentators. Similarly, models trained on the Warner Dataset had associated toxicity with neutral terms like Hollywood.

Dixon et al. [19], Elizabeth et al. [44] had also proposed some data balancing techniques to reduce the unintended bias which were quite effective. The method they used to reduce the identity bias is adding additional data that contains non-toxic examples of the identity terms which were most disproportionately used in toxic comments than that in non-toxic comments. Dixon et al. [19] had used real world data to balance the data set whereas Elizabeth et al. [44] had used synthetic data generation methods to add additional data. Apart from data balancing strategies,

other approaches like adversarial learning and multitask learning were also used for reducing the identity bias in text classification models [53], [59]. Zhang et al. [62] had used adversarial learning to reduce the identity bias in text classification models.

Recently gender bias, a kind of identity bias, has also gained a lot of research attention. Several methods had been proposed to mitigate the gender bias, one of which is the research done by Park Ji Ho et al. [39]. Park et al. [39] had experimented with three bias mitigation methods (namely debiasing word embeddings, gender swap data argumentation, and fine-tuning with a large corpus) to reduce the gender bias. They used this method on three model architectures, namely CNN, GRU(Gated Recurrent Unit) and Bidirectional GRU with self-attention and compared them. They stated that using gender swap data argumentation with GRU had provided the best results, which reduces the gender bias up to 90-98%.

Prates et al. [42] and Vammassenhove et al. [54] had revealed the existence of the gender bias in machine translation models i.e machine translation models are leaned to disproportionately represent the speaker of the sentence as male, this is because the training data contains more male sourced data points, as a result the models trained on this data will be skewed towards male sourced data points and thus these models will wrongly predict the speaker of the sentence as mail. In the paper [54] had proposed a method called gender tapping to mitigate this kind of bias in machine translation.

Apart from gender and identity bias, there is a little research which focuses on a bias called sentiment bias which is observed in some sentiment analysis models. A model is said to exhibit sentiment bias if it assigns positive or negative sentiment to neutral words or phrases. Yang et al. [59] had proposed a new mechanism called polar attention to mitigate the sentiment bias. They also used this mechanism in various model architecture like RNN, CNN, BERT and compared the results. They conclude that applying the polar attention mechanic had reduced the sentiment bias in all the models.

Chris et al. [9] had used adversarial learning to mitigate sentiment bias corresponding to some demographics attributes. In this research, the authors focused on two famous word embeddings namely word2vek [35] and GloVe [40] embeddings. They stated that adversarial learning provides satisfactory debiasing results and they also mentioned that this debiasing approach not only reduces the sentiment bias but it also helps in reducing identity bias in word embeddings. In this study, the authors had used the metrics pinned AUC [19] to find how well the proposed debiasing approach had removed the identity bias over different identity subgroups.

Sweeney et al [51] had proposed a metric to measure the unintended sentiment bias associated with particular demographic identities in word embeddings. They proposed a metric named RNBS(Relative Negative Sentiment Bias) which measures the fairness in word embeddings using relative negative sentiment associated with various demographic identity terms. In this work, the author had focused only on measuring negative sentiment bias associated particular identity subgroups but they

haven't explored or provided ways to measure positive sentiment bias associated with certain identity subgroups.

2.8 Multitask Learning

Generally in machine learning, we train a single model or an ensemble of models to perform a single task, we fine-turn and tweak these model until the performance no longer increases. Although we achieve acceptable performance by laser-focusing on one single task, we are ignoring the information that might help us do even better. Specifically, the information coming from the training signals of related tasks. Sharing the representations between the related tasks will help the model generalize better on the original task. This approach is called MTL(Multitask Learning) [46]. MTL is a mechanism of inductive transfer whose primary objective is to enhance the efficiency of generalization. MTL enhances generalization by exploiting the domain-specific information found in the training signals of similar tasks [6] [13]. MTL has many applications in various fields including computer vision, bioinformatics, health informatics, speech, processing of natural languages, web applications, ubiquitous computing and so on.

In MTL several learning tasks are trained simultaneously, each of which may be a general learning task such as supervised tasks (e.g. Problems of classification or regression), unsupervised tasks (e.g.Problems of clustering), semi-supervised tasks and tasks of reinforcement learning. Among these learning tasks, if all of them or at least a subset of them are connected to each other then it is found that learning these tasks jointly will lead to a great deal of change in performance when compared to learning them individually. Therefore, MTL seeks to enhance the efficiency of various tasks which are related to each other [64]. The most commonly used approaches for implementing multitask learning in the context of deep learning are hard parameter sharing and soft parameter sharing.

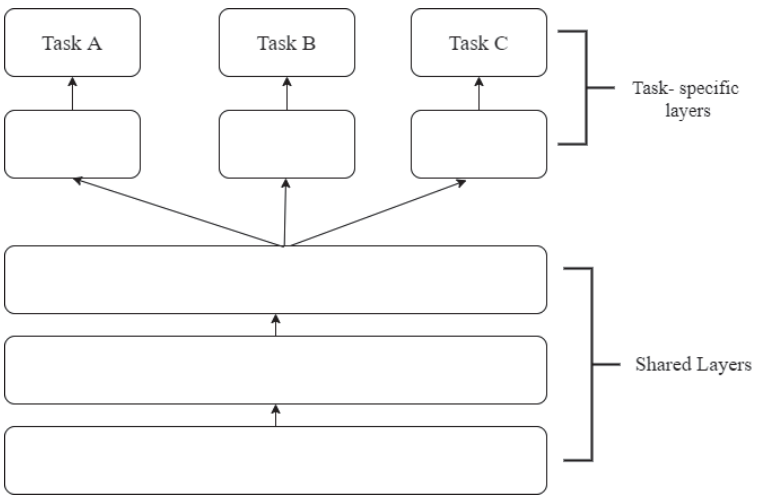

Figure 2.2: Hard parameter sharing for multitask learning in deep neural networks

Hard parameter sharing: It is the most widely used approach for MTL in deep neural networks. In this approach, all the tasks have the same hidden layers and every task has its own set of task specific layers as shown in the Figure 2.2. Hard parameter sharing is known for its ability to reduce over fitting [64] [2].

Sort parameter sharing: In soft parameter sharing each task has its own model with has its own parameters as shown in the Figure 2.3. In this approach, the distance between the parameters of different models is reduced by employing regularisation between them.

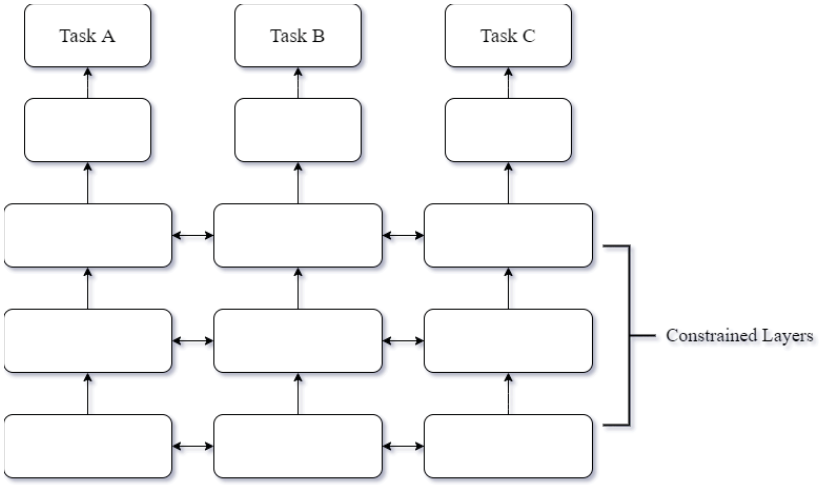

Figure 2.3: Soft parameter sharing for multitask learning in deep neural networks

2.9 Models used and their architecture:

In this comparative analysis, we are using three models namely BERT-base, DistilBERT and Bi-LSTM, BERT proposed by Vaidya et al. [53].

2.9.1 Architecture of LSTM model:

The overview of the Bi-LSTM model proposed in the previous study is illustrated in the Figure 2.4. The starting layer of the model is an embedding layer which uses pre-trained GloVe embeddings [40] to convert tokenized words into word vectors. These word vectors are then fed as inputs to the pair of Bi-LSTM layers which outputs the hidden state representations for every given word vector. Then the hidden states for all the words are fed as inputs to the attention layer which uses the feed-forward attention proposed by Colin et al. [43] to calculate the attention scores to all the hidden states with respect to one another. Then the outputs from the attention layer are fed as inputs to a pair of feed-forward layers and finally, the outputs from the feed-forward layers are fed to an out layer which uses a soft-max activation to predict the out labels.

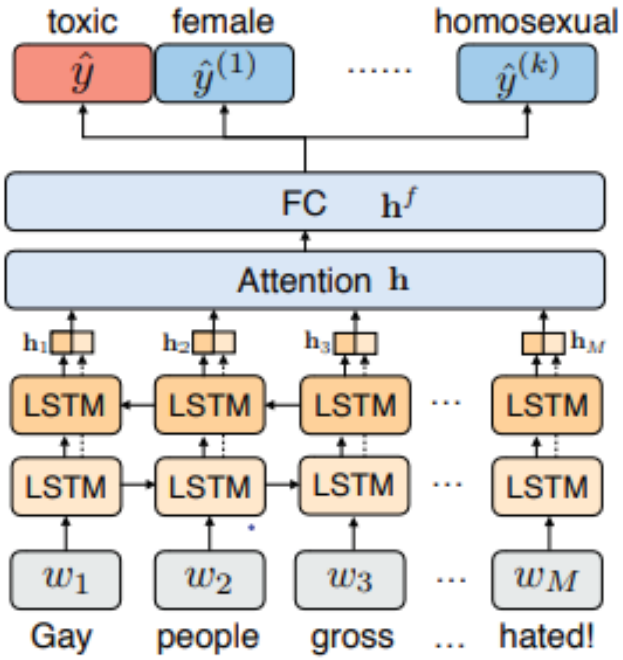

Figure 2.4: An overview of the Bi-LSTM model proposed by Vidya et al. [53]

2.9.2 Architecture of the BERT model:

BERT is a language representation model built using transformer architecture which is based on the attention mechanism [55], [18]. It is pre-trained deep learning model which can be used in various language tasks like question answering, text classification, machine translation and so on [18]. It is trained on large text corpus collected from Wikipedia and BookCorpus. So the model has the pre-training knowledge and just needs task-specific fine-tuning to get it working. Models based on BERT had outperformed the previous deep learning models in various NLP tasks [18]. The state-of-the-art performance of the BERT is due to the self-attention mechanism that it uses to understand the language. For example, consider the sentence "The animal didn't cross the street because it was too wide ". As humans, we know that the word 'it' refers to the 'street'. But it is quite a difficult task for the machine to understand whether the word 'it' refers to the 'animal' or the 'street'. Now let us see how self-attention works here. While moving through the sentence from left to right self-attention for every word is calculated, which means while we are at the word 'it' the model will calculate its attention (attention refers to the extent to which the two words of the sentence are connected to each other) towards every other word in the sentence, which helps the model to understand the extent to which the word 'it' is connected to all the other words. This self-attention mechanism makes the BERT model understand the language in a better way.

As mentioned earlier BERT uses a famous attention mechanism called transformers which was proposed by Vaswani et . [55]. The transformer contains two blocks

Figure 2.5: Example of "How Self-Attention Mechanism works"[11]

namely an encoder and a decoder as as shown in the Figure 2.6. It was proposed to over come the problems in the existing machine translation models. In the transformer architecture the encoder block will use multi head attention to encode the context and the meaning of every word in the input sentence and it produces the contextual embeddings for every word in the given sentence. Now the decoder will use the contextual embeddings produced by the encoder to generate the translation for the input sentence.

BERT uses a cut down version of transformer i.e.. it only uses the encoder blocks of the transformer. BERT is basically a stack of encoder blocks in which the output from one encoder is given as input to the next encoder as shown in the Figure 2.7. First, the input text sentence is tokenized and then the tokens are fed as inputs to the embedding layer which generates word embeddings for all the tokens in the given sentence. Then these embeddings are fed as inputs to the encoder stack as shown in the Figure 2.6. Now the encoder will use the multi head attention to calculate the attention scores of a given word embedding with respect to all the other words in the given sentence. Based on this attention scores the input word embeddings are modified in such a way that they will encapsulate the meaning and their contact in the given sentence. As the embeddings pass sequentially through the encoder stack they will be modified in such a way that each embedding will encapsulate the meaning and its context with respect to every other words in the input sentence.

