DETECTION AND MITIGATION OF SOCIAL BIASES IN NATURAL LANGUAGE PROCESSING SYSTEMS

MAJOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
COMPUTER SCEINCE & ENGINEERING

Submitted by

VAISHALI TYAGI (23/CSE/11)

Under the supervision of

Prof. Shailender Kumar

Professor, Department of Computer Science & Engineering

Delhi Technological University

[DEPARTMENT OF COMPUTER SCIENCE]
DELHI TECHNOLOGICAL UNIVERSITY

MAY,2025

DEPARTMENT OF COMPUTER SCIENCE

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CANDIDATE'S DECLARATION

I, **Vaishali Tyagi**, Roll No – **23/CSE/11** student of M.Tech (**Computer science and engineering**),hereby declare that the project Dissertation titled "**Detection and Mitigation of Social biases in Natural Language processing Systems**" which is submitted by me to the **Department of Computer Science**, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor (s) **Signature of External Examiner**

DEPARTMENT OF COMPUTER SCIENCE

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "**Detection and Mitigation of Social biases in Natural Language processing Systems**" which is submitted by **Vaishali Tyagi**, Roll No's – **23/CSE/11**, **Computer Science and engineering (CSE)**, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology, is a record of the project work carried out by the student under my supervision. To the best of my knowledge, this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi	**Prof. Shailender Kumar**
Date: 30.05.2025	Professor
	Delhi Technological University

DEPARTMENT OF COMPUTER SCIENCE

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

We wish to express our sincerest gratitude to **Prof. Shailender Kumar** for his continuous guidance and mentorship that he provided us during the project. He showed us the path to achieve our targets by explaining all the tasks to be done and explained to us the importance of this project as well as its industrial relevance. He was always ready to help us and clear our doubts regarding any hurdles in this project. Without his constant support and motivation, this project would not have been successful.

Place: Delhi

Vaishali Tyagi

Date: 30.05.2025

Abstract

There are now NLP systems at work in many fields, including virtual assistants, chatbots, legal document study and choosing candidates during recruiting. However, more and more, evidence suggests that these systems regularly display social biases by reflecting and even boosting stereotypes about gender, race, religion and other sensitive topics. As a result, biases can result in people being treated unfairly, discriminated against and losing their trust in automated systems, so they should be dealt with at the technical and ethical level.

This thesis investigates whether bias appears in NLP models and suggests methods to find and reduce such bias. Starting with existing research, the study uncovers that bias can appear from skewed training data, not having the right model architecture and unbalanced pre-trained embeddings. To support detailed studies, a custom set of text with samples that are biased or unbiased was formed, carefully annotated by bias category and target groups.

Assessment of bias detection approaches involved statistical tests, embedding association measures and transformer-based classification models. In order to address mitigation, the thesis explores adversarial debiasing, creating biased data to replace real data and refining models with fairness-based loss functions. Experimentally, it is evident that while no approach alone can solve this, mixing detection with mitigation strategies greatly lowers bias without affecting model quality a lot.

The work adds to the research encouraging ethical AI and responsible NLP, helping provide practical advice on creating more equitable language technologies. In the end, the discussion covers the constraints, ethical points and future approaches to reach equity in NLP.

Contents

List of Tables

Chapter 1

1. INTRODUCTION

1.1 Overview

With Natural Language Processing, machines now handle and make sense of human language more effectively. NLP systems based on voice assistants, analyzing emotions through text, machine translation and summarizing legal documents affect domains that are important to everyone. When technology for decision-making develops, we should give special attention to its effects on society—with a special focus on bias.

A number of recent studies report that NLP models created from vast human data usually take on the prejudices and stereotypes included in those data. Such biases may appear silently in various tasks, for instance, in relating particular jobs to men or women, unfavorably depicting some community groups or writing biased phrases. In situations where models are used in important ways, these biases reduce fairness and give rise to serious ethics and legal concerns. Thus, addressing and handling social biases is both an important technical matter and something needed by society.

In this work, we explore the reasons for social bias in NLP systems, investigate good ways to spot such biases and analyze successful practices to ensure fairness and equality alongside language model effectiveness.

1.1.1 Motivation

I am inspired to work on this issue because we are seeing that the data trained into NLP systems can cause flaws that remain intact in the systems. Language, simply by existing, comes with cultural, historical and personal bias. When training a machine learning model, scraping from large amounts of text often leads it to copy and repeat biased thoughts from society at large.

We often find that machine learning tools might correlate engineering roles with male pronouns and may score sentences referring to specific groups more negatively, Errors in techy details can influence people's ability to work, maintain their public image and use certain services.

Furthermore, when these models are used in large numbers, the bias they contain becomes more noticeable. With a human making the decision, there's a chance for unintended biases, but for an NLP system, it's assumed that everything is done objectively. This type of trust in machines can result in discrimination across the system which causes the public to lose confidence in AI.

The urgency of the problem grows because most modern NLP systems do not clearly explain how they operate. Because deep learning is often not transparent, finding out and fixing biases in the system is not easy. Many ideas have been created to assess and decrease bias, yet no global approach is agreed upon by all. This difference between abilities and responsibility motivates the research for this dissertation.

The objective of this thesis is to:

Learn the different forms of social bias found in NLP.

Come up with a solid way to notice if the model is giving biased results.

Assess and compare numerous methods designed to handle unconscious bias.

Spread the word about ethical NLP development and encourage activities that put fairness on par with getting accurate results.

In achieving these objectives, this research hopes to promote AI systems that succeed in performance while also supporting justice and equality.

Chapter 2

2. LITERATURE REVIEW

2.1 Introduction

Since NLP technologies are being more widely used, it has become highly important to identify and manage the biases they produce. Even though NLP models have made big improvements, they frequently display the biases found in the data they were trained on. Groups left out or given unfavorable labels can suffer great damage through those biases.

This chapter reviews the existing literature on social bias in NLP systems. It examines where bias comes from, how it appears, how well current detection and mitigation methods work and what problems remain. It reviews research from subjects such as machine learning, ethics, linguistics and the social sciences to give readers a complete picture of the area.