BERT-base was proposed in the paper [18], it has 12 encoder blocks stacked one on top of each other and each block uses a multi-head attention block of size 12 i.e.. in an encoder block every word embedding will have 12 independent attention patterns with respect to every other word in the given sentence. In total BERT-base has approximately 110 million trainable parameters.

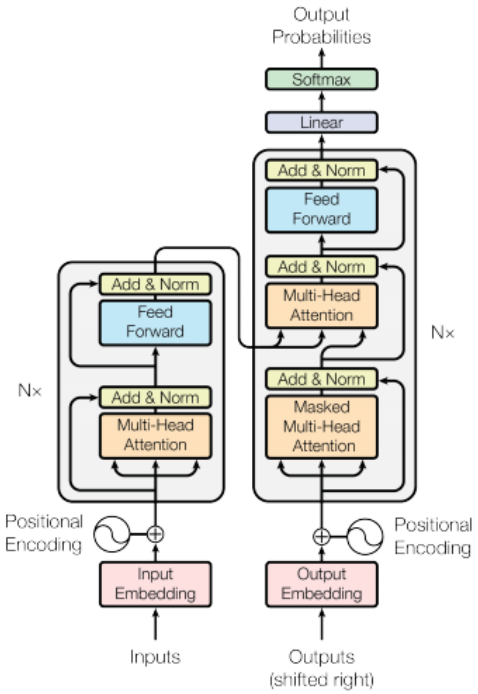

Figure 2.6: The architecture of the Transformer proposed in [55]

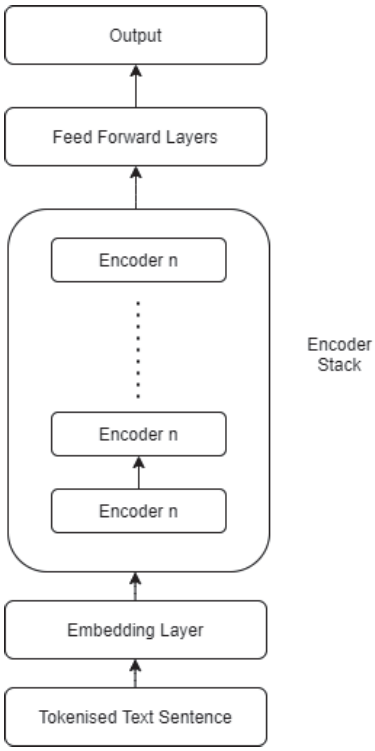

Figure 2.7: Overview of the BERT architecture.

DistilBERT [48] is smaller version of BERT-base [18], it is obtained by applying knowledge distillation on smaller transformer using the preparing knowledge of the

BERT-base. Knowledge distillation is compression technique proposed by Harrington [25] which is used to reduce the size of the neural networks, in this technique a smaller model called the student is trained to resemble the behaviour a larger model or ensemble of models called the teacher. DistilBERT uses 40% smaller transformers when compared to the original transformer [55] used in BERT-base. The transformers used in DistilBERT have almost half the layers of the transformers used in BERT-base. Although DistilBERT is smaller, it is still able to retains 97% of the language understanding capabilities of the original BERT-base model and is also 60% faster than BERT-base in terms of inference time.

Chapter 3

Literature Review

A literature review is conducted to answer the RQ1: "What metrics are best suitable to measure the bias mitigation performance of the proposed models and why?". Recent research on fairness in machine learning had provided many definitions of fairness and several metrics were also proposed to measure the fairness in machine learning models. But our interest is to measure the fairness of the text classification models namely BERT-base, DistilBERT and Bi-LSTM that were used in current study to detect toxic comments with reduced identity bias. In this context we can relate the fairness to the extent to which the model are able to reduce the identity bias.

A model is said to be unfair or exhibit identity bias if it assigns different toxicity scores for similar sentences with different identity terms. So a model is said to be fair if it assigns similar toxicity scores to sentences with different identity terms. In order to measure the fairness of the model, we should focus on measuring the extent to which the toxicity scores given by the model were skewed while dealing with sentences containing different identity terms. Several metrics were proposed to measure this kind of biases in the text classification models, in order to find the metric that suits our problem we had done a literature review on various metrics that were used to measure biases in text classification models.

3.1 Mythology

The following are the steps followed while conducting the literature review.

1. Investigation of the primary study
2. Selection criteria.

3.1.1 Investigation of primary study:

Initially, the keywords related to the research question are extracted. Then these words are used to formulate search strings using which we can search for the relevant literature in different libraries like IEEE Xplore, Scopus, ACN digital Library and so on. The keywords used in this step are bias, gender bias, identity bias, fairness, text classification and metrics. Different search strings are formulated using these

keywords to search for the available literature.

3.1.2 Selection criteria:

While searching the literature using the search strings the following criteria have been used to select the required literature

Inclusion criteria:

1. Studies that are published after 2012 are considered.
2. Studies that are available in full text are considered
3. Studies in the English language are considered.
4. Studies that are related to the biases in text classification such as gender bias, identity bias are considered.
5. Studies that are related to the current thesis are considered.

Exclusion Criteria:

1. Studies that are not in the English language are excluded.
2. Articles in the form of abstract and PPT's are not considered.
3. Studies that are related to various biases in other domains like Computer vision, Image processing are excluded.

After selecting the required literature using the selection criteria we used a technique called snowballing [57] to extract additional papers from the selected literature. In snowballing references from the papers collected in the previous step are again filtered using the selection criteria and any new papers which are relevant to the study are added. The following are the various metrics that were extracted from the literature review

3.2 Error Rate Equality Difference

This metric is proposed by Dixon et al. [19] and is based on the equality of odds concept proposed in [23]. According to equality of odds, a model is said to be fair, if the FPR (False Positive Rates) and FNR (False Negative Rates) are equal across comments containing different identity terms.

$$FPR = \frac{FalsePositives}{(FalsePositives + TrueNegatives)} \tag{3.1}$$

$$FNR = \frac{FalseNegatives}{FalseNegatives + TruePositives} \tag{3.2}$$

For example consider two sample sets namely background-sample and identity-sample which are extracted from the test set. The identity-sample contains only the comments corresponding to particular identity subgroup. But the sample set contains comments with and without the identity term which resembles the distribution of the original test set. A model is said to be fair or unbiased towards a particular identity if it has equal false positives and false negative rates on both the test sets. Based on this concept the author had defined the error rate equality difference as pair of two metrics namely false positive equality rate and false negative equality rate.

3.2.1 FPED (False-positive equality difference):

It is the sum of pairwise differences between the false positive rates on sample set and identity sample set corresponding to all the identity subgroups [19].

$$FPED = \sum_{t \in T} |FPR - FPR_t| \tag{3.3}$$

Whereas

FPR refers to False Positive Rates on the sample set.
FPR$_t$ refers to False Positive Rates on an identity sample set corresponding to a particular identity subgroup.
t refers to a set of identity subgroups.

3.2.2 FNED (False-negative equality difference):

It is the sum of pairwise differences between the false negative rates on sample set and identity sample set corresponding to all the identity subgroups [19] .

$$FNED = \sum_{t \in T} |FNR - FNR_t| \tag{3.4}$$

Whereas

FNR refers to False-negative rates on the sample set.
FNR$_t$ refers to False-negative rates on an identity sample corresponding to a particular identity subgroup.
t refers to a set of identity subgroups.

A model is said to be fair if it has lower false positive equality and false-negative equality differences. While comparing multiple models in terms of fairness or bias mitigation, a model which has low error rate equality difference(False positive equality difference and False positive equality difference) is considered to be the fairest model. Error rate equality difference can only be applied while dealing with distinct classification outputs(either 0 or 1). Some models will output a probability i.e... value between 0 and 1, in such cases, we have to use a threshold to convert the output probabilities into distinct classification outputs(either 0 or 1).

3.3 Pinned AUC

Pinned AUC(area under the receiver operator characteristic) is a threshold agnostic metric and is based on the AUC metric proposed in [21]. This metric is proposed in the paper [19], to overcome the challenges of the error rate equality difference. Many classification models provide a classification score or probability rather than a direct classification value. But metrics like error rate equality difference will evaluate the bias only while dealing with direct classification values or after applying a threshold to the output model probabilities. In order to overcome this problem, the pinned AUC metric is proposed and it uses a threshold agnostic approach to measure the bias in a wide range of models [19].

The pinned AUC metric is inspired by the popular AUC metric proposed in [21]. The direct application AUC metric [21] to a biased model will provide a measure which is inappropriate for measuring unintended bias i.e application of AUC to a biased model will result in a lower AUC score. But other than bais there may be also several other reasons for the lower AUC score, moreover, this lower AUC score does not also help in distinguishing which identity subgroups are responsible for the model bias.

Inorder to know the bias corresponds to the individual identity group the metric pinned AUC is proposed. The pinned AUC is calculated for every identity sub group using a dataset containing two equally balanced components: sample of contains from the testset that has identity mention and sample of comments from the test set which represents the original distribution of test set. Combining or pinning the identity subgroup with the underlying distribution enables the AUC metric to capture the performance difference of the model on the identity subgroup with respect to the original distribution, there by giving a direct measure of bias.

$$pD_t = s(D_t) + s(D), |s(D_t)| = |s(D)| \tag{3.5}$$

$$pAUC_t = AUC(pD_t) \tag{3.6}$$

Here,
$\mathbf{pD_t}$ refers to the dataset obtained pinning identity subgroups with original distribution.
$\mathbf{s(D_t)}$ refers to a sample of comments from the testset that has a particular identity mentioned.
$\mathbf{s(D)}$ refers to a sample of comments which represents the original distribution of the test set.
$\mathbf{pAUC_t}$ refers to the pinned AUC corresponding to a particular identity 't' which is obtained by calculating the AUC on $\mathrm{pD_t}$.

3.3.1 Pinned AUC equality difference P AUCED:

Pinned AUC equality difference is similar to the error rate equality difference but the only difference is that, instead of using False positive or false negative rates we

will use the AUC [19] measures.

$$PinnedAUCEqualityDifference = \sum_{t \in T} |AUC - pAUC_t| \tag{3.7}$$

Here,
AUC refers to AUC on the complete testset.
pAUC$_t$ refers to the pinned AUC corresponding to a particular identity 't'.

3.4 Limitations of PinnedAUC:

Although pinned AUC is proven to be efficient in measuring the unintended bias but from the research by Lucas Dixon [4], it is evident that pinned AUC is prone to show inaccurate results when data contains different distributions of class labels corresponding to different identity subgroups. The original paper [19] which proposed the pinnedAUC used a synthetic data set in which different identity subgroups have the same class distribution. If the test set contains a different distribution of class labels among identity subgroups then the pinned AUC metric will over or under-represent the bias that is present [4]. In general, it is very hard to find data sets in which different identity subgroups will have the same class distribution, in such cases, the pinned AUC can not capture the correct measure of the bias.

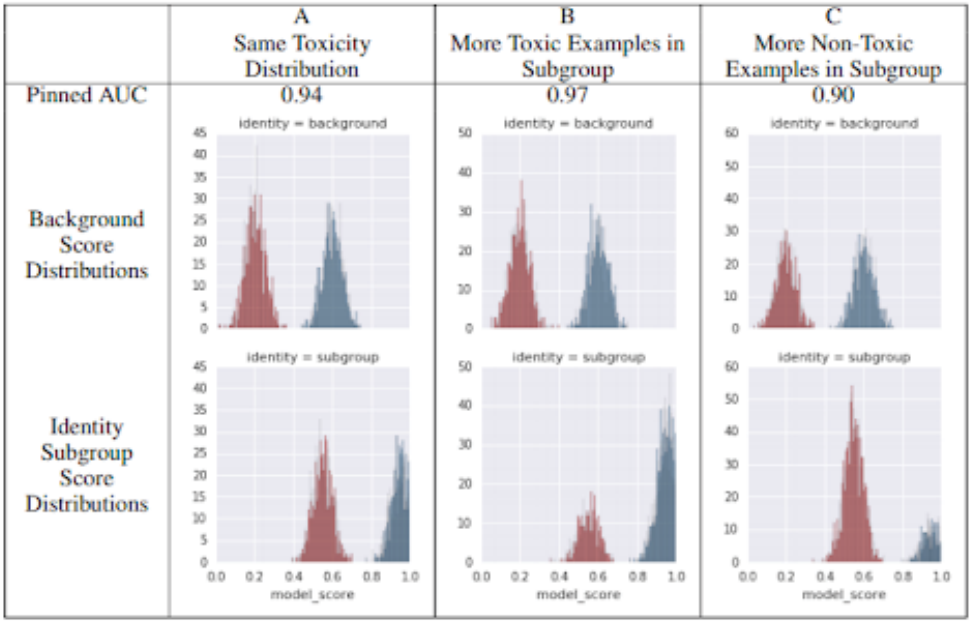

Figure 3.1: An Example of toxic distributions showing the bias corresponding to an identity subgroup and how pinned AUC is affected by the background class distribution [4].

In the Figure 3.1, the first and second rows show the distribution of toxicity scores of toxic and non-toxic examples corresponding to the hypothetical background data

and identity subgroup respectively. Here toxicity scores are calculated by a model which is known to have a bias. The red distribution represents the non-toxic examples and the blue one is corresponding to the toxic examples.

In Fig 3.1 all the columns A, B and C the identity subgroups have higher toxicity scores than that of the background. In fact, the non-toxic examples corresponding to the identity subgroup have similar scores to that of toxic examples in the background, so it is very difficult to find a single threshold which separates the toxic examples from non-toxic examples. In column B the identity subgroup has fewer non-toxic examples than the toxic examples and in column C the identity subgroup has more non-toxic examples than toxic examples. It is clear that all the three columns have the same kind of bias. In general, an ideal metric should give the same measure of bias for all the three data sets. But strangely the pinned AUC measures are different for all the three data sets. B has the highest pinned AUC whereas C has the least and A has a measure which is in between B and C. Here the pinned AUC measures are pointing out that C has the highest bias followed by A and then B. But in reality, all the scenarios represent the same bias.

3.5 New suit of metrics

In order to overcome the problem of pinned AUC, Lucas et al. [5] had proposed a suit of metrics to measure the unintended bias. In this paper, the author had used a combination of five metrics namely three metrics based on ROC-AUC, two metrics based on equality gap, each of which captures various kinds of biases and also explained how this new suite of metrics can pick the bias that the pinned AUC had obscured.

3.5.1 AUC based metrics:

In paper [5] the author had proposed three types of AUC metrics which can measure different kinds of biases in models by overcoming the limitations of pinned AUC and are as follows

Subgroup-AUC: It is the AUC measure calculated on a secondary dataset in which has equal number of positive(toxic) and negative(non-toxic) examples taken from the identity subgroup [5]. By this measure, we can see how well the model is able to separate or classify positive and negative sentences within the subgroup.

Background positive and subgroup negative(BPSN) AUC: It is the AUC measure calculated on a secondary data set in which all the positive(toxic) examples are taken from the background data and all the negative(non-toxic) examples are taken from the identity subgroup [5]. Here the sampling is done in such way that the secondary data set will have an equal number of positive and negative examples. A lower AUC measure indicates that the negative examples in the subgroup are given higher toxicity scores than the positive examples in the background. This means the positive examples from the subgroup are most likely to be classified as false positives

by the model. This measure indicates how well a model is able to reduce the false positive bias.

Background negative and subgroup positive(BNSP) AUC: It is the AUC measure calculated on a secondary dataset in which all the negative(non-toxic) examples are taken from the background data and all the positive(toxic) examples are taken from the identity subgroup[5]. Here the sampling is done in such the secondary dataset will have an equal number of positive and negative examples. A lower AUC measure indicates that the positive examples in the subgroup are given higher toxicity scores than the negative examples in the background. This means the negative examples from the subgroup are most likely to be classified as false negatives by the model. This measure indicates how well a model is able to reduce the false negative bias.

By using these three metrics we can measure different biases that may present in the model. It is also important to note these three AUC metrics are robust to the proportion of positive and negative examples in the test set. Here the samples which have equal number of positive and negative examples corresponding an identity group are used to measure the AUC, which means that disproportionate distribution of classes among identity sub groups(which is common in most real-world datasets) will not affect the results as it does in pinned AUC [4] [5].

3.5.2 Average equality gap

Equality gap is the difference between true positive rates of sub groups and the background data at a specific threshold. Borkan et al. [5] the author proposed two metrics namely positive average equality gap and negative average equality gap and are defined as follows.

Positive average equality gap:

Consider the graph in the figure 3.2, it is the plot of true positive rates x(t) of identity subgroup and background y(t) for every possible threshold t. Positive average equality gap is the area between the true positive plot and the line y =x i.e the area of the shaded portion in the figure3.2.

Negative average equality gap:

It is the same as the Positive average equality gap but the only difference is instead of using the true positive rates we will use true negative rates.

Dixon et al. [19] also defined the equality gap in terms of equality of odds [23]. Consider 'i' as a hypothetical data point selected from the positive background data(D_g) and 'j' is a hypothetical data point selected from the positive identity subgroup data(D_g^+), if the equality of opportunity is satisfied by the classifier then at every selected threshold the probability that i or j will have a higher toxicity score

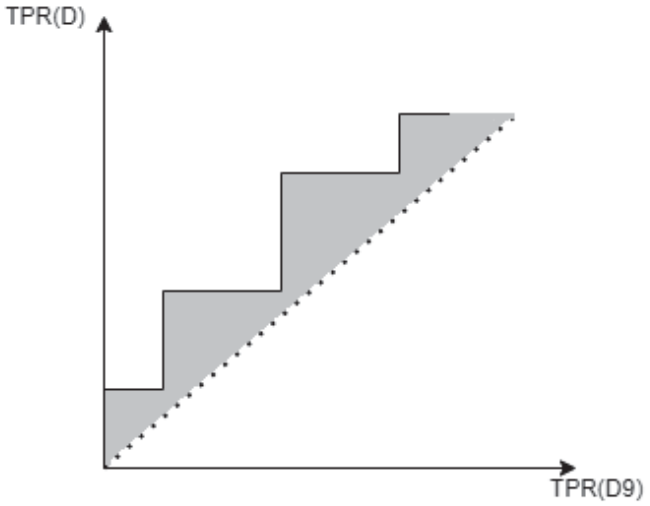

Figure 3.2: A plot of true positive rates of sub groups and the background distribution of a hypothetical classifier in which the shaded area represents the positive average equality gap.

is equal i.e.

$$P\{\hat{Y}_i > \hat{Y}_j | Y_i \in D^+, Y_j \in D_g^+\} = \frac{1}{2} \tag{3.8}$$

From this assumption, we can define the positive equality gap as

$$PositiveAEG = \frac{1}{2} - P\{\hat{Y}_i > \hat{Y}_j | Y_i \in D^+, Y_j \in D_g^+\} \tag{3.9}$$

The negative equality gap will also have a similar definition but the negative data points corresponding to the identity subgroup and the background are considered instead of positive. Positive or negative average equality gap will have a value ranging from -0.5 to 0.5. An optimal value for this metric is 0.

The five metrics namely subgroup-AUC, BPSN-AUC, BNSP-AUC, positive equality gap and negative equality gap can be used in finding different kinds of biases. The equality gap metrics are good at finding small score shifts, whereas the AUC metrics are good at finding large score shifts [5]. Here a small scale shift refers to a scenario where the classifier will give slightly higher or lower scores to a particular identity subgroup group but this shift will not affect the classification as the shift in the score is too small so that,it is still possible to find a threshold which can perfectly separate the positive and negative examples corresponding to both the background and the identity subgroup. Whereas a large score shift refers to a scenario where the classifier will give a much higher or lower score to the comments related to the identity subgroup so that, it is very difficult to find a single threshold which can separate the positive and negative examples corresponding to both the background and the identity subgroup. A threshold that performs better on separating positive and negative comments of background will do a bad job of separating comments from

the identity subgroup and vice versa.

In this thesis we are focusing on reducing the identity bias i.e negative examples belonging to particular identity subgroups are given a much higher toxicity score so that they are ended up classified as toxic. So, we can conclude that we are dealing with a problem corresponding to large score shifts, thus we can use AUC based metrics i.e subgroup-AUC, BPSN-AUC and BNSP-AUC to measure the bias mitigation performance of the models. In the upcomming sections we will calling theese three metrics namely subgroup-AUC, BPSN-AUC and BNSP-AUC as per-identity metrics.

3.5.3 Generalised mean AUC:

The per identity AUC metrics mentioned above i.e subgroup-AUC, BPSN-AUC and BNSP-AUC, only specify the bias mitigation performance over a single identity subgroup. But the metric generalised mean AUC [53] combines all the three metrics(namely subgroup-AUC, BPSN-AUC and BNSP-AUC) corresponding to all the identity subgroups into a single measure and represents the overall bias mitigation performance of the model.

$$M_p(m_s) = \left(\frac{1}{N} \sum_{s=1}^{N} m_s^p \right)^{\frac{1}{p}} \tag{3.10}$$

Here,
M_p = p^{th} power mean function.
m_s = the AUC metrics m calculated for the sub groups.
N = number of identity terms(nine in our experiment).
In our experiment, the value of p is taken as 5 as suggested by Vaidya et al. [53].

3.6 Metrics Selection:

After studying various metrics used to measure the biases in text classification we decided to use the per identity AUC metrics i.e subgroup-AUC, BPSN-AUC and BNSP-AUC to measure the bias mitigation performance of the model over individual identity subgroup and the metric generalised mean AUC to measure the overall bias mitigation performance.

Chapter 4

Experiment

An experiment is conducted to answer the RQ2: How well the proposed model can measure the toxicity by reducing the unintended bias over certain non-toxic identity terms when compared to the previous model?

In this experiment, we will compare BERT based models i.e.. BERT-base and DistilBERT with the Bi-LSTM model proposed in [53]. In this comparative analysis, we are considering three models namely two variants of BERT i.e. BERT-base, DistilBERT and the Bi-LSTM model proposed by Vaidya et al. [53]. Multitask learning is used to train these models so that they can perform two tasks namely identity task and the toxicity task in parallel. Identity task refers to predicting the presence of the identity in the given sentence. Toxicity task refers to predicting the toxicity score for the given sentence. First, we will train all the three models and then we will test these models to compare the bias mitigation performance using the metrics obtained from RQ1.

4.1 Dependent and Independent variables:

Independent variables are the variable that are being changed or controlled in a experiment to test its effect on the dependent variable. Where as dependent variables are the variable that are being tested and measured.

Independent variables: In this experiment the models used in the comparative analysis are the independent variables.

Dependent variables: In this experiment the metrics used to measure the classification and bias mitigation performance of the models are the dependent variables.

4.2 Experiment set-up / Tools used

4.2.1 Software Environment

We have chosen the python programming language as Python has great support for implementing machine learning and deep learning projects and also provides several libraries like NumPy, torch and transformers with which we can build complex deep learning models like LSTM, BERT and so on.

Python Libraries used: The following are the python libraries used

- **PyTorch:** It is an open-source library which is used to program various types of neural networks like LSTM, CNN and so on. It is most widely used in various applications related to computer vision and natural language processing.

- **NumPy:** This library provides the facility to use multidimensional arrays and matrices along with various mathematical functions that can be applied to these arrays.

- **Sci-kit learn:** This is an open-source machine learning library which provides various classification, regression models and also provides data preprocessing techniques.

- **SciPy:** It is an open-source python library used for scientific and technical computing.

- **Pandas:** It is an open-source python library which provides various data manipulation and analysis tools for manipulating tabular data like data sets used for machine learning.

4.2.2 Hardware Environment:

The models used in the comparative analysis ie., BERT-base and DistilBERT are large models, so we need more memory and computational power to train these models, so we had to decide to train these models on cloud platforms. We have used a GPU(Graphics Processing Unit) based compute engine from google cloud to train these models. The following table 4.1 tells about the specifications of the system which is used to train these models

GPU	FP-16: computer engine which has Tesla P100 GPU
OS	Linux
Memory	16gb vRAM
Storage	50gb

Table 4.1: Hardware configuration of the system used to train the model

4.3 Dataset Description:

The dataset that we use for this experiment is taken from Kaggle. This data set is prepared by a Jigsaw research unit with in the google. This database contains 1,804,874 comments in the train set and 97320 comments in the test set which are annotated by the civil comments platform. Each comment is shown at least 10 annotators and these annotations are asked questions like "Is the given sentence toxic or not?", "what genders are refereed in this comment? " and "which races or identity terms are mentioned in this comment?" while annotating the comments. Annotations are asked to select the set of identities present in the comment from a provided list of identities like in the table 4.2. Annotator will give a score of 1 if the label fits the sentence or else he will give a score of 0. The average of scores given by all the annotators is calculated to get the final values for all the labels. The values of these labels range from 0 to 1 which indicates the fraction of annotators who believe that the label fits the given sentence. This data set has a toxicity label called 'target' and nearly 24 identity labels. The toxicity label tells whether the given comment is toxic or not and the identity labels tell whether the given comment contains particular identities or not. The following table 4.2 tells about different identity labels present in the dataset.