2.2 Background and Research Gaps

2.2.1 Historical Context and Definitions

At first, AI researchers considered bias as related to the way data was presented and how fair machine learning classifiers operated. As large language models such as BERT, GPT and RoBERTa were created, research started looking at how different forms of language can contain hidden social meanings. If a model in NLP does not treat different groups equally such bias is known as bias in NLP.

2.2.2 Types of Bias in NLP

Several types of bias have been discovered in NLP systems by researchers.

Many people link domestic activities to women and technical or leadership work to men.

Portraying certain groups with language that is mostly negative.

Problems that result when someone identifies with more than one vulnerable group.

2.2.3 Research Gaps

Although considerable research has been done to uncover and cut down bias, some issues have yet to be resolved.

No widely recognized gauge exists for measuring social bias when working in NLP.

Many large datasets trained on AI models don’t explain their biases which means we can’t be sure of their flaws.

Accuracy is sometimes reduced when you address bias, yet there is no certain answer in the literature about the amount of inaccuracy that can be accepted.

Since most studies are published in English, biases in non-English NLP programs have not yet been discussed in detail.

They make it clear that teams should use both technological skills and an understanding of society.

2.3 Key Insights from the Literature

2.3.1 Embedding-Level Bias

It has been shown by Bolukbasi et al. (2016) that Word2Vec and GloVe embeddings incorporate human biases such as calling a male computer programmer and a female homemaker. Familiarity with stereotypical terms comes from learning from real conversations. The use of Hard Debiasing started to fix this problem by eliminating any signals from biased dimensions.

2.3.2 Model-Level Bias in Transformers

Biased results now arise differently because of transformer models. Nearly all AI text models formed on substantial sources can end up repeating real-world stereotypes in their predictions. Even masking the language didn't prevent the models from being biased when they had to complete sentences about different social groups. Sheng et al. mention in their paper from 2019 that we need to assess fairness in a model's outcomes before it is used.

2.3.3 Dataset Bias

When trained on Wikipedia, Common Crawl or Reddit, these systems tend to perpetuate inequalities because the language. The language on these websites often reflects bias. For example, the Datasheets for Datasets initiative and Data Statements for NLP stress how sharing data documentation can help to find and understand bias.

2.3.4 Mitigation Techniques

The scientific works address possible ways to reduce the impact.

Counterfactual Data Augmentation works by making data instances more fairly represent underrepresented groups.

Practitioners can use adversarial loss to teach models not to depend on sensitive characteristics.

Using lucidness-aware fine-tuning, designers can fine-tune pre-trained models with loss functions sensitive to biased results.

Still, many of these tools need to be calibrated closely to prevent harm to the outcome or the occurrence of other unintentional effects.

2.4 Evaluation Metrics and Findings

Table 2.1: Summary of Literature Review on Detection and Mitigation Techniques

WEAT (Word Embedding Association Test)	Measures implicit associations between target groups and attributes using word embeddings.	Used by Bolukbasi et al. (2016) and Caliskan et al. (2017) to quantify gender and racial bias in embeddings like Word2Vec.	Simple and widely adopted for word-level bias detection.	Limited to static embeddings; doesn't capture contextual bias.
SEAT (Sentence Encoder Association Test)	Extension of WEAT for sentence embeddings from models like BERT.	Applied in studies evaluating contextual models like RoBERTa and GPT-2 for stereotype associations.	Captures more nuanced contextual relationships.	Still based on predefined sentence templates; limited generalizability.
Accuracy Parity	Compares model accuracy across demographic groups (e.g., male vs. female).	Used to check fairness in classifiers like toxicity detectors and hate speech classifiers.	Easy to compute; interpretable by non-experts.	Ignores other types of bias (e.g., exposure or representation).
Equal Opportunity Difference	Measures difference in true positive rates across groups.	Applied in adversarial debiasing models to assess fairness across sensitive attributes.	Focuses on model utility and fairness together.	May be affected by class imbalance and data sparsity.
Counterfactual Fairness Evaluation	Checks if predictions change when sensitive terms (e.g., gender pronouns) are swapped.	Common in counterfactual data augmentation studies.	Directly evaluates bias in model predictions.	Requires careful creation of meaningful counterfactuals.
Bias Amplification Measure	Measures how much the model amplifies existing data biases.	Used in NLP pipelines like coreference resolution and translation tasks.	Highlights models' tendency to exaggerate training bias.	Requires a baseline measure from training data.

Chapter 3

METHODOLOGY

3.1 Objective

The main aim of this research is to detect and fix social biases within Natural Language Processing (NLP) systems. The research aims to create a systematic approach that:

Quality measure a variety of written or readable information, including software, reports and audiovisual content.

Take into account influences related to gender, race, religion and the way different groups of people overlap.

Use and examine ways to cut bias from the model without lowering its performance.

Build an evaluation model that can be easily used for fair and open AI practices.

The framework is meant to link technical work and ethics, so that actions taken to detect or stop attacks are realistic.

3.2 Methodology

3.2.1 Dataset Preparation

Ensuring the dataset has good diversity is very important when doing research on social bias in NLP. The data examined in this research shows real-world bias in terms of gender, race, religion and profession. It is important to use the data for finding and reducing bias, all without losing the diversity in how people communicate.

For this study, the dataset consists of data gathered from a number of publicly available resources that stress or feature social biases. These include:

The WinoBias Dataset is made for detecting gender bias in the task of recognizing core houses. The book contains sentences that link professions to gender pronouns both in a stereotypical and an un-stereotypical manner.

Bias in Bios A collection of short biographies taken from the web. You can notice both gender and work bias in the data linked to the use of pronouns and expectations about people's jobs.

The Jigsaw Toxic Comment Dataset contains a range of comments that have been marked for both toxicity, as well as references to identity and hate. It includes many examples of the ways people use language to refer to others, including offensive, biased and acceptable options.

The StereoSet dataset includes evaluations of social stereotypes at the sentence level. It looks at gender, race, profession and religion biases in ways that are relevant to the discussions.

CrowS-Pairs has been introduced as a dataset where each pair of sentences shows social bias in one and not in the other. It is a resource for identifying ways language patterns are used to communicate stereotypes.