Category	**Identity options**
Gender	**Male**, **Female**, Transgender, Other gender
Sexual Oreintation	**Homosexual**, Hetrosexual, Bisexual, Other sexual orientation
Religion	**Christian**, **Jewish, Muslim**, Hindu, Buddhist, Atheist, Other religion
Race or ethnicty	**Black**, **White**, Latino/Latin/Latinx, other race or ethnicty
Disability	**Physical disability**, Intellectual or learn disability, Psychiatric disability or mental illness, Other disability

Table 4.2: List of identity labels mentioned in the dataset

Although the training set contains nearly 24 identity labels we only focus on using 9 identity labels which have more than 500 examples in the test set as we need sufficient examples to test the models [53]. The identity labels that we are using in this study are represented in bold in the table 4.2.

4.4 Data preprocessing:

4.4.1 Adding identity information:

The data set contains nearly 1.8 million instances in which only 0.4 million instances are given with both the toxicity and identity labels, for the remaining 1.4 million

instances only the toxicity labels are given. Among the 0.4 million instances which are labelled with identity labels, only a subset of them have the identities mentioned i.e sentences whose identity labels are grater than or equal to 0.5, the following table shows the percentage of comments corresponding to different identity subgroups in the entire training data.

Identity subgroup	**Percentage**
Male	2.46
Female	2.96
Homosexual	0.60
Christian	2.23
Muslim	0.42
Jewish	1.16
Black	0.82
White	1.38
Psychiatric or mental illness	0.27

Table 4.3: Percentage of comments corresponding to different identity subgroups in the training data

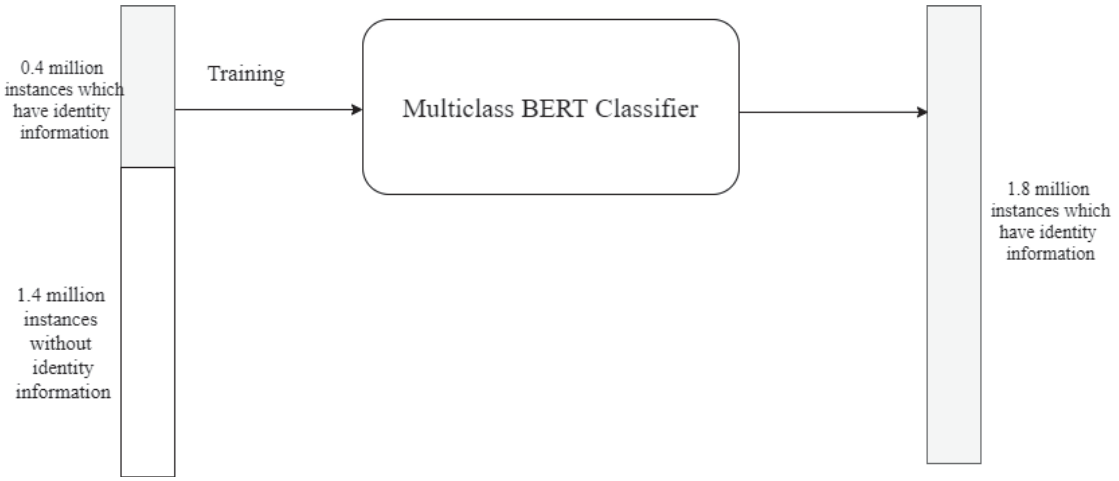

Figure 4.1: Adding the identity information using a multi class BERT classifier.

From the table 4.3, we can see that the percentage of comments having identities mentioned is very less. As we are using multitask learning to reduce the identity bias we need a sufficient amount of training examples which have the identity information. So we had trained a multi-class classifier which is based on BERT-base on the 0.4 million instances which have the identity labels so that it can predict the identity labels for the remaining 1.4 million instances which does not have the identity labels. In the previous study [53] the authors had used a simple LSTM model to predict the identity labels but here we are using BERT-base classifier as it is known to provides better results by the considering contextual information while predict the labels [18].

We had done this pre-processing step based on the previous work done by Vaidya et al. [53].

4.4.2 Tokenizing the sentences:

In all the three model the input text sentences are tokenized into individual tokens and padded to a fixed length of 300. If a sentence has less than 300 tokens then 0 are padded at the end.

4.4.3 Converting the Tokenized words into word vectors:

As we are dealing with deep learning models, the input text can not be given to the models directly, the input sentence which is in text format should be converted into the numerical format. First, the sentences are tokenized into individual tokens as mentioned above and then these tokens are converted into a vectors of numbers called word vectors as shown in the Figure 4.2, which are then fed into the model as inputs.

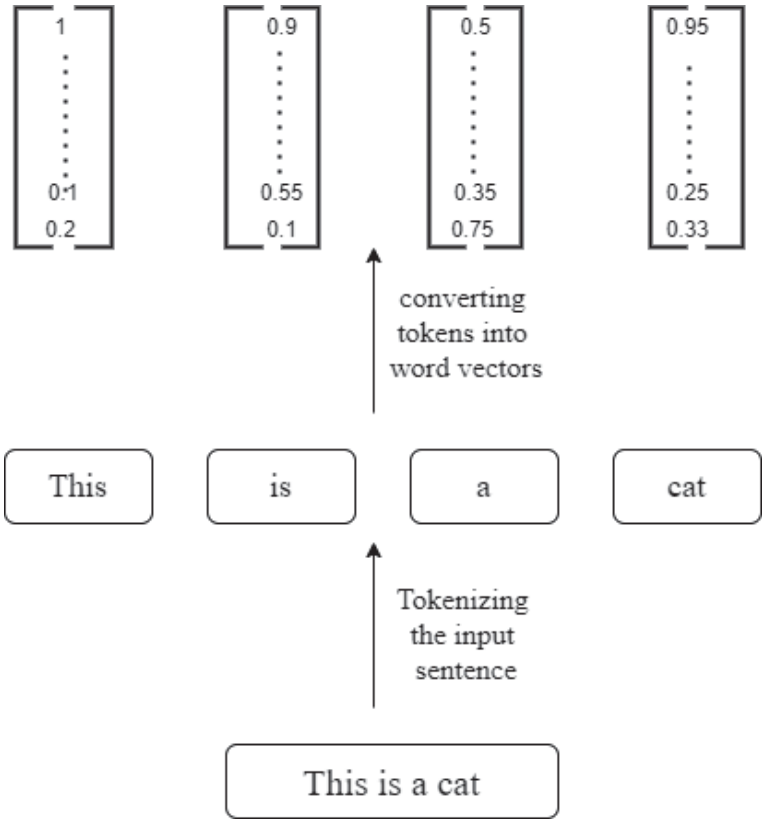

Figure 4.2: Tokenizing the sentence into individual tokens and converting the tokens into word vectors

Bi-LSTM:

After tokenizing the words as described above, the tokenized words are fed as inputs to the embedding layer which uses pre-trained glove embedding [40] for converting the tokenized words into word vectors. The variant of glove embeddings used here is 'glove.6B.300d' which has a vocabulary size of 1.9 million words and for a given word it will provide a word embedding with a dimension of size 300.

BERT-base and DistilBERT:

After tokenizing the sentences using the as discussed above, the stream of tokens is given to the BERT embedding layer which uses word piece embeddings [58] to convert the tokens into vectors. The BERT embedding layer produces three types of embeddings namely token embeddings, segment endings and position embeddings and these embeddings are summed up to get final word embeddings which are then fed as inputs to the next layer [18].

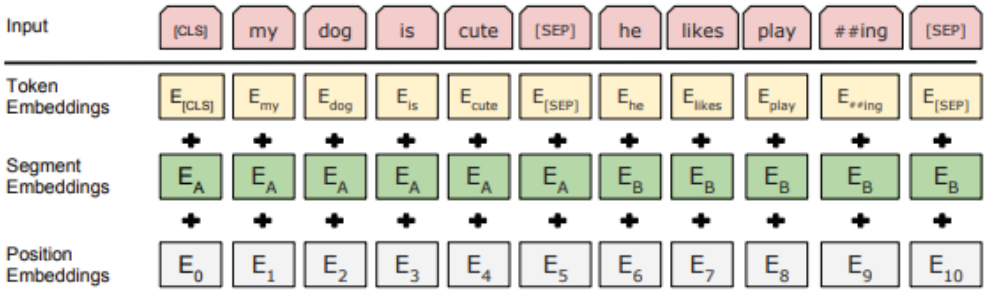

Figure 4.3: Overview of the BERT embedding layer [18].

4.5 Training the models using multi-task learning:

All the three models BERT-base, DistilBERT and Bi-LSTM are trained using multi-task learning to perform two tasks in parallel namely identity task and toxicity task. In toxicity tasks, the model should learn to predict the class label "target" which tells whether the sentence is toxic or not. In identity tasks, the model should learn to predict whether certain identity terms were present in the given sentence or not. In this study we are focusing on reducing identity bias over the nine identity subgroups namely Male , Female, Homosexual, Christian, Jewish, Muslim, Black, White and Physical disability as they at least 500 training examples in the test set. So the identity task is trained to prefect the presence of the nine selected identity subgroups. Vaidya et al.[53] said that involving the identity task with the toxicity task will help the reduce the confusion between the toxic and the identity terms which is the main cause of the identity bias.

In all the three models both the identity task and the toxicity task will share the same network as shown in the Figure 4.4, this will ensure hard parameter sharing as mentioned in [46]. Here the hard parameter sharing is employed because of its

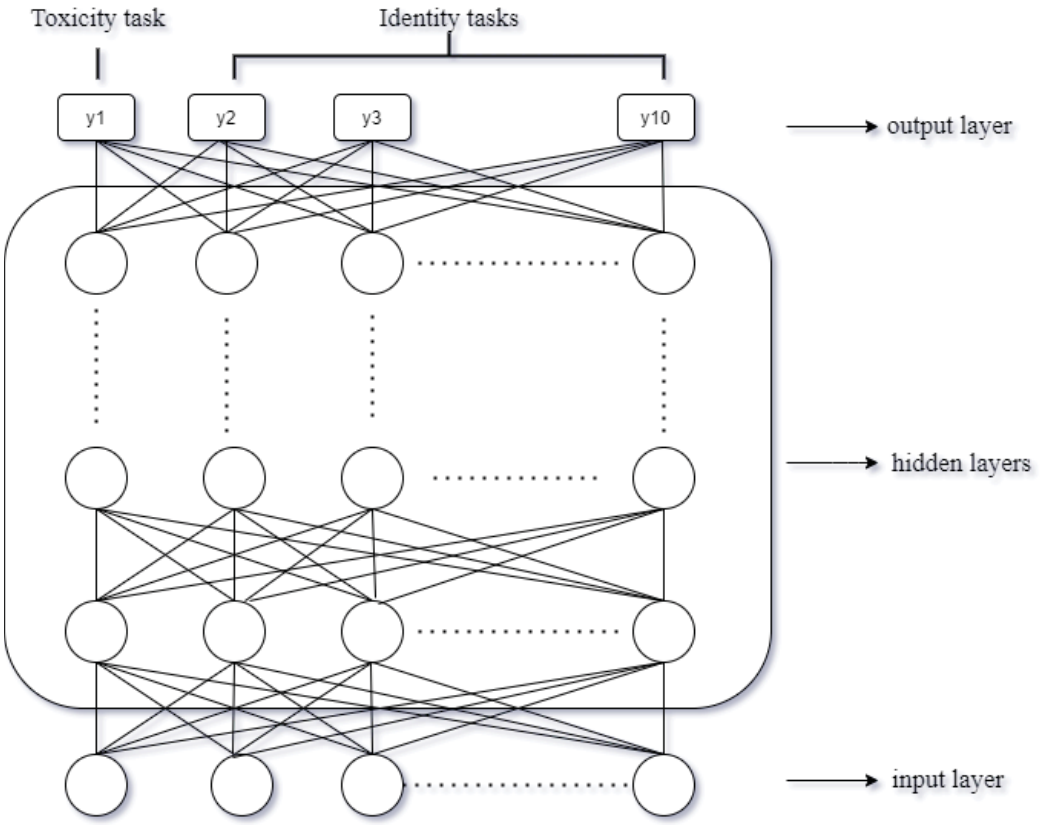

Figure 4.4: Sharing the same network parameters for all the tasks to ensure hard parameter sharing.

ability to reduce over fitting and it is also provides effective way of sharing information between identity task and the toxicity task [46] [53]. In all the three models the input and hidden layers are jointly used by the toxicity task and the identity task, where as the finial out put layer is different for the identity task and the toxicity task. The output layer has ten nodes in which the first node is corresponding to the toxicity prediction and remaining 9 nodes corresponding to the identity predictions.

4.5.1 Using a custom loss Function:

In the training process, we had used a custom loss function similar to the one used in the previous study [53]. The loss function that we used is as follows

$$L = \sum_{n=1}^{N} \beta_n \big[\alpha J_{CE}\big(\hat{y_n}, y_n\big) + \big(1-\alpha\big) \sum_{k=1}^{K} J_{CE}\big(\hat{y_n^k}, y_n^k\big)\big] \tag{4.1}$$

The loss function mentioned above is a weighted cross-entropy loss function. By default, the values of α and β are configured as $\alpha = 0.6$ and $\beta = 1$. But while dealing with nontoxic examples with identity information the value of β is changed to 3. By using this custom loss function which changes according to the training examples the model is able to put more focus on non-toxic examples with have the identity terms

during the training process which helps the model reduce identity bias. In all the three models, the network parameters are trained using back propagation and are optimized using Adam algorithm [28] given its efficiency and its ability to avoid over fitting.

4.5.2 Hyper-parameter tuning:

Hyper-parameter is a parameter whose value is used to control the learning process. The values of these parameters are set before the learning process. These parameters can be tuned and they effect how well a model trains.

Learning rate: It is a hyper-parameter which tells the extent to which the model parameters need to be changed while back-propagating the loss through the network during the training process. The value of the learning rate should be tuned carefully because using a very small rate will slow down the training process whereas selecting a large learning rate will result in a bad parameter tuning while back-propagating the loss through the network.

Batch size: Batch size is the number of training examples the model looks at before updating the model weights using back-propagation. The training data is divided into batches of fixed batch size, then the loss is calculated for every training example in the batch and finally, the total loss is back propagated through the network at once.

Epoch: One epoch means training the model using all the instances in the training data once. Generally, we train the model for multiple epochs to make the model correctly generalize the training data.

	Learning rate	Batch size	Epoch
BERT base	4e-5	128	4
BERT large	4e-5	128	3
Bi-LSTM	2e-5	32	3

Table 4.4: Hyper-parameter tuning in all the three models

4.6 Testing the Models:

This section discusses the metrics and methods used to test the classification and bias mitigation performance of three models selected in this study.

4.6.1 Classification performance:

To measure the overall classification performance of the models we have used the metrics f1 score and Overall-AUC i.e AUC [21] calculated on the entire test set. The test set is imbalanced i.e only 7.9% of instances belong to toxic class and the remaining 82.1 belong to non-toxic class. So we had chosen the metrics overall-AUC and f1-score as they are not effected by the unbalanced class distribution. The overall-AUC is a threshold agnostic metric so we can apply it directly on the model's prediction. But to calculate the metric f1-score we need to model's predictions into binary format and the corresponding ground truth values into 0 or 1 using a threshold of 0.5.

- **Precision:** It is the proportion of true positives that were correctly predicted by the model to the total number of positive predictions that were made by the model.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositive} \tag{4.2}$$

- **Recall:** It is the proportion of true positives that were correctly predicted by the model.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{4.3}$$

- **False positive rate:** It is the proportion of negative predictions that were wrongly classified as false positives by the model.

$$Recall = \frac{FalsePositives}{FalsePositives + TrueNegatives} \tag{4.4}$$

- **F1-score:** F1-score is the harmonic mean of precision and recall and is defined as follows

$$F1\text{-}score = \frac{2 * precision * recall}{precision * recall} \tag{4.5}$$

- **AUC:** AUC is the area under the ROC(Receiver Operating Characteristics) curve [19]. ROC curve of a classifier is the obtained by plotting true positive rates(on the y-axis) and false-positive rates(on the x-axis) of the classifier at various thresholds as shown in the graph 4.5. Now calculating the area under it will give the AUC measure. If a classifier has an AUC score close to 1 then it is doing a good job of correctly classifying positive and negative classes and if a classifier has an AUC close to 0 then it is doing a bad job of classifying positive and negative classes.

4.6.2 Bias mitigation performance:

For measuring the bias mitigation performance over different identity subgroups we have used the AUC based metrics that were selected in the literature review i.e

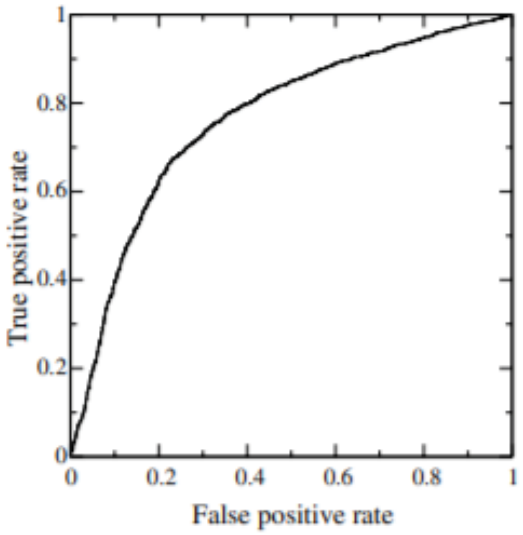

Figure 4.5: ROC cure of a hypothetical classifier [19]

subgroup-AUC, BPSN-AUC, BNSP-AUC and Generalized mean AUC.

Subgroup-AUC: For calculating this measure, a sample set which contains an equal number of toxic and non-toxic examples corresponding to particular identity subgroup is selected from the test set. Calculating the AUC metric on this sample set will give us the get subgroup AUC for the particular identity. The subgroup-AUC will tell haw well the model is able to distinguish positive and negative sentences corresponding to a particular identity subgroup. The subgroup AUC for all the nine identity terms is calculated and recorded.

BPSN-AUC: For calculating this measure, a sample set with an equal number of toxic and non-toxic examples is sampled from the test set. Here the non-toxic examples will have a particular identity mentioned but the toxic examples will not have any identity mentioned. Calculating the AUC metric on this sample set will give us the BPSN-AUC corresponding to the particular identity. The BPSN-AUC will tell the extent to which a model is able to separate the non-toxic examples corresponding identity subgroup from the toxic examples corresponding to background data which resembles the original distribution of test set. The BPSN-AUC for all the nine identity items is calculated and recorded.

BNSP-AUC: For calculating this measure, a sample set with an equal number of toxic and non-toxic examples is sampled from the test set. Here the toxic examples will have a particular identity mentioned but the non-toxic examples will not have any identity mentioned. Calculating the AUC metric on this sample set will give us the BNSP-AUC corresponding to the particular identity. The BNSP-AUC will tell the extent to which a model is able to separate the toxic examples from identity subgroup and the non-toxic examples from background data which resembles the original distribution of test set. The BPSN-AUC for all the nine identity items is

calculated and recorded.

Generalised mean AUC: After calculating the per-identity metrics(namely subgroup-AUC, BPSN-AUC and BNSP-AUC) for all the nine identity subgroups, we will calculate the generalised mean-AUC as mentioned in the section 3.5.3.

Chapter 5

Results

In this section we will discuss the results obtained from the experiment. First, we will discuss the classification performance of the models i.e f1 score and Overall-AUC. Finally, we will discuss the bias mitigation performance of all the three models used in this experiment.

5.1 Classification Performance:

The table 5.1 presents the values for the metrics f1-score and overall-AUC of all the three models calculated on the entire test set.

Model	**F1-score**	Overall-AUC
BERT base	0.674	0.962
DistilBERT	0.665	0.955
Bi-LSTM	0.588	0.933

Table 5.1: F1-score and Overall-AUC for all the three models

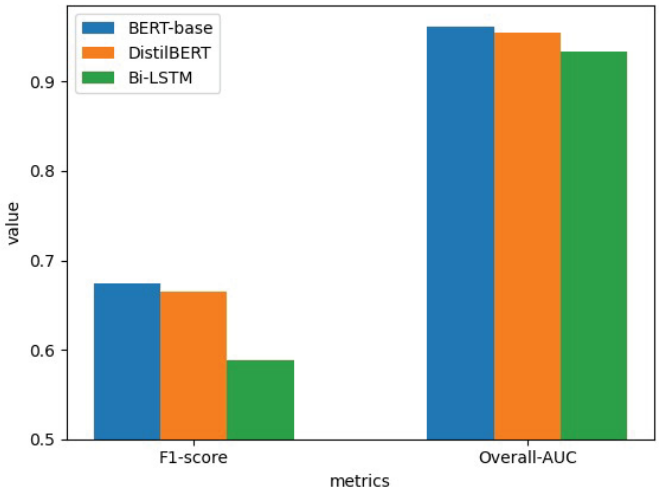

Figure 5.1: Bar plot showing the F1-scores, Overall-AUC of all the models

5.2 Bias mitigation performance:

In this section, we will look at the bias mitigation performance of all the three models over the nine identity subgroup that are chosen in this study. First, we will look at the values of the metrics subgroup-AUC, BPSN-AUC, BSPN-AUC corresponding to individual identity subgroup and then will look at the Generalised mean AUC for all the three models.

5.2.1 Male:

The toxic and non-toxic examples corresponding to the identity "male" are sampled as mentioned in the section 3.5.1 and the metrics subgroup-AUC, BPSN-AUC and BNSP-AUC for all the three models were calculated and presented in the table 5.2. The figures 5.2, show the barplots of subgroup-AUC, BPSN-AUC, BNSP-AUC respectively for all the three models.

Model	Subgroup-AUC	BPSN-AUC	BNSP-AUC
BERT base	0.908375	0.930723	0.943065
distill-BERT	0.906260	0.924483	0.946060
Bi-LSTM	0.883052	0.899050	0.885417

Table 5.2: Values of per-identity AUC metrics for identity subgroup 'male'

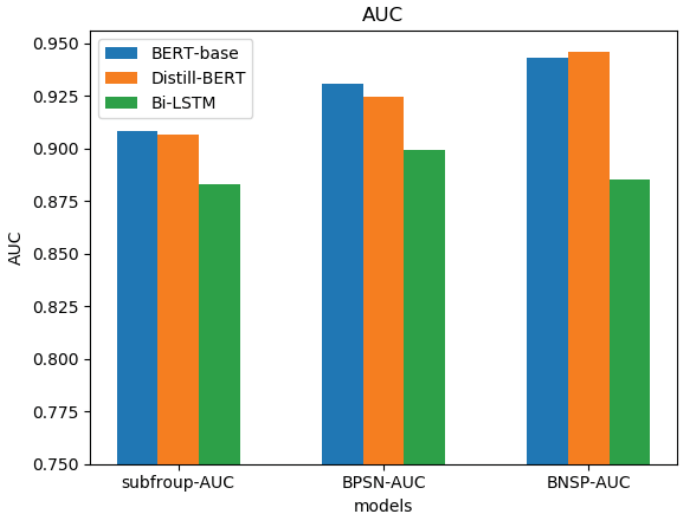

Figure 5.2: Bar plot showing the per-identity matrix for identity subgroup "Male"

5.2.2 Female:

The toxic and non-toxic examples corresponding to the identity “female” are sampled as mentioned in the section 3.5.1 and the metrics subgroup AUC, BPSN-AUC and BNSP-AUC for all the three models were calculated and presented in the table 5.3. The figures 5.3, show the barplots of subgroup-AUC, BPSN-AUC, BNSP-AUC respectively for all the three models.