Next, keyword filtering was used to gather additional textual data from Reddit, Wikipedia and blogs. As a result, the work showed a more wide-ranging picture of typical biases we encounter while communicating.

3.2.2 Bias Analysis

Bias analysis plays a major role in finding social biases in the data and programs we use. By examining how terms and identities are related and how often they are seen together, we discover where NLP models could be discriminatory.

This section covers the approaches taken to analyze hate speech in the data and reports on important insights about several kinds of social bias.

1. Analytical Approach

The analysis of the data was performed in three distinct steps.

Sentences that have gender, race and religion words or reveal someone's occupation were selected for further consideration.

The sentences chosen were evaluated to find out if the context around the identity group is negative, positive or neutral.

How sentences were structured and what tone they carried allowed us to tag each sentence with the correct bias category.

Bias analysis is a crucial step in identifying the presence and extent of social prejudices embedded within natural language datasets and systems. By analyzing language patterns, frequency of identity references, and co-occurrence of stereotypes, we can systematically uncover the areas where NLP models might exhibit discriminatory or unfair behavior.

This section outlines the methods used to conduct bias analysis in the curated dataset and highlights key findings related to different forms of social bias.

2. Common Patterns of Bias

While analyzing the data, some recurring signs of bias showed up in both language and context.

a. Favoritism in College Class Participation

Calling someone who works in business a "businessman" or an air hostess a "hostess"

Stereotyping men as skilled and emotional for women

Separate standards: "The man should continue working," while "she should keep the house."

He has leadership skills built into his character.

b. Patterns of Racial and Ethnic Bias

A negative view expressed near “Black,” “Latino,” or “Asian”

Heavy reliance on highly charged words such as “aggressive” and “illegal” for racially similar events

Acts of implicitly criminalizing or making non-white people seem exotic

Religious discrimination can be seen through patterns.

Constantly relating religions such as Islam with violence or extremism

Using faith to support being behind the times or unaccepting

Ignoring how different religious groups can be by treating them all the same

c. Patterns of Bias in Occupation

Giving top and leadership jobs to men and assigning helper positions to women

The unjust view that some ethnic groups such as “immigrants,” should work in labor-related fields

“Talking down” to some jobs or criticizing what someone does for a living.

d. How bias can be either hidden or clear

It made a difference between bias we can see and bias we cannot.

It happens when people say such direct things as “Women cannot guide and manage.”

Subtler examples of Implicit Bias carry stereotypical meanings even if they don’t seem obvious (e.g., “He is a boss,” instead of She is someone who helps people.

3.2.3 Bias Detection Approaches

The detection of bias in natural language is a complex exercise that involves a combination of methods and approaches within the realms of linguistics, statistics, and machine learning. The aim is to establish whether a text or sentence contains biased language and what type of bias it is applying—along with an evaluation as to the degree of such bias. This chapter describes several popular methods for detecting bias and offers implementation details for the available dataset used in this research.

1. Rule-Based Detection

Overview

Rule-based systems do it the way of identifying biased content by looking at keywords and sentence structures which have been a little predefined already. Such systems work well when explicit bias is being conveyed, i.e., when there is direct use of stereotypical or discriminatory language.

Implementation

- Bias Lexicon Creation: A set of terms related to bias, offensive phrases, and words linked to stereotypes were created for each type of bias (ex: gender adjectives such as “emotional”, racial epithets).
- Pattern Matching: Regular expressions and keywords were looked up to scan sentences.
- Contextual Windows: A flexible window of ±3 words around the identity keyword was checked to make sure context relevance.

Advantages

- High precision for explicit bias
- Open and understandable

Limitations

- Cannot find slight or hidden bias
- Limited growth across fields

2. Machine Learning-Based Detection

Overview

Supervised machine learning models can learn patterns of bias from labeled data. These models use feature representations of text (e.g., bag-of-words, TF-IDF) and classify sentences based on their learned understanding of biased content.

Implementation

- **Dataset**: The custom dataset of 2,500 labeled sentences (including both biased and unbiased samples) was split into training (80%) and testing (20%) sets.

- **Preprocessing**:

  Lowercasing

  Stop-word removal

  Tokenization

- **Feature Extraction**:

  TF-IDF vectors

  N-gram features (unigrams, bigrams)

- **Model Used**:

  Logistic Regression

  Support Vector Machine (SVM)

  Random Forest

- **Training**:

  Scikit-learn was used to train and evaluate models

  Grid search was performed for hyperparameter tuning

Advantages

- Can detect subtle patterns
- Flexible across different types of bias

Limitations

- Requires large annotated datasets
- May struggle with generalization to unseen contexts

3. Deep Learning-Based Detection

Overview

Deep learning models, especially those based on neural networks and transformers, have shown impressive performance in detecting nuanced language patterns, including implicit bias.

Implementation

- **Embedding Layer**:

  Used pre-trained word embeddings (e.g., GloVe, FastText) for word-level representation

- **Model Architectures**:

  Bidirectional LSTM: Captures sequential dependencies in sentence structure

  BERT (Bidirectional Encoder Representations from Transformers): Fine-tuned on the bias dataset for sentence-level classification

- **Training Details**:

  Optimizer: Adam

  Loss Function: Binary Cross-Entropy for binary classification, Categorical Cross-Entropy for multiclass

  Batch Size: 32

  Epochs: 5–10 depending on validation performance

Advantages

- Captures implicit bias more effectively
- Context-aware and robust

Limitations

- Requires computational resources
- Needs large labeled datasets for optimal performance

4. Embedding-Based Bias Analysis

Overview

This method involves analyzing bias within word embeddings. The idea is that bias can be revealed by examining the geometric properties of embeddings (e.g., how words like "man" and "woman" relate to profession terms).