Model	Subgroup-AUC	BPSN-AUC	BNSP-AUC
BERT base	0.907883	0.956751	0.933006
distill-BERT	0.901316	0.943760	0.929469
Bi-LSTM	0.888367	0.915755	0.877870

Table 5.3: Values of per-identity AUC metrics for identity subgroup ’female’

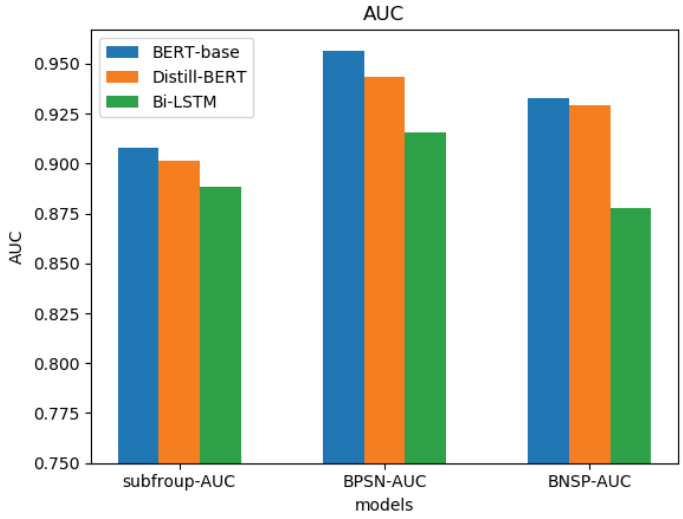

Figure 5.3: Bar plot showing the per-identity matrix for identity subgroup ”Female”

5.2.3 Homosexual:

The toxic and non-toxic examples corresponding to the identity “Homosexual” are sampled as mentioned in the section 3.5.1 and the metrics subgroup-AUC, BPSN-AUC and BNSP-AUC for all the three models were calculated and presented in the table 5.4. The figures 5.4 show the barplots of subgroup-AUC, BPSN-AUC, BNSP-AUC respectively for all the three models.

Model	Subgroup-AUC	BPSN-AUC	BNSP-AUC
BERT base	0.791466	0.863375	0.943098
distill-BERT	0.804273	0.858207	0.946267
Bi-LSTM	0.787254	0.821421	0.899710

Table 5.4: Values of per-identity AUC metrics for identity subgroup 'homosexual'

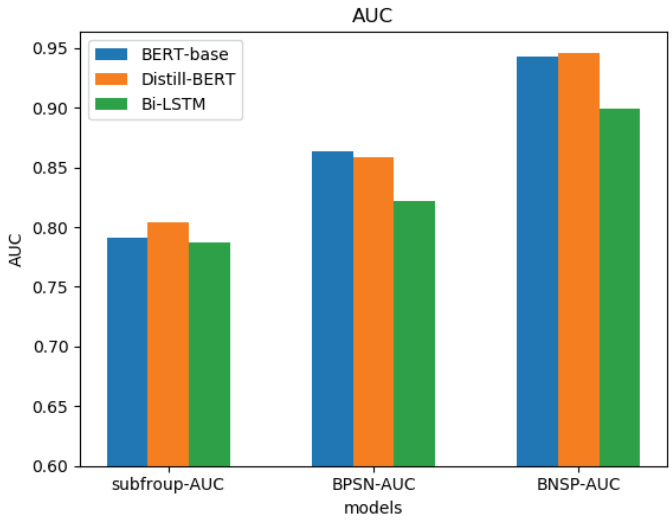

Figure 5.4: Bar plot showing the per-identity matrix for identity subgroup "Homosexual"

5.2.4 Christian:

The toxic and non-toxic examples corresponding to the identity "christian" are sampled as mentioned in the section 3.5.1 and the metrics subgroup-AUC, BPSN-AUC and BNSP-AUC for all the three models were calculated and presented in the table 5.5. The figures 5.5 show the barplots of subgroup-AUC, BPSN-AUC, BSPN-AUC respectively for all the three models.

Model	Subgroup-AUC	BPSN-AUC	BNSP-AUC
BERT-base	0.917633	0.966464	0.900105
distill-BERT	0.928317	0.933692	0.944349
Bi-LSTM	0.907048	0.919530	0.862651

Table 5.5: Values of per-identity AUC metrics for identity subgroup 'christian'

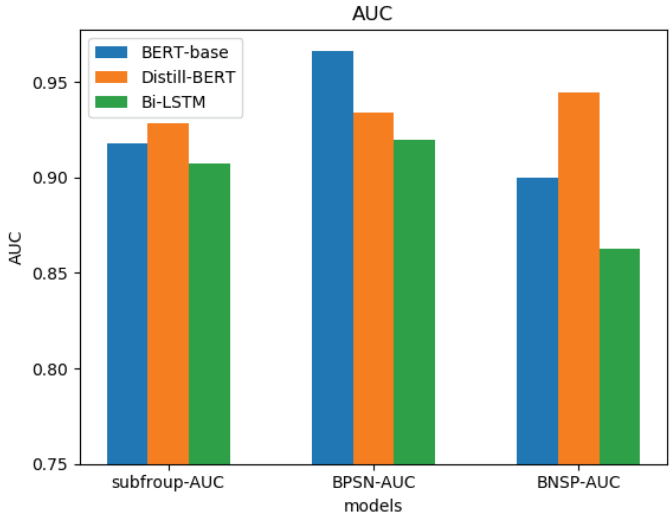

Figure 5.5: Bar plot showing the per-identity matrix for identity subgroup "Christian"

5.2.5 Jewish:

The toxic and non-toxic examples corresponding to the identity "jewish" are sampled as mentioned in the section 3.5.1 and the metrics subgroup AUC, BPSN-AUC and BNSP-AUC for all the three models were calculated and presented in the table 5.6. The figures 5.6 show the barplots of subgroup-AUC, BPSN-AUC, BSPN-AUC respectively for all the three models.

Model	**Subgroup-AUC**	**BPSN-AUC**	**BNSP-AUC**
BERT-base	0.909221	0.919708	0.944281
Distill-BERT	0.894693	0.920513	0.936529
Bi-LSTM	0.884968	0.897642	0.913323

Table 5.6: Values of per-identity AUC metrics for identity subgroup 'jewish'

5.2.6 Muslim:

The toxic and non-toxic examples corresponding to the identity "muslim" are sampled as mentioned in the section 3.5.1 and the metrics subgroup AUC, BPSN-AUC and BNSP-AUC for all the three models were calculated and presented in the table 5.7. The figures 5.7 show the barplots of subgroup-AUC, BPSN-AUC, BNSP-AUC respectively for all the three models.

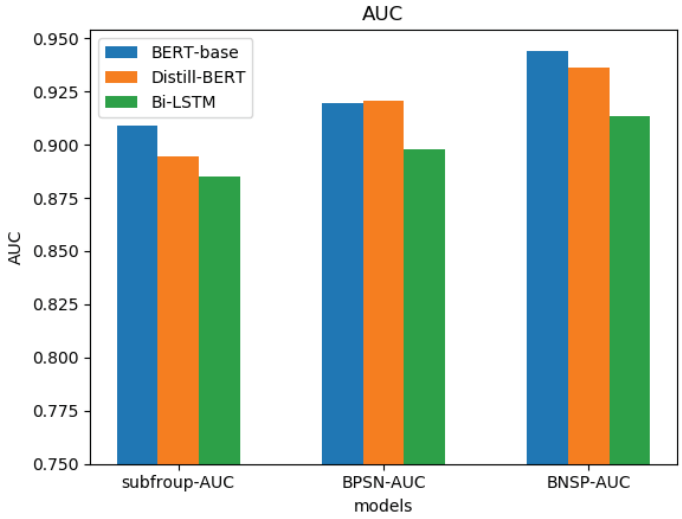

Figure 5.6: Bar plot showing the per-identity matrix for identity subgroup "Jewish"

Model	Subgroup-AUC	BPSN-AUC	BNSP-AUC
BERT-base	0.866421	0.887160	0.955368
Distill-BERT	0.859747	0.891036	0.947040
Bi-LSTM	0.827166	0.865009	0.912416

Table 5.7: Values of per-identity AUC metrics for identity subgroup 'Muslim'

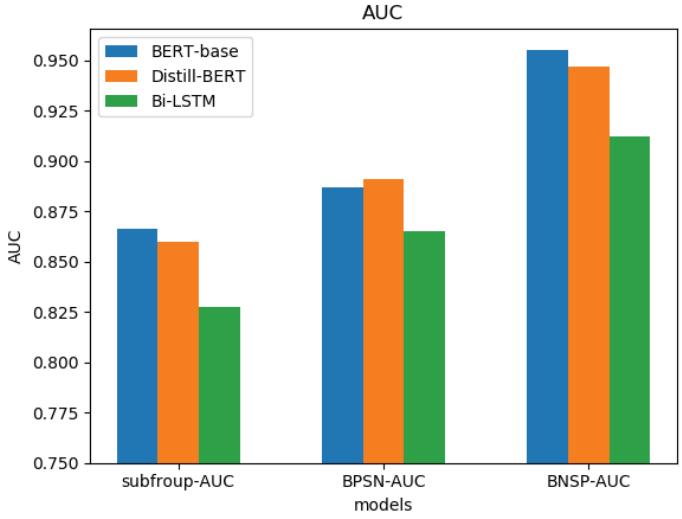

Figure 5.7: Bar plot showing the per-identity matrix for identity subgroup "Muslim"

5.2.7 Black:

The toxic and non-toxic examples corresponding to the identity "black" are sampled as mentioned in the section 3.5.1 and the metrics subgroup-AUC, BPSN-AUC and

BNSP-AUC for all the three models were calculated and presented in the table 5.8. The figures 5.8 show the barplots of subgroup-AUC, BPSN-AUC, BNSP-AUC respectively for all the three models.

Model	Subgroup-AUC	BPSN-AUC	BNSP-AUC
BERT-base	0.817610	0.853244	0.953521
DistilBERT	0.837407	0.849607	0.957810
Bi-LSTM	0.804104	0.835072	0.919536

Table 5.8: Values of per-identity AUC metrics for identity subgroup 'black'

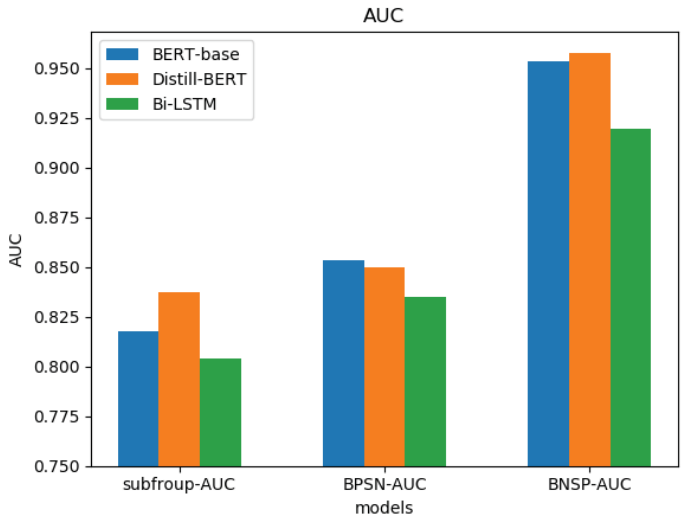

Figure 5.8: Bar plot showing the per-identity matrix for identity subgroup "Black"

5.2.8 White:

The toxic and non-toxic examples corresponding to the identity "white" are sampled as mentioned in the section 3.5.1 and the metrics subgroup-AUC, BPSN-AUC and BNSP-AUC for all the three models were calculated and presented in the table 5.9. The figures 5.9 show the barplots of subgroup-AUC, BPSN-AUC, Bnsp-AUC respectively for all the three models.

Model	Subgroup-AUC	BPSN-AUC	BNSP-AUC
BERT-base	0.849023	0.878084	0.959008
Distill-BERT	0.848352	0.871324	0.95258
Bi-LSTM	0.821289	0.858401	0.922700

Table 5.9: Values of per-identity AUC metrics for identity subgroup 'white'

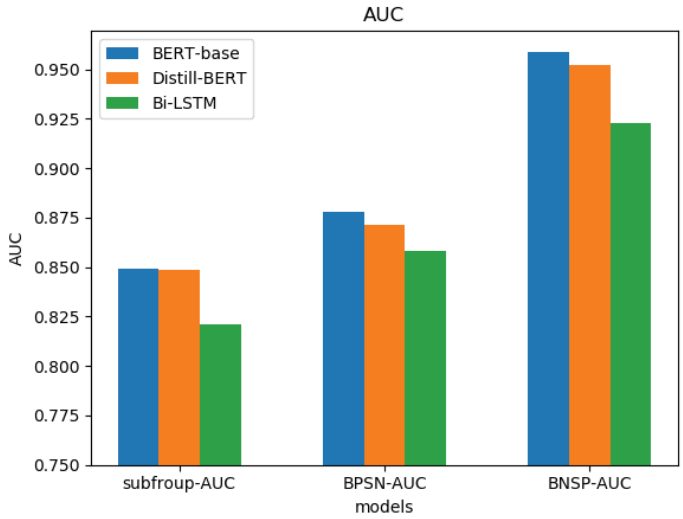

Figure 5.9: Bar plot showing the per-identity matrix for identity subgroup "White"

5.2.9 Physical disability:

The toxic and non-toxic examples corresponding to the identity "physical disability" are sampled as mentioned in the section 3.5.1 and the metrics subgroup-AUC, BPSN-AUC and BNSP-AUC for all the three models were calculated and presented in the table 5.10. The figures 5.10 show the barplots of subgroup-AUC, BPSN-AUC, BNSP-AUC respectively for all the three models.

Model	Subgroup-AUC	BPSN-AUC	BNSP-AUC
BERT-base	0.860773	0.912598	0.935872
DistilBERT	0.863383	0.856766	0.962448
Bi-LSTM	0.821622	0.846204	0.905946

Table 5.10: Values of per-identity AUC metrics for identity subgroup 'physical disability'

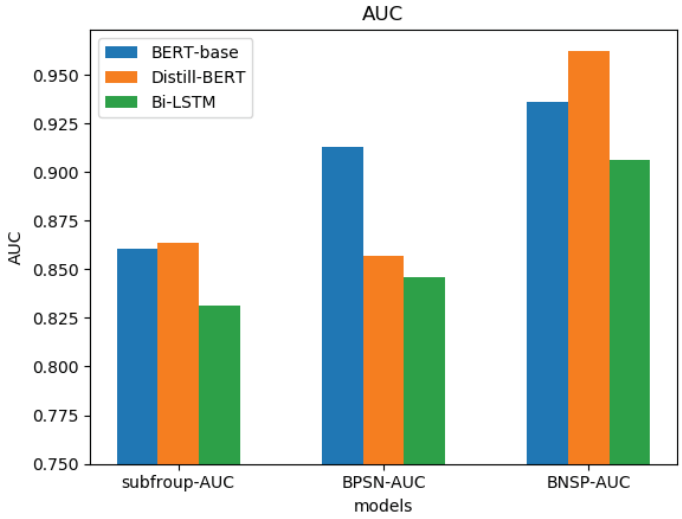

Figure 5.10: Bar plot showing the per-identity matrix for identity subgroup "Physical disability"

5.2.10 Generalised mean AUC:

The generalised mean AUC for all the four models is presented in the table 5.11. The figure 5.11 shows the barplots of generalised mean-AUC for all the three models.

Model	Generalised mean AUC
BERT-base	0.9047
DistilBERT	0.9007
Bi-LSTM	0.8622

Table 5.11: Values of generalised mean AUC for all the models

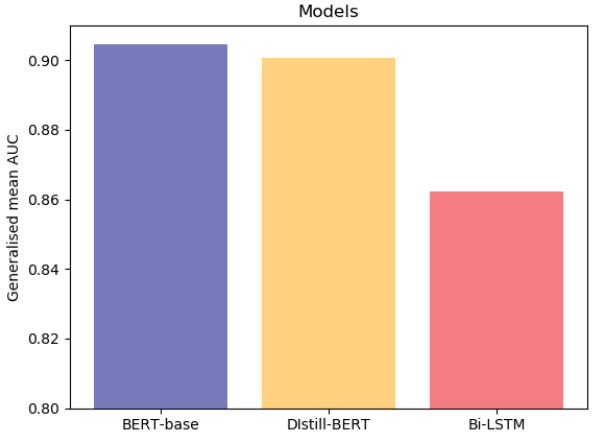

Figure 5.11: Bar plot showing the per-identity matrix for identity subgroup "Generalised AUC Mean".

5.3 Significance Test

In order to prove that the proposed BERT models i.e. BERT-base and DistilBERT provide significant improvement in bias mitigation performance than the previous Bi-LSTM model we had performed Friedman test and Nemenyi test using the guidelines provided by Peter Flach [22]. We had performed the significance test on the results of subgroup-AUC, BPSN-AUC, BNSP-AUC corresponding to nine identity subgroups.

5.3.1 Friedmans test:

Friedmans test is used to know whether all the algorithms perform similarly or not. First, we will formulate the null hypothesis as "All the algorithms perform similarly". For every sample(identity subgroup) the algorithms are ranked according to their scores and average ranks for all the algorithms are calculated. Then the Friedmans statistic is calculated as mentioned by Peter Flach [22]. Then we will compare the Friedmans statistic with the critical value. If the value of the Friedmans statistic is less than the critical value then will accept the null hypothesis or else we will reject the null hypothesis.

5.3.2 Nemenyi test:

Friedmans test can only tell whether algorithms perform similarly or not, it can not specify which of the algorithm is performing differently from the others. To find that a post-hoc test called the Nemenyi test is performed. The pair-wise average rank differences for all the algorithms are calculated and are compared with the critical difference. If any of the pair-wise rank differences exceeds the critical difference then we can say that those two algorithms are performing differently.

5.3.3 Subgroup-AUC:

Null hypothesis: All the algorithms perform similarly in terms of subgroup-AUC.

Identity subgroup	BERT-base	DistilBERT	Bi-LSTM
Male	0.908375(1)	0.906260(2)	0.883052(3)
Female	0.907883(1)	0.901316(2)	0.883670(3)
Homosexual	0.791466(2)	0.804273(1)	0.787254(3)
Christian	0.917633(2)	0.928317(1)	0.907048(3)
Muslim	0.909211(1)	0.894693(2)	0.884968(3)
Jewish	0.866421(1)	0.859747(2)	0.827166(3)
Black	0.817610(2)	0.837407(1)	0.804104(3)
White	0.849023(1)	0.848352(2)	0.821289(3)
Psychiatric or mental illness	0.860773(2)	0.863383(1)	0.821622(3)
Average Rank	1.44	1.55	3

Table 5.12: Subgroup-AUC for all the nine identity subgroups

From the table 5.12
Number of samples n = 9
Degrees of freedom k =3
Significance level $\alpha = 0.05$

The Friedmans statistic calculated on the subgroup AUC is 13.5 which is greater than the critical value 7.9 (n = 9, k = 3 and $\alpha = 0.05$) so the null hypothesis:" All the algorithms perform similarly in terms of subgroup-AUC" can be rejected. As the null hypothesis is rejected we can perform the Nemenyi test. The table 5.13 shows the pairwise average rank differences between all the three algorithms.

Algorithm Pair	Rank difference
BERT-base and DistilBERT	0.11
BERT-base and Bi-LSTM	1.56
DistilBERT and Bi-LSTM	1.45

Table 5.13: Pairwise average rank difference for subgroup AUC

From the table 5.13 is evident that the average rank difference between BERT-base and DistilBERT is 0.11 which is less than the critical difference 1.047 (for n = 9, k = 3 and alpha = 0.05) so we can say that these algorithms are performing similarly. Also, the average rank difference BERT-base and Bi-LSTM is 1.56 where as it is 1.45 in the case of DistilBERT and Bi-LSTM. In both cases, the rank differences are greater than the critical difference 1.047. Thus, we can say that Bi-LSTM is performing differently from BERT-base and DistilBERT. So we can conclude that BERT-base and DistilBERT are offering significantly better performance in terms of subgroup-AUC when compared to Bi-LSTM.

5.3.4 BPSN-AUC:

Null hypothesis: All the algorithms perform similarly in terms of subgroup-AUC

Identity subgroup	BERT-base	DistilBERT	Bi-LSTM
Male	0.930723(1)	0.924483(2)	0.899050(3)
Female	0.956751(1)	0.943760(2)	0.915755(3)
Homosexual	0.863375(1)	0.858207(2)	0.821421(3)
Christian	0.966464(1)	0.933692(2)	0.919530(3)
Muslim	0.919708(2)	0.920513(1)	0.897642(3)
Jewish	0.887160(2)	0.891036(1)	0.865009(3)
Black	0.853244(1)	0.849607(2)	0.835072(3)
White	0.878084(1)	0.871324(2)	0.858401(3)
Psychiatric or mental illness	0.912598(1)	0.856766(2)	0.846204(3)
Average Rank	1.22	1.77	3

Table 5.14: BPSN-AUC for all the nine identity subgroups

From the table 5.14,
Number of samples n = 9
Degrees of freedom k =3
Significance level $\alpha = 0.05$

The Friedmans statistic calculated on the subgroup AUC is 13.5 which is greater than the critical value 7.9 (n = 9, k = 3 and $\alpha = 0.05$) so the null hypothesis:" All the algorithms perform similarly in terms of BNSP-AUC" can be rejected. As the null hypothesis is rejected we can perform the Nemenyi test. The following table shows the pairwise average rank differences between all the three algorithms.

Algorithm Pair	Rank difference
BERT-base and DistilBERT	0.55
BERT-base and Bi-LSTM	1.78
DistilBERT and Bi-LSTM	1.23

Table 5.15: Pairwise average rank difference for BPSN AUC

From the table 5.15 is evident that the average rank difference between BERT-base and DistilBERT is 0.55 which is less than the critical difference 1.047 (for n = 9, k = 3 and alpha = 0.05) so we can say that these algorithms are performing similarly. Also, the average rank difference BERT-base and Bi-LSTM is 1.78 whereas it is 1.23 in the case of DistilBERT and Bi-LSTM. In both cases, the rank differences are greater than the critical differenceof 1.047. Thus, we can say that Bi-LSTM is performing differently from BERT-base and DistilBERT. So we can conclude that BERT-base and DistilBERT are offering significantly better performance in terms of BPSN-AUC when compared to Bi-LSTM.

5.3.5 BNSP-AUC:

Null hypothesis: All the algorithms perform similarly in terms of subgroup-AUC

Identity subgroup	BERT-base	DistilBERT	Bi-LSTM
Male	0.943065(2)	0.946060(1)	0.885417(3)
Female	0.933006(1)	0.929469(2)	0.877870(3)
Homosexual	0.943098(2)	0.946267(1)	0.899710(3)
Christian	0.900105(2)	0.944349(1)	0.862651(3)
Muslim	0.944281(1)	0.936529(2)	0.913323(3)
Jewish	0.955368(1)	0.947040(2)	0.912416(3)
Black	0.953521(2)	0.957810(1)	0.919536(3)
White	0.959008(1)	0.952580(2)	0.922700(3)
Psychiatric or mental illness	0.935872(2)	0.962448(1)	0.905946(3)
Average Rank	1.55	1.44	3

Table 5.16: BNSP-AUC for all the nine identity subgroups

From the table 5.16
Number of samples n = 9
Degrees of freedom k =3
Significance level $\alpha = 0.05$

The Friedmans statistic calculated on the subgroup AUC is 13.5 which is greater than the critical value 7.9 (n = 9, k = 3 and alpha = 0.05) so the null hypothesis:" All the algorithms perform similarly in terms of BNSP AUC" can be rejected. As the null hypothesis is rejected we can perform the Nemenyi test. The following table shows the pairwise average rank differences between all the three algorithms.

Algorithm Pair	Rank difference
BERT-base and DistilBERT	0.11
BERT-base and Bi-LSTM	1.45
DistilBERT and Bi-LSTM	1.56

Table 5.17: Pairwise average rank difference for BNSP-AUC

From the table 5.17 is evident that the average rank difference between BERT-base and DistilBERT is 0.11 which is less than the critical difference 1.047 (for n = 9, k = 3 and $\alpha = 0.05$) so we can say that these algorithms are performing similarly. Also, the average rank difference BERT-base and Bi-LSTM is 1.45 whereas it is 1.56 in the case of DistilBERT and Bi-LSTM. In both cases, the rank differences are greater than the critical difference of 1.047. Thus, we can say that Bi-LSTM is performing differently from BERT-base and DistilBERT. So we can conclude that BERT-base and DistilBERT are offering significantly better performance in terms of BPSN AUC when compared to Bi-LSTM.

Chapter 6

Analysis and Discussion

6.1 Analysis of Literature Review

Answer to the RQ1: "What metrics are best suitable to measure the bias mitigation performance of the proposed model and why?" is obtained by conducting a literature review. After studying various metrics used in the past for measuring biases in text classification models, the metrics subgroup-AUC, BPSN-AUC, BSPN-AUC and Generalised mean AUC are selected to measure the bias mitigation performance in the text classification models which are considered in our comparative analysis.

6.2 Analysis of Experiment

6.2.1 Classification performance:

On observing the results in the section 5.1 it is evident that BERT-base has the highest classification performance followed by Distill-BERT, then followed by Bi-LSTM in terms of the all tow metrics f1-score and overall-AUC. It is important to note that the f1-scores of BERT-base and Distil-BERT differs from Bi-LSTM with a considerable margin i.e. F1-scores for BERT-base and Distill-BERT are 0.674 and 0.665 respectively, whereas Bi-LSTM model scores a much lower F-1 score of 0.588.