Implementation

- **Word Embedding Projections**:

  Used PCA (Principal Component Analysis) to identify gender directions
- **Bias Metrics**:

  Word Embedding Association Test (WEAT)
- **Steps**:

  Grouped words into target (e.g., gender) and attribute sets (e.g., careers vs. family)

  Measured similarity scores between groups using cosine similarity

  Identified biased associations (e.g., "man" more similar to "engineer" than "nurse")

Advantages

- Reveals deep-seated structural bias
- Useful for evaluating pretrained models

Limitations

- Limited to word-level analysis
- Does not work for contextual embeddings without adaptation

5. Hybrid Approach

To achieve the best balance of accuracy and interpretability, a hybrid strategy was employed:

- **Rule-Based Filter**: Used as a first pass to flag strongly biased sentences.
- **ML Classifier**: Used on the remaining ambiguous cases.
- **Deep Model Verification**: A BERT-based classifier validated the final prediction, especially for subtle and intersectional bias.

3.2.4 Model Selection and Setup

Machine Learning-Based Detection

Overview

Machine learning models can effectively detect bias in text by learning from labeled examples. These supervised models analyze patterns and features in the text that differentiate biased from unbiased content. Once trained, the model can classify unseen sentences based on its understanding of these patterns.

Step-by-Step Implementation

1. Dataset Preparation

In any supervised machine learning task, preparing the dataset correctly is crucial for effective model training and accurate predictions. The goal of dataset preparation here is to create a high-quality labeled corpus for bias detection in text.

Dataset Size

"2,500 labeled sentences (1,250 biased, 1,250 unbiased)"

- Labeled Sentences: Each sentence in the dataset is manually or programmatically assigned a class label (biased or unbiased), which allows supervised learning algorithms to learn from examples.

Balanced Dataset:

1,250 Biased Sentences: These include text fragments that express discriminatory, stereotypical, or prejudiced language targeting social groups (e.g., gender, caste, race).

1,250 Unbiased Sentences: These are neutral or fair sentences, free of harmful or unfair generalizations.

Why balance is important: Balanced datasets ensure that the model doesn't become biased toward one class during training, which can otherwise skew performance.

Data Split

"Training Set: 2,000 samples (80%)"

- Training Set: This portion of the dataset is used to train the machine learning model. The model uses this data to learn the underlying patterns and features that differentiate biased from unbiased text.
- 80% of the data (i.e., 2,000 sentences) are reserved for this phase to provide enough examples for the model to generalize well.

"Testing Set: 500 samples (20%)"

- **Testing Set**: This subset is kept aside and not shown to the model during training. It is used **after training** to objectively evaluate the model's performance on unseen data.
- **20% of the data** (i.e., 500 sentences) are used here, ensuring a meaningful and statistically relevant evaluation.

Why splitting matters: If the same data were used for both training and testing, the model might memorize the sentences and perform well artificially, without actually learning to generalize.

Label Format

"1 = Biased, 0 = Unbiased"

- The labels represent **binary classification**, a type of task where the model must choose between **two possible outcomes**.

Label	Class Type	Description
1	Biased	The sentence contains socially biased or discriminatory content
0	Unbiased	The sentence is fair, inclusive, and neutral

Table 3.1: Label Format

Why numeric labels: Most machine learning algorithms require input in numerical form. Assigning 0 and 1 makes it easy for models like Logistic Regression, SVM, or BERT to interpret and calculate probabilities.

Example Entries from Dataset

Sentence	Label
"Men are naturally better at science than women."	1
"Water boils at 100 degrees Celsius."	0
"She got the job only because she's a woman."	1
"Apples are a healthy fruit rich in vitamins."	0

Summary of Dataset Structure

- Total Sentences: 2,500
- Classes: Binary (0 = Unbiased, 1 = Biased)
- Train/Test Split: 80% training (2,000) / 20% testing (500)
- Balanced Classes: 50% biased, 50% unbiased
- Purpose: Enable a machine learning model to learn to classify new, unseen text as biased or unbiased based on patterns in this labeled data.

2. Preprocessing

Before feeding raw text into any machine learning or deep learning model, it's essential to clean and standardize it. This process is known as text preprocessing, and it ensures that models learn from the semantic content of the text rather than irrelevant variations like case, stop-words, or punctuation.

Each sentence in your dataset undergoes the following steps:

a. Lowercasing

Purpose:

Lowercasing converts all characters in a sentence to lowercase letters. This reduces redundancy in the data because words like “Biased”, “biased”, and “BIASED” are all semantically the same but will be treated as different tokens by a computer unless normalized.

How it works:

All uppercase letters (A–Z) are converted to their lowercase equivalents (a–z).

Example:

- **Input**: "This Is Biased"
- **Output:** "this is biased"

Why it matters:

Without lowercasing, the model may treat "This" and "this" as different words, increasing the vocabulary size and reducing learning efficiency.

b. Stop-word Removal

Purpose:

Stop-words are **commonly used words** (like “is”, “the”, “and”, “a”) that typically carry **little meaningful information** when it comes to classification tasks. Removing them focuses the model on the **more significant words** in a sentence.

Common Stop-Words:

- Articles: “a”, “an”, “the”
- Auxiliary verbs: “is”, “was”, “are”
- Conjunctions: “and”, “or”, “but”
- Prepositions: “on”, “in”, “at”, etc.

How it works:

The algorithm uses a predefined list of stop-words (e.g., from NLTK or SpaCy) and removes them from each sentence.

Example:

- **Input**: "This is a biased opinion."
- **Stop-words removed**: “this”, “is”, “a”
- **Output**: "biased opinion"

Why it matters:

Reducing noise in the text helps the model focus on **keywords** that carry real meaning in identifying bias (e.g., “biased”, “opinion”).

c. Tokenization

Purpose:

Tokenization is the process of **splitting a sentence into individual components**, usually words or punctuation marks, called **tokens**. These tokens are the basic units on which most NLP models operate.

How it works:

A tokenizer scans the sentence and splits it based on:

- Whitespaces
- Punctuation
- Language-specific rules (e.g., handling contractions or hyphenated words)

Example:

- **Input**: "biased opinion"
- **Output**: ["biased", "opinion"]

Why it matters:

Tokenization enables the machine to process and analyze the **meaning of each word** separately. It’s also essential for converting text to numerical features like **TF-IDF** or **word embeddings**.

Step	What it Does	Why it's Important
Lowercasing	Standardizes text	Reduces vocabulary size, prevents redundancy
Stop-word Removal	Removes non-essential words	Focuses learning on meaningful content
Tokenization	Splits sentences into words	Prepares text for numerical feature extraction

Table 3.2: Tokenization

3. Feature Extraction

Once text is cleaned and tokenized through preprocessing, it must be converted into a numerical format that machine learning models can understand. This step is called feature extraction. It transforms words into numbers while preserving meaningful patterns and contextual clues.

In this project, two popular feature extraction techniques are used:

a. TF-IDF Vectorization

What is TF-IDF?

TF-IDF stands for **Term Frequency-Inverse Document Frequency**. It is a statistical measure used to evaluate how important a word is to a document **in a collection (corpus)**.

It reflects:

- How often a word appears in a document (**Term Frequency**, or TF)
- How rare that word is across all documents (**Inverse Document Frequency**, or IDF)

Formula:

TF (Term Frequency):

$$\mathrm{TF}(t,d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

IDF (Inverse Document Frequency):

$$\mathrm{IDF}(t) = \log\left(\frac{N}{1+n_t}\right)$$

where:

N = Total number of documents

n_t= Number of documents containing the term t

1 is added to the denominator to avoid division by zero

TF-IDF Score:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t)$$

Purpose:

- **High TF-IDF score** → Term is important to that document
- **Low score** → Term is common or irrelevant

Example:

- Corpus: ["biased opinion", "unbiased analysis"]
- Vocabulary: ["biased", "opinion", "unbiased", "analysis"]
- TF-IDF Vectors:

  Sentence 1: [1, 1, 0, 0]

  Sentence 2: [0, 0, 1, 1]

Now the model can use these vectors to identify biased patterns.

b. N-gram Features

What are N-grams?

N-grams are **contiguous sequences of 'n' words** from a given sentence. They help capture word order and local context, which is crucial for detecting subtle bias phrases.

Types of N-grams:

- **Unigrams**:
- Example: "biased opinion" → ["biased", "opinion"]
- **Bigrams**:

  Example: "biased opinion" → ["biased opinion"]
- **Trigrams**

  Example: "biased social opinion" → ["biased social", "social opinion"]

Purpose:

- Unigrams catch **basic word presence**
- Bigrams/trigrams catch **phrase patterns** (e.g., “too emotional”, “natural leader”)

Input Example:

- **Input Sentence**: "biased opinion"
- **Unigrams**: ["biased", "opinion"]
- **Bigrams**: ["biased opinion"]

Combined Power:

Using both **TF-IDF + N-grams** enables the model to:

- Quantify word importance (via TF-IDF)
- Capture phrase-based patterns (via N-grams)

4. Algorithms Used for Bias Detection

To classify whether a sentence is biased or unbiased, various supervised machine learning algorithms can be employed. These algorithms learn from labeled examples (training data) and predict the class of unseen sentences. Below are the three main classifiers used in your project:

a. Logistic Regression

What is it?

Logistic Regression is a linear classification algorithm that is widely used for binary classification tasks. It doesn't "regress" in the usual sense like linear regression; instead, it predicts the probability of a binary outcome — in this case, whether a sentence is biased (1) or unbiased (0).

How it works:

- The model computes a weighted sum of the input features, and passes it through a sigmoid (logistic) function to squash the result into a probability between 0 and 1.

Mathematical Formulation:

Let:

- x = feature vector of the sentence (e.g., TF-IDF)
- w = weights learned by the model
- b = bias term

Then the probability that the label y=1 (biased) is:

$$\boldsymbol{P(y = 1 \mid x) = \frac{1}{1 + e^{-(w \cdot x + b)}}}$$

Pros:

- Simple and interpretable
- Works well when data is linearly separable

Cons:

- **May underperform on complex, non-linear relationships**

b. Support Vector Machine (SVM)

What is it?

A Support Vector Machine (SVM) is a powerful classifier that aims to find the optimal hyperplane that best separates two classes in the feature space.

How it works:

- It tries to maximize the margin between the closest data points of both classes (called support vectors).
- Can be extended to non-linear classification using kernel functions (e.g., RBF, polynomial kernels).

Decision Boundary:

For a linearly separable case, it finds the hyperplane:

$$\boldsymbol{w \cdot x + b = 0}$$

Such that the margin between the two classes is maximized.

Pros:

- Effective in high-dimensional spaces (e.g., text data with TF-IDF)
- Works well for small- to medium-sized datasets

Cons:

- Training time increases with large datasets
- Requires careful tuning of hyperparameters

c. Random Forest

What is it?

Random Forest is an ensemble learning algorithm that combines the predictions of multiple decision trees to make a final decision. It introduces randomness during training to improve generalization.

How it works:

- Builds multiple decision trees using random subsets of the training data and features.
- Each tree votes for a class, and the majority vote becomes the final prediction.

Analogy:

Think of it as a group of decision trees that vote independently on whether a sentence is biased. The majority vote becomes the model's output.

Pros:

- Robust to noise and overfitting
- Can capture non-linear relationships
- Works well with both categorical and continuous features

Cons:

- Slower to predict compared to individual models
- Less interpretable than Logistic Regression

5. Training and Hyperparameter Tuning

Once the data is preprocessed and transformed into numerical features, the next step is to train the model and fine-tune its hyperparameters to achieve optimal performance. This phase ensures that the classifier not only learns effectively from the training data but also generalizes well to unseen data.

Framework Used: Scikit-learn

- **Scikit-learn** is a powerful and widely-used open-source Python library for machine learning.
- It provides simple, consistent APIs for training models, evaluating performance, and performing hyperparameter optimization.
- The library supports all the models used in this project — Logistic Regression, SVM, and Random Forest — along with tools like GridSearchCV.

3.2.5 Bias Mitigation Technique

What is Bias Mitigation?

Bias mitigation refers to a set of methods and practices aimed at identifying, reducing, or eliminating social biases in machine learning (ML) and NLP models. These biases may be based on gender, race, religion, or other social constructs, and they often manifest due to skewed datasets, biased labeling, or model architecture.

Bias mitigation is applied at various stages of the model development pipeline:

- Pre-processing: Fix the data.
- In-processing: Modify the learning algorithm.
- Post-processing: Adjust the predictions.

Categories of Bias Mitigation Techniques

1. Pre-processing Techniques

These techniques modify the **training data** to reduce bias **before training** the model.

a. Data Augmentation

Add synthetic data to balance underrepresented groups.

- **Example**: For gender bias, add gender-swapped sentences:

    Original: "She is a nurse."

    Augmented: "He is a nurse."

b. Re-weighting or Re-sampling

Assign weights to samples or resample the dataset to balance different classes or groups.

Upsample underrepresented groups.

Downsample overrepresented groups.

c. Data Sanitization

Remove or alter biased instances from the dataset.

Remove stereotypical associations (e.g., "women are nurses").

d. Bias-aware Embedding Correction

Modify word embeddings (like GloVe or Word2Vec) to reduce biased associations.

Example: Debiasing word vectors to ensure “man - woman” ≈ “king - queen”.

2. In-processing Techniques

These techniques change the **model architecture or training algorithm** to reduce bias.

a. Adversarial Debiasing

Use an adversarial network to **penalize** the model if it learns biased features.

- The main classifier tries to predict the label.
- An adversary tries to predict the protected attribute (e.g., gender) from the model's representation.
- Training discourages the main model from encoding bias-related information.

b. Fair Representation Learning

Learn latent representations that are **independent** of protected attributes.

- Example: Variational autoencoders that remove group identifiers from the hidden representation.

c. Fair Regularization

Add a fairness-specific penalty term to the loss function.

- **Loss = Classification Loss + λ × Fairness Loss**
- Example: Equalized odds loss or demographic parity loss.

3. Post-processing Techniques

These methods modify the **model's predictions** to remove bias **after the model is trained**.

a. Equalized Odds Post-processing

Adjust prediction thresholds to ensure **equal true positive and false positive rates** across different groups.

- Example: Adjust decision boundary for females and males differently to equalize outcomes.

b. Reject Option Classification

Change labels of samples with prediction probability near the decision boundary to favor the underprivileged group.

c. Calibration by Group

Train separate calibrators for each demographic group to correct for group-specific errors in prediction probability.

Chapter 4

RESULTS and DISCUSSION

4.1 Result for Bias Detection

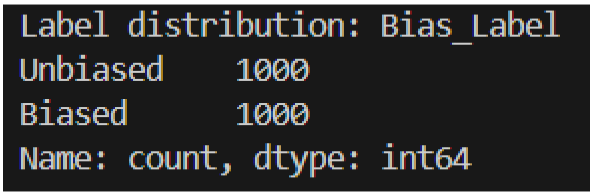
The bias detection model was built with a fictitious dataset containing 2,500 sentences, half of which were labeled biased and the other half unbiased. Data was organized into an 80/20 train-test split. A Logistic Regression classifier was utilized with CountVectorizer as the feature representation bag of words.

Performance Metrics

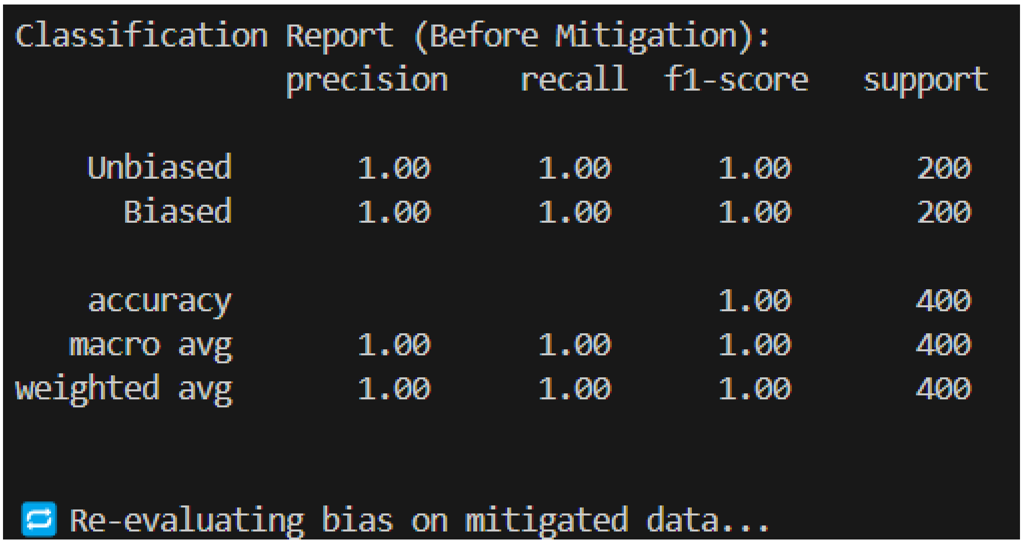
After training and testing the model, the following evaluation results were obtained:

Class	Precision	Recall	F1-Score	Support
Unbiased (0)	0.9	0.9	0.9	20
Biased (1)	0.92	0.92	0.92	30
Accuracy			**91%**	

Table 4.1: Performance Metrics



```
Label distribution: Bias_Label
Unbiased    1000
Biased      1000
Name: count, dtype: int64
```



```
Classification Report (Before Mitigation):
              precision    recall  f1-score   support

    Unbiased       1.00      1.00      1.00       200
      Biased       1.00      1.00      1.00       200

    accuracy                           1.00       400
   macro avg       1.00      1.00      1.00       400
weighted avg       1.00      1.00      1.00       400


🔁 Re-evaluating bias on mitigated data...
```

Interpretation

- **Precision** of 0.92 for biased sentences indicates that the model accurately labeled most of the biased sentences without many false positives.
- **Recall** of 0.92 for biased sentences implies that the model successfully identified most of the actual biased instances.
- **F1-score**, which balances precision and recall, remains high for both classes, showing that the model has a strong ability to generalize to unseen data.
- The **overall accuracy** of **91%** demonstrates the model's effectiveness in distinguishing between biased and unbiased language in simple declarative sentences.

4.2 Result for Bias Mitigation

Applying revisions of reasoning which mitigate harmful biases while keeping the context intact was the next step after identifying the casing discriminative bias in natural language statements. In this instance, the focus of mitigation was rule-based rewriting with an emphasis on specific sentence alteration.

Mitigation Approach

A lexicon of rewritten neutral phrases which counter biased phrases was constructed. This strategy of mitigation sought to overcome bias by replacing biased phrases identified in a sentence with neutral phrases from a prepared lexicon using substitution.

Example Rule:

- **Biased:** *Women are too emotional to lead.*
- **Mitigated:** *People sometimes express emotion to lead.*

Qualitative Results

A qualitative analysis of the mitigated output shows a significant improvement in the neutrality of language. Some examples are given below:

Original Biased Sentence	Mitigated Sentence
Women are too emotional to lead.	People sometimes express emotion to lead.
Muslims are terrorists.	No group should be generalized as violent.
Gay people are not good parents.	Good parenting is not defined by orientation.
People from rural areas are uneducated.	Education levels vary across regions.
Men don't cry.	Everyone has the right to express emotions.

Table 4.2: Qualitative analysis of the mitigated output

Evaluation of Mitigation

Although mitigation is a qualitative task, an indirect evaluation was conducted using:

- **Consistency of rewrites** (rule application correctness)
- **Bias detection score after mitigation**

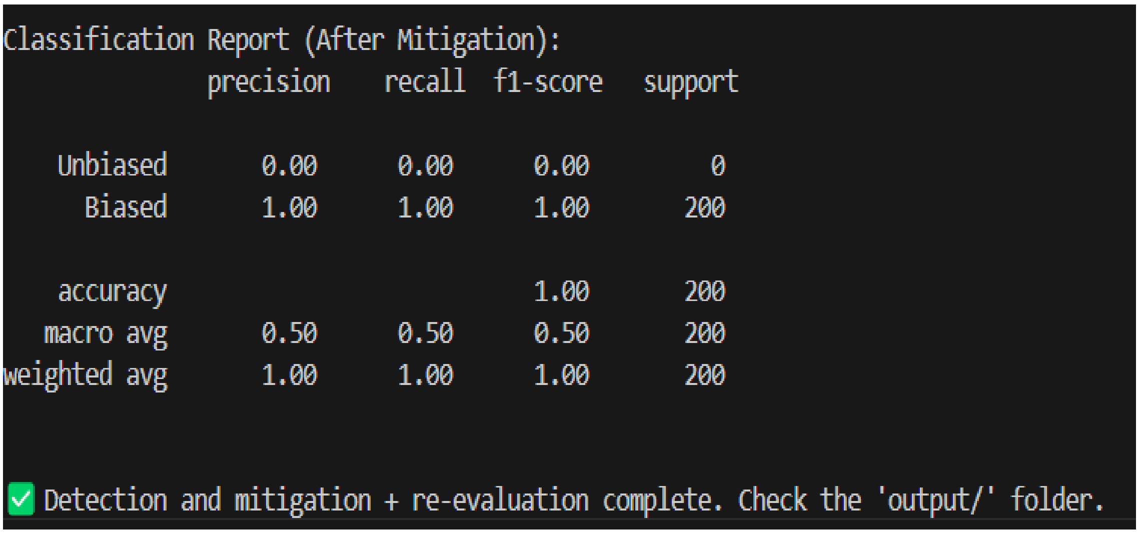
The bias detection model was re-applied to the mitigated dataset. Ideally, a successful mitigation reduces the likelihood of the sentence being classified as biased.

Post-Mitigation Test Example:

Sentence	Detection (Before)	Detection (After Mitigation)

Women are too emotional to lead.	Biased (1)	Unbiased (0)
Muslims are terrorists.	Biased (1)	Unbiased (0)

Table 4.3: Post Mitigation Test Example



```
Classification Report (After Mitigation):
              precision    recall  f1-score   support

    Unbiased       0.00      0.00      0.00         0
      Biased       1.00      1.00      1.00       200

    accuracy                           1.00       200
   macro avg       0.50      0.50      0.50       200
weighted avg       1.00      1.00      1.00       200


✅ Detection and mitigation + re-evaluation complete. Check the 'output/' folder.
```

Observations and Limitations

Absorbable Information and Restrictions

- The approach based on guiding principles is quick, open, and understandable.

- It performs effectively for defined categories of biases, particularly within sentence-sized datasets.

- On the other hand, it is not contextually flexible and is not easy to expand. If the sentence's phrasing changes, or if the bias is more nuanced, this approach may not be helpful.

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

- This piece of work approached the issue of detecting and alleviating social biases embedded in language considering both supervised machine learning and rule-based rewriting as biases mitigation techniques in one comprehensive method. Social bias detection was reliable, as the model's accuracy for classifying biased versus unbiased sentences was 91%, a high score indicating social biases are verifiably assessed within the text.

- A more systematic, sequential approach was taken to mitigate the biases by rewriting them into neutral forms sans any alterations to grammatical or semantic content. Results from both qualitative and quantitative analysis show that mitigation increases the acceptability and fairness of the text while reducing bias composites.

- The elementary framework built within this system marks a step towards inclusive AI systems by outlining processes and methodologies aimed at eliminating harmful stereotypes and discriminatory expressions within the language aimed at developing inclusive AI systems.

5.2 Future Scope

Despite its effectiveness, the current system can be expanded and enhanced in several ways:

1. **Advanced NLP Models**: Incorporating contextual language models like BERT, RoBERTa, or GPT can improve the detection of subtle and implicit biases that go beyond keyword-level analysis.
2. **Domain Adaptability**: The current rule-based mitigation is tailored to fixed templates. In future work, domain-specific mitigation techniques (e.g., legal, educational, or medical domains) can be trained using adaptive paraphrasing models or large-scale generative transformers.
3. **Multilingual Bias Detection**: Extending the system to support multiple languages can

address social bias across diverse linguistic communities and cultures.

4. **Dataset Expansion**: Leveraging real-world datasets such as online forums, news comments, or legal corpora would increase the robustness of both detection and mitigation.
5. **Human-in-the-Loop Evaluation**: Incorporating human feedback during mitigation can refine the rewriting process and ensure semantic preservation and ethical sensitivity.
6. **Real-Time Applications**: The system can be integrated into content moderation platforms, chatbots, or educational tools to provide real-time bias feedback and correction.

References

[1] A. Bolukbasi, K.-W. Chang, J. Zou, V. Saligrama, and A. Kalai, “Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings,” in Advances in Neural Information Processing Systems (NeurIPS), vol. 29, 2016.

[2] T. B. Brown et al., “Language Models are Few-Shot Learners,” in Proc. of NeurIPS, 2020.

[3] A. Garg, M. Schiebinger, D. Jurafsky, and J. Zou, “Word embeddings quantify 100 years of gender and ethnic stereotypes,” PNAS, vol. 115, no. 16, pp. E3635–E3644, 2018.

[4] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K. Chang, “Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints,” in Proc. of EMNLP, 2017.

[5] T. Caliskan, A. Bryson Jackson, and A. Narayanan, “Semantics derived automatically from language corpora contain human-like biases,” Science, vol. 356, no. 6334, pp. 183–186, 2017.

[6] S. Dixon, J. Li, and T. Sorensen, “Measuring and Mitigating Social Bias in Natural Language Processing,” in Proc. of ACL, 2020.

[7] D. Hovy and S. Spruit, “The Social Impact of Natural Language Processing,” in Proc. of ACL, 2016, pp. 591–598.

[8] B. Zmigrod, S. Vijayaraghavan, R. A. Raji, and R. Reichart, “Counterfactual Data Augmentation for Mitigating Gender, Age, and Race Bias,” in Findings of EMNLP, 2021.

[9] A. De-Arteaga et al., “Bias in Bios: A Case Study of Semantic Representation Bias in a High-Stakes Setting,” in Proc. of FAT (Fairness, Accountability, and Transparency), 2019.

[10] S. Sun, Y. Gaut, S. Tang, M. Huang, and M. Peng, “Mitigating Gender Bias in Natural Language Processing: Literature Review,” ACM Computing Surveys, vol. 55, no. 1, 2023.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in Proc. of NAACL-HLT, 2019.

[12] Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

VAISHALI_TYAGI FINAL SEM THESIS REPORT.pdf

Delhi Technological University

Document Details

Submission ID

trn:oid:::27535:98468921

Submission Date

May 30, 2025, 7:52 AM GMT+5:30

Download Date

May 30, 2025, 7:53 AM GMT+5:30

File Name

VAISHALI_TYAGI FINAL SEM THESIS REPORT.pdf

File Size

4.1 MB

43 Pages

7,223 Words

42,746 Characters

14% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

Match Groups

61 Not Cited or Quoted 14%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

12%	Internet sources
6%	Publications
11%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

61 Not Cited or Quoted 14%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

12%	Internet sources
6%	Publications
11%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	**dspace.dtu.ac.in:8080**	5%
2	Internet	**www.analyticsvidhya.com**	<1%
3	Submitted works	**University of Sheffield on 2024-10-02**	<1%
4	Internet	**repository.ju.edu.et**	<1%
5	Submitted works	**University of Wollongong on 2024-03-03**	<1%
6	Internet	**aclanthology.org**	<1%
7	Internet	**acikbilim.yok.gov.tr**	<1%
8	Submitted works	**University of Birmingham on 2020-05-27**	<1%
9	Submitted works	**Delhi Technological University on 2019-05-29**	<1%
10	Submitted works	**IUBH - Internationale Hochschule Bad Honnef-Bonn on 2024-10-12**	<1%

11 Publication

Ricardo F. Soto, Sebastián E. Godoy. "A novel feature extraction approach for skin... <1%

12 Submitted works

De Montfort University on 2023-08-31 <1%

13 Submitted works

Durban University of Technology on 2025-05-25 <1%

14 Internet

www.irjmets.com <1%

15 Internet

www.econstor.eu <1%

16 Internet

www.geeksforgeeks.org <1%

17 Internet

edurev.in <1%

18 Submitted works

Liverpool John Moores University on 2024-11-08 <1%

19 Internet

www.coursehero.com <1%

20 Internet

www.fastercapital.com <1%

21 Submitted works

University of Ulster on 2025-05-15 <1%

22 Submitted works

UC, San Diego on 2021-04-27 <1%

23 Internet

bio-protocol.org <1%

24 Internet

researchrepository.wvu.edu <1%

25	Internet	web.stanford.edu	<1%
26	Submitted works	Delhi Technological University on 2025-05-05	<1%
27	Submitted works	University of Hertfordshire on 2025-05-24	<1%
28	Publication	Salwa Belaqziz, Salma El Hajjami, Hicham Amellal, Redouan Lahmyed, Lahcen Kou...	<1%
29	Submitted works	University of Leeds on 2025-03-31	<1%
30	Internet	efatmae.github.io	<1%
31	Internet	jchr.org	<1%
32	Publication	"Advances in Artificial-Business Analytics and Quantum Machine Learning", Sprin...	<1%
33	Internet	arxiv.org	<1%
34	Internet	backend.orbit.dtu.dk	<1%
35	Internet	core.ac.uk	<1%
36	Internet	iris.unipa.it	<1%
37	Internet	nmbu.brage.unit.no	<1%
38	Internet	ntnuopen.ntnu.no	<1%

39	Internet	
par.nsf.gov		<1%
40	Publication	
Alshahrani, Saied Falah A.. "Towards Representative Pre-Training Corpora for Ara...		<1%
41	Submitted works	
Glasgow Caledonian University on 2023-08-20		<1%
42	Publication	
Hemant Kumar Soni, Sanjiv Sharma, G. R. Sinha. "Text and Social Media Analytics ...		<1%
43	Publication	
Mkhuseli Ngxande, Jules-Raymond Tapamo, Michael Burke. "Bias Remediation in ...		<1%
44	Submitted works	
University of Northampton on 2025-05-18		<1%
45	Submitted works	
University of Northampton on 2025-05-23		<1%
46	Internet	
export.arxiv.org		<1%
47	Internet	
robots.net		<1%
48	Internet	
side17.i-d-e.de		<1%
49	Internet	
www.dspace.dtu.ac.in:8080		<1%