6.2.2 Bias mitigation performance:

Per identity Metrics:

On analysing the results from the section 5.2 it is evident that the BERT based models i.e BERT-base and DistilBERT have a higher subgroup-AUC, BPSN-AUC and BSPN-AUC for all the nine identity subgroups when compared to Bi-LSTM model. Also the results from the Friedman's and Nemenyi test presented in the section 5.3 makes it clear that that the performance difference is significant.

Generalised mean AUC

The BERT based models have a higher Generalised mean AUC when compared to the Bi-LSTM model i.e. The BERT-base and DistilBERT had a generalised mean AUC of 0.9047 and 0.9007 whereas Bi-LSTM had scored a lower Generalised mean AUC of 0.8622.

6.3 Discussion

We have conducted an experiment to answer the RQ2: How well the proposed model can measure the toxicity by reducing the unintended bias over certain non-toxic identity terms when compared to the previous model? To answer this question we had trained the Bi-LSTM model proposed by Vaidya et al. [53] and BERT based models i.e BERT-base and Distill-BERT and extracted the metrics as mentioned in the section 4.6. We then used the results to analyse the classification and bias mitigation performance of the three models.

On observing the results in the section 5.1, it is evident that the BERT based models i.e BERT-base and DistilBERT are providing a better classification performance when compared to Bi-LSTM model. Similarly a closer examination on per identity metrics(subgroup-AUC, BPSN-AUC and BNSP-AUC) and the Generalised mean AUC for all the three models tells that the BERT based models are providing a better bias mitigation performance when compared to the Bi-LSTM model.

The higher scores of BPSN-AUC for BERT based models in all identity subgroups indicated that the BERT based models are able to reduce the false positive bias more effectively when compared to the Bi-LSTM model. Whereas the higher scores of BNSP-AUC for BERT based models implies the BERT based models are able to reduce the false negative bias more effectively when compared to the Bi-LSTM model. The higher value of Generalised mean AUC indicates that the overall bias mitigation performance of BERT based models is better when compared to the previous Bi-LSTM model proposed in the paper [53].

6.4 Validity Threats

This section tells about the validity threats that are identified and the mitigation methods applied to nullify these threats.

6.4.1 Internal validity:

Internal validity refers to how well the research had been conducted. Internal validity threats that may occur in our thesis are using inappropriate data for training the models and using inefficient hyper parameter settings while training the models. In order to mitigate these threats we had prepossessed the data according to our problem and only used the data which is necessary for the experiment. We also used different configurations for the hyper parameters and finalised the one which yields the best performance.

6.4.2 External validity:

External validity refers to the extent to which the results obtained in the thesis can be generalized to the different real world scenarios. In this study the models are

trained on a data set which contains real world text conversations, so the results of this thesis can be easily generalised in various other scenarios.

6.4.3 Conclusion validity:

Conclusion validity refers to the threats that may occur due to the application of inappropriate metrics while extracting the results from the experiment. In order to overcome this threats we had conducted a literature review on various metrics used to measure the bias in text classification and then selected the appropriate metrics with suits our case.

Chapter 7

Conclusion and Future work

7.1 Conclusion

Over years many deep learning and machine learning models have been proposed to detect the toxicity in text conversations. In this thesis study we focus on using attention based models like BERT-base and DistilBERT for reducing identity bias in toxicity classification. The BERT based models are trained using multitask learning in such a way that they can predict the toxicity score along with the identity terms present in a given comment. After training the models, the BERT based models are compared with the previous Bi-LSTM model proposed in the paper [53] in terms of classification and bias mitigation performance. From the results we can conclude that BERT based models offer better classification and Bias mitigation performance when compared to the Bi-LSTM model proposed in the paper [53].

7.2 Future work

In this thesis we had only focused on reducing identity bias over a selected identity sub groups due to the lack of data for training and testing the models. Also, the data set used in this thesis provides insufficient identity information i.e only a subset of comments are presented with identity labels. In order to provide the models with sufficient training data, we had trained a multi-class classifier on the subset of data which had the identity information and used it to predict the identity labels for the remaining data points which do not have identity information. So, a viable future extension of this thesis work is using a richer data set which contains comments related to more identity subgroups or using synthetic data preparation methods to generate more data for training the models.

Bibliography

[1] S. Babar and P. D. Patil, "Improving performance of text summarization," *Procedia Computer Science*, vol. 46, pp. 354–363, 2015.

[2] J. Baxter, "A bayesian/information theoretic model of learning to learn via multiple task sampling," *Machine learning*, vol. 28, no. 1, pp. 7–39, 1997.

[3] M. Bogen and A. Rieke, "Help wanted: An examination of hiring algorithms, equity, and bias," 2018.

[4] D. Borkan, L. Dixon, J. Li, J. Sorensen, N. Thain, and L. Vasserman, "Limitations of pinned auc for measuring unintended bias," *arXiv preprint arXiv:1903.02088*, 2019.

[5] D. Borkan, L. Dixon, J. Sorensen, N. Thain, and L. Vasserman, "Nuanced metrics for measuring unintended bias with real data for text classification," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 491–500.

[6] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[7] E. Charniak, *Introduction to artificial intelligence*. Pearson Education India, 1985.

[8] K. Chowdhary, "Natural language processing," in *Fundamentals of Artificial Intelligence*. Springer, 2020, pp. 603–649.

[9] S. Chris and M. Najafian, "Reducing sentiment polarity for demographic attributes in word embeddings using adversarial learning," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 359–368.

[10] P. Cimiano, C. Unger, and J. McCrae, "Ontology-based interpretation of natural language," *Synthesis Lectures on Human Language Technologies*, vol. 7, no. 2, pp. 1–178, 2014.

[11] K. Clark, U. Khandelwal, O. Levy, and C. Manning, "What does bert look at? an analysis of bert's attention," 01 2019, pp. 276–286.

[12] L. Cohen, Z. C. Lipton, and Y. Mansour, "Efficient candidate screening under multiple tests and implications for fairness," *arXiv preprint arXiv:1905.11361*, 2019.

[13] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 160–167.

[14] J. M. Conroy and D. P. O'leary, "Text summarization via hidden markov models," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001, pp. 406–407.

[15] S. Corston-Oliver, E. Ringger, M. Gamon, and R. Campbell, "Task-focused summarization of email," in *Text Summarization Branches Out*, 2004, pp. 43–50.

[16] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: A new paradigm to machine learning," *Archives of Computational Methods in Engineering*, pp. 1–22, 2019.

[17] L. Deng and Y. Liu, *Deep learning in natural language processing*. Springer, 2018.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[19] L. Dixon, J. Li, J. Sorensen, N. Thain, and L. Vasserman, "Measuring and mitigating unintended bias in text classification," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 67–73.

[20] M. Duggan, "Online harassment 2017," 2017.

[21] T. Fawcett, "An introduction to roc analysis: Pattern recognition letter, v. 27," 2006.

[22] P. Flach, *Machine learning: the art and science of algorithms that make sense of data*. Cambridge University Press, 2012.

[23] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Advances in neural information processing systems*, 2016, pp. 3315–3323.

[24] P. Harrington, *Machine learning in action*. Manning Publications Co., 2012.

[25] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS Deep Learning and Representation Learning Workshop*, 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

[26] X. Hu and H. Liu, "Text analytics in social media," in *Mining text data*. Springer, 2012, pp. 385–414.

[27] D. Kapashi and P. Shah, "Answering reading comprehension using memory networks," *Report for Stanford University Course cs224d*, 2015.

[28] D. Kingma and J. Ba, "Adam: A method for stochastic optimization in: Proceedings of the 3rd international conference for learning representations (iclr'15)," *San Diego*, 2015.

[29] J. L. L. J. A. Kirchner and S. Mattu, "Machine bias: Thereâăźs software used across the country to predict future criminals. and itâăźs biased against blacks.(may 2016)," 2016.

[30] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong,

R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in *International conference on machine learning*, 2016, pp. 1378–1387.

[31] S. P. Lende and M. Raghuwanshi, "Question answering system on education acts using nlp techniques," in *2016 world conference on futuristic trends in research and innovation for social welfare (Startup Conclave)*. IEEE, 2016, pp. 1–6.

[32] Y. Liu and J. Zhang, "Deep learning in machine translation," in *Deep Learning in Natural Language Processing*. Springer, 2018, pp. 147–183.

[33] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *arXiv preprint arXiv:1908.09635*, 2019.

[34] K. Merchant and Y. Pande, "Nlp based latent semantic analysis for legal text summarization," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2018, pp. 1803–1807.

[35] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[36] A. Mukerjee, R. Biswas, K. Deb, and A. P. Mathur, "Multi–objective evolutionary algorithms for the risk–return trade–off in bank loan management," *International Transactions in operational research*, vol. 9, no. 5, pp. 583–597, 2002.

[37] C. O'neil, *Weapons of math destruction: How big data increases inequality and threatens democracy*. Broadway Books, 2016.

[38] E. Ortiz-Ospina, "The rise of social media," *Our World in Data*, vol. 18, 2019.

[39] J. H. Park, J. Shin, and P. Fung, "Reducing gender bias in abusive language detection," *arXiv preprint arXiv:1808.07231*, 2018.

[40] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[41] M. Popel, M. Tomkova, J. Tomek, Ł. Kaiser, J. Uszkoreit, O. Bojar, and Z. Žabokrtskỳ, "Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals," *Nature communications*, vol. 11, no. 1, pp. 1–15, 2020.

[42] M. O. Prates, P. H. Avelar, and L. C. Lamb, "Assessing gender bias in machine translation: a case study with google translate," *Neural Computing and Applications*, pp. 1–19, 2019.

[43] C. Raffel and D. P. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," *arXiv preprint arXiv:1512.08756*, 2015.

[44] E. Reichert, H. Qiu, and J. Bayrooti, "Reading between the demographic lines: Resolving sources of bias in toxicity classifiers," *arXiv preprint arXiv:2006.16402*, 2020.

[45] J. Risch and R. Krestel, "Toxic comment detection in online discussions," in *Deep Learning-Based Approaches for Sentiment Analysis*. Springer, 2020, pp. 85–109.

[46] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[47] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," *arXiv preprint arXiv:1509.00685*, 2015.

[48] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[49] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[50] W. Song, M. Feng, N. Gu, and L. Wenyin, "Question similarity calculation for faq answering," in *Third International Conference on Semantics, Knowledge and Grid (SKG 2007)*. IEEE, 2007, pp. 298–301.

[51] C. Sweeney and M. Najafian, "A transparent framework for evaluating unintended demographic bias in word embeddings," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1662–1667.

[52] O. Tas and F. Kiyani, "A survey automatic text summarization," *PressAcademia Procedia*, vol. 5, no. 1, pp. 205–213, 2007.

[53] A. Vaidya, F. Mai, and Y. Ning, "Empirical analysis of multi-task learning for reducing identity bias in toxic comment detection," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 14, 2020, pp. 683–693.

[54] E. Vanmassenhove, C. Hardmeier, and A. Way, "Getting gender right in neural machine translation," *arXiv preprint arXiv:1909.05088*, 2019.

[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[56] M. Wiegand, J. Ruppenhofer, and T. Kleinbauer, "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)*, 2019, pp. 602–608.

[57] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, 2014, pp. 1–10.

[58] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[59] T. Yang, R. Yao, Q. Yin, Q. Tian, and O. Wu, "Mitigating sentimental bias via a polar attention mechanism," *International Journal of Data Science and Analytics*, pp. 1–10, 2020.

[60] H. Yeo, "A machine learning based natural language question and answering

system for healthcare data search using complex queries," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 2467–2474.

[61] M. Yousefi-Azar and L. Hamey, "Text summarization using unsupervised deep learning," *Expert Systems with Applications*, vol. 68, pp. 93–105, 2017.

[62] B. H. Zhang, B. Lemoine, and M. Mitchell, "Mitigating unwanted biases with adversarial learning," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 335–340.

[63] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[64] Y. Zhang and Q. Yang, "An overview of multi-task learning," *National Science Review*, vol. 5, no. 1, pp. 30–43, 2018.

[65] S. Zimmerman, U. Kruschwitz, and C. Fox, "Improving hate speech detection with deep learning ensembles," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden