
# Natural Language Processing for Analysing Abusive Behaviour on Social Media

Author: Linyi Li

Supervisor: Dr.Nedjma Ousidhoum

MSc Computing

School of Computer Science and Informatics, Cardiff University

Date: September 2024

**Abstract**

Social media platforms are facing challenges with the proliferation of toxic comments. These toxic comments can cause negative societal impacts and platform risks. This dissertation addresses the detection of toxic content through automated methods, leveraging Artificial Intelligence (AI), particularly Natural Language Processing (NLP), machine learning, and deep learning models. The project uses the BERT model and logistic regression for classification, aiming to create a lightweight and efficient model for real-time toxic comment detection. The trained model is deployed on a static web page and users can input text and receive immediate feedback on its toxicity. Although the models trained by this project have achieved some success in terms of overall accuracy, they still have some limitations when faced with imbalanced datasets.

# Table of Content

## 1 Introduction

Internet social networking has become an essential part of modern people's lives. It provides people with a huge platform for information and social interaction. However, abusive behaviours such as hate speech, cyberbullying, and spam have also proliferated, disrupting the community and negatively impacting users' mental health and platform reputations (Gandhi et al. 2023). Therefore, most Internet platforms have also tried various forms of review mechanisms. Currently, toxic comment reviews can be categorized into manual and automated approaches. Among these review mechanisms, the accuracy of manual review is quite high, but the efficiency cannot meet the needs of platforms with huge data volumes. This makes automatic review tools a current research hotspot (Gillespie 2018). In response to this, we propose a potential solution.

Therefore, this project focuses on the automated review category, leveraging artificial intelligence techniques, including machine learning, deep learning, and NLP, to determine whether the text content on social media is toxic. The BERT model from deep learning and the Logistic Regression model from machine learning are key technologies in this project.

At present, there are many attempts to detect toxic comments, such as various models based on deep learning and machine learning. However, after comparison, it is found that most models can complete this task well, and the performance of deep learning models is only about 2% better than that of machine learning in F1 score evaluation. Still, compared with the performance overhead of deep learning, such benefits are too small (Bonetti et al. 2023).

This project aims to train a basic lightweight model that can quickly detect harmful information in comments, posts, and messages. In addition, this project

will create a static webpage to display the effect of the trained model. Users can directly type text content and then obtain the test results.

In Chapter 1, we introduce the focus of this paper on applying natural language processing (NLP) to detect abusive behaviour on social media. We provide an overview of the foundational concepts in Chapter 2, covering Artificial Intelligence, Machine Learning, Logistic Regression, Neural Networks, Deep Learning, and essential NLP topics such as Language Models, Transformers, BERT, TF-IDF, and Classification Metrics. Chapter 3 details the methodology, including data handling, and model training with BERT. Chapter 4 presents the results and evaluation of various classification models. In Chapter 5, we analyze the impact of hardware constraints, parameter tuning, and dataset issues. Finally, Chapters 6 and 7 offer conclusions and reflections, with additional details provided in Chapters 8 and 9.

## 2 Background

Machine learning and deep learning approaches are typically used to build toxic content detection systems. In this chapter, we present a methodology that combines machine learning, deep learning, and natural language processing techniques to develop a toxic comment detection tool.

### 2.1 Artificial Intelligence

Artificial Intelligence can be defined as the simulation of human intelligence processes, which include learning, reasoning, and self-correction by machines (Hunt 1975). Artificial intelligence includes a variety of technologies such as machine learning, neural networks, natural language processing, and robotics.

#### 2.1.1 Machine Learning

Machine learning involves a computer program's ability to improve its performance on tasks by learning from past data (Dutt 2019). Machine learning includes various approaches, such as supervised learning, where the model is trained on labelled data to learn the relationship between inputs and outputs, and unsupervised learning, where the model uncovers hidden patterns in unlabeled data (Zhou 2021). With the development of data science and computer performance, machine learning has been widely used in natural language processing, image recognition and other fields.

#### 2.1.2 Logistic Regression Model

The logistic regression model is a widely used linear model for classification tasks in machine learning. Its goal is to understand a binary or proportional response based on one or more predictors(Hilbe 2009). The model works by applying a logistic function to a linear combination of predictors, converting the output into a probability value between 0 and 1(Kleinbaum 2010).

Firstly, the logistic regression model computes a weighted sum of input features through a linear equation, expressed as follows:

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

Here, w represents the weight vector, x denotes the feature vector, and b is the bias term. The linear combination z represents the weighted sum of features.
Next, the model utilizes the sigmoid function to transform this result into a probability value between 0 and 1:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

This function transforms the result into a number between 0 and 1, indicating how likely it is that the input belongs to one of the two categories we are trying to predict.

To assess how well the logistic regression model performs, we use a measure called Log-Loss. Log-Loss is a way to evaluate the accuracy of the model's predictions by comparing the predicted probabilities to the actual outcomes.
Log: In this context, "log" refers to the logarithm function, which is used to measure the likelihood of the predictions. The logarithm helps quantify how confident the model's predictions are, with higher penalties for incorrect predictions.

Loss: "Loss" represents the cost or penalty associated with incorrect predictions. In machine learning, loss functions are used to evaluate how well a model's predictions match the actual outcomes. The aim is to minimize this loss to improve the model's performance.

The formula for Log-Loss is:

$$Log - Loss = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

In this formula, **m** is the total number of samples, $y_i$ is the actual label of the sample, and $\hat{y}_i$ is the predicted probability of the **I** sample being in the positive class. By minimizing this loss function, the model's parameters **w** and **b** can be optimized.

### 2.1.3 Neural networks

Neural networks, also known as connectionist networks, are computational models designed to mimic the way knowledge is stored, represented, and processed(Judd 1990). They consist of many interconnected processing units that perform simple calculations in parallel. This approach focuses on using multiple processors working together simultaneously to solve complex problems. Neural networks may provide new ways to build computational systems and offer insights into brain function. Their design, input/output methods, and behaviour are often compared to the workings of biological brains. Generally, there are two ways of learning in neural networks: 'learning' and 'retrieval'.

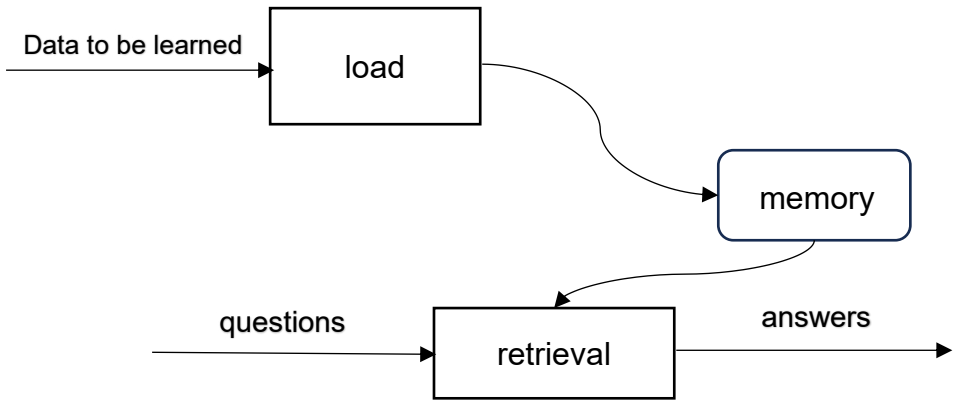


Figure 1: two ways of learning

In the learning model, data are stored in memory base and in the retrieval model,

data are recalled from data base.

The simplest neural network is the perceptron, which consists of a single layer and is used for basic classification tasks. More complex networks with multiple layers, called deep neural networks, can learn intricate patterns from large datasets and are fundamental to deep learning.

### 2.1.4 Deep Learning

Deep learning is a branch of artificial intelligence (AI) based on the concept of neural networks. Unlike the basic neural networks mentioned in the previous section that are suitable for simple tasks, deep learning uses higher-level multi-layer networks to analyze data (Srinivasa Rao 2023). These multi-layer grids are also known as DNNs (Deep Neural Networks), which are a type of artificial intelligence model designed to learn from data through a series of steps and consist of multiple stages where each stage processes data in steps, learning simple patterns in the early stages and more complex patterns in the later stages. These networks consist of multiple layers of interconnected neurons that allow the model to learn complex patterns from large datasets autonomously, without the need for manual feature extraction or intervention (Mueller 2019). In contrast, traditional machine learning techniques typically require significant human involvement to manually engineer features, adjust parameters, and refine algorithms to optimize performance. By leveraging the deep architecture of neural networks, deep learning models excel in extracting hierarchical features from raw data, making them highly effective for a wide range of complex tasks.

## 2.2 Natural Language Processing

Natural Language Processing (NLP) is the branch of artificial intelligence focused on the interaction between computers and human languages, enabling

computers to understand, interpret, and generate human language(Kochmar 2022). This field currently integrates techniques such as statistics and machine learning to apply to various tasks, including text classification, sentiment analysis, language translation, and information extraction(Gacovski 2021).

### 2.2.1 language model

A language model can be defined as a statistical tool used to understand and generate human language. It operates by analyzing large corpora of text to learn patterns and relationships between words, allowing it to estimate the probability of a word or phrase occurring in a specific context.

There are two primary types of language models: statistical and neural. statistical and neural. Statistical language models rely on mathematical formulas to estimate word probabilities based on frequency counts. While Neural language models use neural networks to learn complex patterns and semantic relationships within text, providing more nuanced and contextually appropriate predictions.

### 2.2.2 Transformer

Transformers are a type of deep learning model designed to handle sequential data, such as text. Transformers use a mechanism called self-attention to process all words in a sentence simultaneously. This parallel processing allows transformers to capture relationships between words regardless of their distance from each other in the text. These models are based on an encoder-decoder architecture. The encoder is the part of the model that takes the input data (like a sentence) and converts it into a numerical format that captures the meaning of the words and their relationships. The decoder is the part that takes this encoded information and translates it back into a human-readable form, such as generating a translated sentence in another language.

Self-attention is a method that allows the model to focus on different parts of the input sequence when processing a word or token. It computes a score for each word in the sequence based on its relationship to every other word, allowing the model to dynamically assign different levels of importance to each word. This means that the model can effectively learn which words in a sentence are more relevant to each other.

In addition, transformers are often described as bidirectional because, during encoding, they consider the context from both directions of the text: from the beginning to the end and from the end to the beginning. (Vaswani et al. 2023). This bidirectional approach allows the model to capture a more complete understanding of the context, as it takes into account the entire sequence of words at once, rather than processing them one by one in a fixed order.

### 2.2.3 BERT and Fine-tuning BERT

BERT stands for Bidirectional Encoder Representations from Transformers, and is a model derived from the Transformer architecture(Raaijmakers 2022). Unlike a full Transformer that includes both encoder and decoder parts, BERT consists solely of the encoder. It was designed to generate word representations, known as vector embeddings, by using context from both directions in the text(Koroteev 2021). BERT achieves this through masked language modelling, where certain words in a sequence are masked, and the model predicts these masked words based on the surrounding context(Jawahar et al. 2019). Because of this, BERT and Transformers have the same characteristics, they can understand the entire text instead of interpreting each word independently. From this point of view, using BERT for text processing will be much more effective than using only TF-IDF for text processing.

Fine-tuning is a common method for adjusting and optimizing the performance of the BERT model and the specific approach to fine-tuning varies depending

on the type of task. Fine-tuning involves taking a pre-trained BERT model and further training it on a smaller set of examples that are specific to the desired task. During this process, the model is adjusted to learn the particular patterns and rules needed for that task. This allows BERT to perform very well on different types of language tasks, such as understanding whether a review is positive or negative or classifying text into categories, without needing to create a new model from scratch for each task. Fine-tuning is essential because it adapts BERT's general language understanding to be more effective for the specific task at hand.

### 2.2.4 TF-IDF

The TF-IDF algorithm is a statistical method used to evaluate the importance of words in text, based on the frequency of their appearance within the text (Liu et al. 2018).

Term frequency (TF) refers to the frequency of occurrence in a document, while inverse document frequency (IDF) evaluates the importance of a term by looking at its occurrence in multiple documents (Rahman et al. 2021). It combines two measures: Term Frequency (TF) and Inverse Document Frequency (IDF). The two formulas for TF and IDF are:

1. Term Frequency (TF): This measures how frequently a term (word) appears in a document. The simplest formula for term frequency is:

$$TF(t,d) = \frac{f(t,d)}{N_d}$$

In this formula, $f(t,d)$ is the number of times the term t appears in document d. $N_d$ is the total number of terms in the document d.

2. Inverse Document Frequency (IDF): This measures how important a term is across the entire corpus of documents. The formula for inverse document frequency is:

$$IDF(t, d) = \log\left(\frac{N}{n_t}\right)$$

In this formula, N is the total number of documents in the corpus D. $n_t$ is the number of documents in which the term t appears.

3. TF-IDF Calculation: The TF-IDF score for a term t in document d is then calculated by multiplying the TF and IDF values:

$$TF - IDF(\mathrm{T}, \mathrm{d}, \mathrm{D}) = \mathrm{TF}(\mathrm{t}, \mathrm{d}) \times \mathrm{IDF}(\mathrm{t}, \mathrm{D})$$

### 2.2.5 Classification

Classification is a fundamental task in machine learning where the goal is to assign items to specific categories or classes based on their features. This involves training a model on a dataset with known labels, where each item is associated with a predefined class. The model learns to recognize patterns and relationships in the data that are indicative of each class. After training, the model can then categorize new, unseen items into one of the classes it has learned. Classification is used in many applications. For example, emails might be classified as "spam" or "not spam," and news articles might be categorized into topics such as "sports," "politics," or "technology."

### 2.2.6 Classification Metrics

Evaluation metrics are tools used to assess the performance of a model in completing classification tasks (Ferrer 2022). The most basic and intuitive metric is the 2x2 confusion matrix

| Actual / Predicted | Positive (1) | Negative (0) |
|---|---|---|
| Positive (1) | True Positives (TP) | False Negatives (FN) |
| Negative (0) | False Positives (FP) | True Negatives (TN) |

Table 1: Classification Metrics

In this table,

True Positives (TP): The model correctly predicts the positive class.

False Positives (FP): The model incorrectly predicts the positive class when the actual class is negative.

False Negatives (FN): The model incorrectly predicts the negative class when the actual class is positive.

True Negatives (TN): The model correctly predicts the negative class.

Based on the confusion matrix, several metrics are derived:

1. Precision (P): the ratio of true positives to the sum of true positives and false positives. It is calculated as:

$$Precision = P = \frac{TP}{TP + FP}$$

2. Sensitivity (S): the ratio of true positives to the sum of true positives and false negatives. It is calculated as:

$$Recall = S = \frac{TP}{TP + FN}$$

3. Accuracy (A): the ratio of true positives and true negatives to the total number of predictions. It is calculated as:

$$Accuracy = A = \frac{TP + TN}{TP + TN + FP + FN}$$

4. F1 Score: a function of precision and recall, primarily used to evaluate model performance on imbalanced datasets. It is calculated as:

$$F1\ Score = \frac{2 \times TP}{2 \times TP + FP + FN}$$

These metrics are important for assessing model performance and guiding parameter tuning and optimization (Naidu et al. 2023).

### 2.3 Toxic Language Detection

In the area of toxic comment detection, AI can develop sophisticated models that can analyze large amounts of text data, identify patterns of abusive language, and make accurate predictions about new comments (Boden and ScienceDirect 1996).

#### 2.3.1 Hate Speech

The term "hate speech" was first introduced by legal scholar Matsuda to specifically refer to racial hate speech (Guillén-Nieto 2023). In general, hate speech is a language that expresses hatred or promotes violence against an individual or group. On social media, hate speech often targets groups based on common characteristics such as race, gender, religion, or sexual orientation. Contemporary linguistic and legal scholars define hate speech as a language that incites harmful actions or spreads hostility, which on social media can include using racial slurs, sharing manipulated images that demean specific groups, or spreading misinformation designed to provoke hatred (Waldron 2012). Social media platforms, due to their broad reach and anonymity, have become a prominent space where hate speech can proliferate, posing significant challenges for content moderation and regulation. For example, Twitter faces ongoing difficulties in controlling the spread of toxic comments and addressing the prevalence of harmful speech. For instance, Instagram has struggled with abusive comments in their community sections, where moderation efforts often lag behind the pace of abusive content generation.

### 2.3.2 Abusive Language

Abusive language is a form of toxic language that is often personal and not necessarily directed at a specific group, unlike hate speech. On social media, abusive language can manifest as cyberbullying, where individuals are targeted with harmful remarks or insults (Saleem 2021). It is frequently seen in heated online debates that spiral out of control, where users engage in hostile exchanges (Niemann et al. 2020). The anonymity provided by social media platforms can encourage people to use abusive language, including racist, sexist, or otherwise discriminatory remarks, that they might not express in face-to-face interactions. This anonymity can escalate conflicts and foster environments where harmful speech becomes more prevalent and challenging to manage. For instance, Instagram has struggled with abusive comments in their community sections, where moderation efforts often lag behind the pace of abusive content generation.

## 2.4 Challenges in Detecting Toxic Comments

### 2.4.1 Evolving language and slang

The internet has become a breeding ground for new forms of communication. Social media platforms and online communities play a significant role in creating and popularizing new slang and idiomatic expressions. These expressions often originate from specific subcultures or online groups and spread widely through viral trends and memes (Al-Kadi and Ahmed 2018).

Traditional models and lexicons often struggle to adapt to the fast changes in language on social media. Once neutral words can develop offensive meanings, and new slang with specific meanings can appear quickly (Pei et al. 2019). Static models find it hard to identify these changes in terminology. To overcome this, models can be trained on large, recent datasets that are regularly updated to include the latest toxic language and emerging slang (Rothe et al. 2023).

This approach improves the model's ability to detect new and evolving terms.

However, if datasets remain imbalanced with too many neutral comments compared to toxic ones, the model may still struggle to accurately detect new and evolving terms.

### 2.4.2 Imbalance in Datasets

Dataset balance is a critical issue. Hate speech and offensive comments, although less frequent, attract significant attention and can have a major impact when they do occur. In real online forums and existing datasets, benign or normal comments often vastly outnumber toxic comments (Ibrahim et al. 2018). This imbalance poses challenges for model training. When a dataset is dominated by normal comments, the model can become biased toward the majority class and may struggle to identify the less frequent toxic comments effectively. This can lead to poor detection of harmful content, as the model might not learn to recognize the subtle features of offensive language (Wang and Zhang 2021).

This chapter covered the fundamentals of machine learning, deep learning, and natural language processing techniques as they relate to toxic comment detection. The following chapter will present specific methods for applying these techniques.

## 3 Methodology

We begin by using BERT to extract vector features from the text data, which are then fed into a logistic regression model for training. Subsequently, we train a fine-tuned BERT model and compare the performance of the two approaches. Finally, we develop a static webpage to deploy the model for trial operations.

### 3.1 Data

#### 3.1.1 Data source

For this project, we use datasets in [1]Kaggle to train the model. The names of these two datasets are *Toxic Comment Classification Challenge* [no date] and *Jigsaw Unintended Bias in Toxicity Classification* [no date] . The datasets aim to develop machine learning models that detect toxic comments while minimizing unintended bias, especially related to identity mentions. The goal is to improve fairness and accuracy in identifying different types of toxicity in online conversations. The datasets include comments from various sources, with labels indicating toxicity and identity mentions to optimize model performance against biases.

#### 3.1.1 Data structure

Datasets provided by Kaggle are labeled with detailed annotations, and structured as follows. The number 1 indicates that the comment has the behavior described by the label, and 0 indicates that it does not exist.

| Comment_text | toxic | Severe_toxic | obscene | threat | insult | Identity_hate |
|---|---|---|---|---|---|---|
| alignment…. of DuLithgow | 0 | 0 | 0 | 0 | 0 | 0 |
| Not sure…. it contain? | 0 | 0 | 0 | 0 | 0 | 0 |

[1] Kaggle: Your Machine Learning and Data Science Community

| | | | | | | |
|---|---|---|---|---|---|---|
| UNBLOCK ME…TO FREE | 1 | 0 | 0 | 0 | 0 | 0 |
| [expletive], block me, you [expletive]! | 1 | 0 | 1 | 0 | 0 | 0 |
| … | … | … | … | … | … | … |

Table 2: Example of toxic comments

### 3.1.2 Data Cleaning

To simplify the data structure and train a binary classifier, we labelled all forms of toxic comments, such as hate speech and sexism, as toxic (label 1), while non-toxic comments were labelled as 0 and their structure is as follows.

| Content_text | Is_toxic |
|---|---|
| Try making a positive contribution. | 0 |
| What are they teaching you guys about COIN? | 0 |
| UNBLOCK ME…TO FREE | 1 |
| [expletive], block me, you [expletive]! | 1 |
| … | … |

Table 3: Example of toxic comments

## 3.2 Model Training

### 3.2.1 Use of the BERT Model

To extract feature vectors using BERT, we first preprocess the text data by converting it to lowercase and removing punctuation. We initialise the tokenizer of a BERT uncased model. The tokenizer converts the text data into input tensors suitable for the BERT model. Finally, the extracted feature vectors are used as input into the logistic regression model for training.

### 3.2.2 Fine-tuning BERT

We use a pre-trained BERT model (BertForSequenceClassification) with weights from the "bert-base-uncased" model, configured for binary classification. The model is optimized with the AdamW optimizer, which is well-suited for fine-tuning large language models like BERT.

The training process runs for three epochs, meaning the model goes through the entire training dataset three times to learn and update its parameters. During this process, the model calculates the loss using cross-entropy, a metric that measures how well the model's predictions match the actual results. Lower cross-entropy means better performance. To track how well the model is learning, the average loss across batches is monitored. Meanwhile, the model's performance on a separate validation set is evaluated to calculate validation loss and accuracy without updating the model's weights. This helps to prevent overfitting, which occurs when the model learns the training data too well, including its noise and specific details, and performs poorly on new, unseen data.

### 3.2.3 Use of TF-IDF

To extract feature vectors using TF-IDF, we apply the TF-IDF vectorizer to convert the cleaned text data into numerical feature vectors. The vectorizer calculates the term frequency-inverse document frequency scores for each term in the dataset, capturing the importance of words relative to their occurrence in the documents. These feature vectors are then used as input into the logistic regression model for training.

### 3.2.4 Input and Output

The input refers to the textual data of online comments or posts that need to be classified based on their content. Each comment or post, provided in the form of a text string, is fed into the model and the output is the predicted classification

label assigned to each input comment. This label indicates whether the comment is "toxic" or "not toxic."

### 3.2.5 Example of Input and Output

**Input:**" You are so stupid and worthless"

When this comment is fed into the logistic regression model and the fine-tuned BERT model, the model processes the text to understand its content. After analyzing the input, the model produces an output:

**Output:**

Logistic Regression Model Result: Toxic
Fine-Tuned BERT Model Result: Toxic

In this example, the model identifies the abusive language in the comment, classifying it as toxic.

**Caution**: The image below contains content from a toxic comment. Viewer discretion is advised.

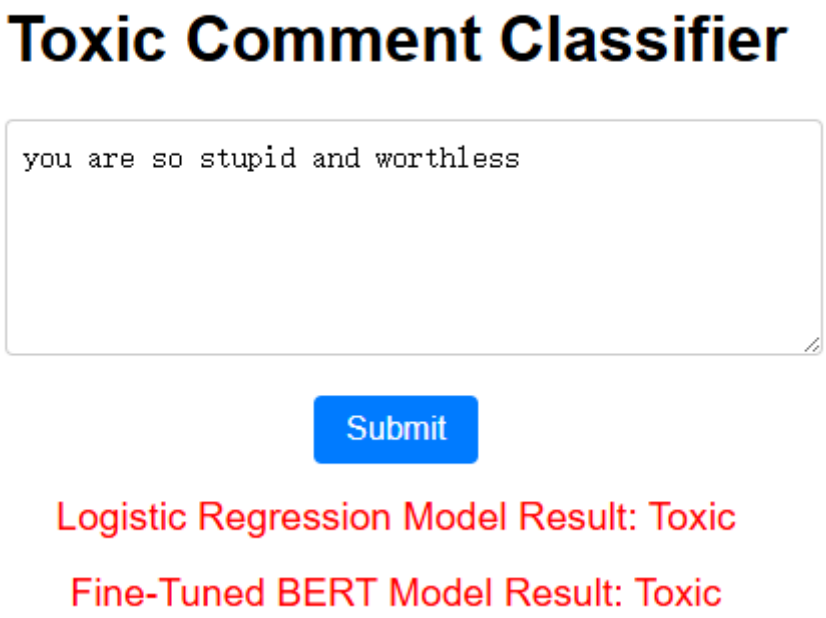


Figure 2: interface of example

### 3.3 Static Web Page Deployment

The static webpage for the Toxic Comment Classifier was designed using HTML and CSS. The interface features a text area for user input and a submit button. When the user submits a comment, the text is sent to the Flask backend for processing. The Flask application is responsible for preprocessing the input text, and using the trained model to classify the comment as "Toxic" or "Not Toxic."

## 4 Results and Evaluation

This section presents the results of the models evaluated in this project, including the logistic regression model, BERT model, and the original specification using TF-IDF with logistic regression. Each model's performance is assessed using classification metrics such as accuracy, precision, recall, and F1-score to provide a overview of how well each model performs on the binary classification task.

The TF-IDF with logistic regression model, originally intended as the primary approach, serves as a baseline for comparison. In contrast, the logistic regression, which uses BERT-extracted feature vectors, and the BERT model itself represent more advanced techniques to improve classification accuracy and handle class imbalance.

### 4.1 Classification Metrics Report

| Metric | BERT + Logistic Regression | BERT Model | TF-IDF + Logistic Regression |
|---|---|---|---|
| Overall Accuracy | **93.28%** | 69.54% | 53.12% |
| precision | **82%** | 79% | 60% |
| recall | **79%** | 32% | 58% |
| F1-score | **81%** | 46% | 52% |

Table 4: Classification Metrics Result

### 4.2 Explanation

#### 4.2.1 Logistic regression model

In this project, the overall accuracy of the model is 93.28%, indicating a high level of predictive capability. Specifically, the model performed well on class 0 (negative class), achieving 66,734 true negatives and 3,000 false negatives.

For class 1 (positive class), the model identified 2,152 true positives and 4,738 false positives. The macro average and weighted average metrics further suggest that the model has some limitations in handling class imbalance. Overall, while the model demonstrates high accuracy, there is room for improvement in predicting the minority class.

```
PS E:\LAB3> python train_logistic_regression.py
Accuracy: 0.932762580914596
Confusion Matrix:
 [[66734  2152]
 [ 3000  4738]]
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.97      0.96     68886
           1       0.69      0.61      0.65      7738

    accuracy                           0.93     76624
   macro avg       0.82      0.79      0.81     76624
weighted avg       0.93      0.93      0.93     76624
```

Figure 3: Report of Logistic regression model

### 4.2.2 BERT model

In this model, the overall validation accuracy of the model is 69.54%, indicating moderate predictive capability. The confusion matrix shows that for class 0 (negative class), the model correctly identified 2,300 instances (true negatives) but incorrectly labelled 1,107 instances as positive (false positives). For class 1 (positive class), the model correctly identified 535 instances (true positives) but missed 135 instances, labelling them incorrectly as negative (false negatives).

The model's performance indicates notable limitations. The high number of false positives and false negatives suggests challenges in distinguishing between classes, which points to issues with class imbalance. This imbalance affects the model's ability to accurately identify the minority class. Additionally, the model’s moderate accuracy highlights the need for further refinement and

potentially more sophisticated methods to address these classification errors and improve performance.

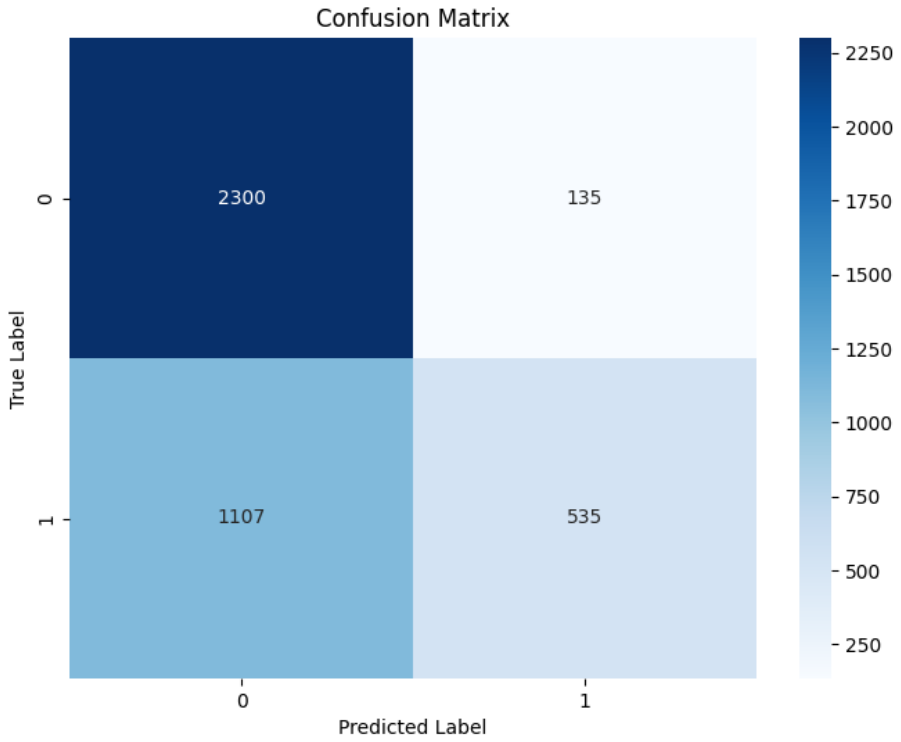


Figure 4: Confusion matrices

### 4.2.3 TF-IDF and Logistic Regression Model

TF-IDF was planned to be used for feature extraction, followed by training with a logistic regression model. However, as indicated by the report, the overall accuracy was 54,309 out of 102,580, which is considerably low for practical applications. The confusion matrix shows a substantial number of misclassifications, with the model failing to accurately distinguish between the two classes. Specifically, class 0 had a high recall of 54,309 out of 64,078 but a low precision of 54,309 out of 114,278, indicating a high rate of false positives. Conversely, class 1 had a higher precision of 27,052 out of 37,021 but a very low recall of 27,052 out of 89,186, signifying a high rate of false negatives. The macro average F1-score of 0.52 and the weighted average F1-score of 0.50

further reflect the inadequacy of the model in handling the classification task effectively. These results underscore the limitations of TF-IDF in capturing contextual information, necessitating the shift to more advanced deep learning techniques for improved performance.

```
Accuracy: 0.5312018489984591
Confusion Matrix:
 [[54309  9669]
 [62134 27052]]
Classification Report:
              precision    recall  f1-score   support

           0       0.47      0.85      0.60     63978
           1       0.74      0.30      0.43     89186

    accuracy                           0.53    153164
   macro avg       0.60      0.58      0.52    153164
weighted avg       0.62      0.53      0.50    153164
```

Figure 5: Confusion matrices

## 5 Analysis

Based on the performance results, the logistic regression model outperformed the BERT model, demonstrating advantages in training and deployment efficiency. Logistic regression requires less computational power and time, making it quicker to train and easier to deploy due to its simpler architecture. In contrast, the BERT model, although potentially more effective, faced constraints due to its complexity. To manage training time and resource demands, we compromised by reducing the dataset to 3,000 samples. This reduction means the BERT model might not fully capture the nuances of a larger dataset. Despite its current performance, the BERT model remains valuable for future use with adjustments to address its higher resource requirements.

### 5.1 Hardware Constraints Affecting Training Intensity

Due to hardware limitations, certain compromises were made regarding the model's performance. The original dataset contained 76,224 samples, but using the BERT model on such a large dataset without GPU acceleration would result in very slow processing speeds. Therefore, the dataset size was reduced to one-quarter of its original size, and the original test set was entirely removed, with 15,244 samples of the training set allocated as the test set. This reduction in dataset size may lead to insufficient learning of certain toxic comment types, thereby affecting the model's accuracy.

During the BERT model training, only 3,000 data samples were retained for training and testing to save time. This limited dataset resulted in suboptimal performance for the BERT model. Despite completing three epochs, the final performance remained unsatisfactory.

### 5.2 Inadequate parameter tuning

Inadequate parameter tuning led to poor performance in the F1 score and recall metrics for the trained models. In the process of training models, we only roughly adjusted the training parameters instead of repeatedly challenging the training parameters to select the best training solution. Therefore, the two models after training performed poorly in terms of F1 score and recall indicators.

### 5.3 Shortcomings of Datasets

In this project, the dataset has several issues, including missing labels, corrupted text, and empty entries. These problems lead to incomplete training data and unreliable information, which negatively impacts the model’s ability to learn effectively. As a result, the model's performance is compromised, affecting its accuracy and overall effectiveness.

## 6 Conclusion

We developed a lightweight toxic comment detection tool that can help social media quickly review whether the content posted on the platform is toxic.

The results show that under the same hardware conditions and similar training time, the logistic regression model can accept more training data to achieve better model performance. Therefore, the logistic regression model has certain advantages in this project.

in the future follow-up work can include:

During model training, we performed only rough parameter adjustments rather than extensive optimization. Therefore, the two models after training performed poorly in terms of F1 score and recall indicators. To improve model performance, more precise tuning methods can be employed, such as grid search, random search, and Bayesian optimization. These techniques allow for a thorough exploration of the parameter space to identify the optimal settings, thus enhancing model performance. Additionally, cross-validation, hyperparameter tuning tools, and early stopping can further improve model generalization and training efficiency. Future efforts should focus on these refined adjustment methods to better address dataset imbalance and optimize model performance.

We will also generate synthetic data to enhance the diversity of the training dataset. Techniques such as back-translation and synonym replacement will be employed to create variations of existing data. Back-translation can be regard as translating text to another language and then back to the original language, which can introduce new phrasing and context while retaining the original meaning. Synonym replacement involves substituting words with their synonyms to produce alternative text versions. These methods aim to augment the dataset with diverse examples, improving the model's robustness (its ability

to handle unexpected or varied inputs) and generalization capabilities (its ability to perform well on new, unseen data). By incorporating synthetic data, we expect to address dataset imbalance and enrich the training process, leading to better performance and more accurate predictions.

In the field of natural language processing, numerous advanced models have been developed, such as RoBERTa, GPT, and T5. These models have shown significant improvements in handling various NLP tasks by leveraging more complex architectures and larger datasets. However, we only experimented with logistic regression and BERT models, which limits the ability to compare their effectiveness against more state-of-the-art approaches. The lack of comparison with these advanced models reduces the comprehensiveness of the experiment, as it does not fully explore the potential benefits and drawbacks of using different NLP models for toxic comment detection.

To improve model performance, we will first restore the data that was previously removed from the dataset. Following this, we will expand the dataset further based on the performance of the models. This approach aims to enhance the training of fine-tuned BERT models by providing a more comprehensive dataset, potentially improving the model's ability to capture nuanced patterns and increasing overall performance.

## 7 Reflection

Throughout the completion of this project, I gained a deep understanding of natural language processing and machine learning algorithms, applying these concepts to the problem of toxic comment detection through extensive research and experimentation.

One of the most valuable aspects of this project was the use of the advanced BERT model for feature extraction, followed by classification using logistic regression. This combination offered a solution to the toxic comment detection tool, though it also highlighted challenges in balancing model performance with computational efficiency. This experience provided me with a deeper understanding of the issue of resource constraints in AI model development.

Another significant challenge was addressing the class imbalance in the dataset. While the model achieved high overall accuracy, its performance on minority classes was less satisfactory. This flaw underscored the importance of data quality, quantity, and sampling techniques in machine learning, motivating me to explore more sophisticated methods for handling imbalanced data in future projects.

Reflecting on the entire process, I gained valuable experience in applying AI theory to the practical task of social media moderation. We not only enhanced my analytical skills in data preprocessing but also deepened my understanding of the relationship between AI and social ethics. This experience has provided a solid foundation for my future exploration of the impact of AI on society.

## 8 References

Al-Kadi, A.M.T. and Ahmed, R.A. 2018. EVOLUTION OF ENGLISH IN THE INTERNET AGE. *Indonesian Journal of Applied Linguistics* 7(3), p. 727. doi: 10.17509/ijal.v7i3.9823.

Boden, M.A. and ScienceDirect. 1996. *Artificial intelligence*. San Diego: Academic Press.

Bonetti, A., Martínez-Sober, M., Torres, J.C., Vega, J.M., Pellerin, S. and Vila-Francés, J. 2023. Comparison between Machine Learning and Deep Learning Approaches for the Detection of Toxic Comments on Social Networks. *Applied sciences* 13(10), pp. 6038-. doi: 10.3390/app13106038.

Dutt, S. 2019. *Machine learning*. [First edition]. Uttar Pradesh, India: Pearson India.

Ferrer, L. 2022. *Analysis and Comparison of Classification Metrics*. Available at: https://arxiv.org/abs/2209.05355v4 [Accessed: 8 August 2024].

Gacovski, Z. 2021. *Natural language processing*. Burlington, Ontario: Arcler Press.

Gandhi, H., Bachwani, R. and Nanade, A. 2023. Detecting Toxic Comments Using FastText, CNN, and LSTM Models. In: *Advances in Computing and Data Sciences*. Communications in Computer and Information Science. Switzerland: Springer, pp. 241–252. doi: 10.1007/978-3-031-37940-6_20.

Gillespie, T. 2018. *Custodians of the Internet: Platforms, Content Moderation, and the Hidden Decisions That Shape Social Media*. New Haven, CT: Yale

University Press. doi: 10.12987/9780300235029.

Guillén-Nieto, V. 2023. *Hate Speech: Linguistic Perspectives*. De Gruyter. Available at: https://www.degruyter.com/document/doi/10.1515/9783110672619/html [Accessed: 3 July 2024].

Hilbe, J. 2009. *Logistic regression models*. 1st ed. Boca Raton: Chapman & Hall/CRC. doi: 10.1201/9781420075779.

Hunt, E.B. 1975. *Artificial intelligence*. New York, New York ; Academic Press, Inc.

Ibrahim, M., Torki, M. and El-Makky, N. 2018. Imbalanced Toxic Comments Classification Using Data Augmentation and Deep Learning. IEEE, pp. 875–878. Available at: https://ieeexplore.ieee.org/document/8614166 [Accessed: 25 August 2024].

Jawahar, G., Sagot, B. and Seddah, D. 2019. What does BERT learn about the structure of language? Available at: https://inria.hal.science/hal-02131630 [Accessed: 8 August 2024].

*Jigsaw Unintended Bias in Toxicity Classification*. [no date]. Available at: https://kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification [Accessed: 31 August 2024].

Judd, J.S. 1990. *Neural network design and the complexity of learning*. Cambridge, Mass: MIT Press.

Kleinbaum, D.G. 2010. *Logistic regression: a self-learning text*. Third edition.

New York, NY: Springer New York. doi: 10.1007/978-1-4419-1742-3.

Kochmar, E. 2022. *Getting Started with Natural Language Processing*. Simon and Schuster.

Koroteev, M.V. 2021. *BERT: A Review of Applications in Natural Language Processing and Understanding*. Available at: https://arxiv.org/abs/2103.11943v1 [Accessed: 8 August 2024].

Liu, C., Sheng, Y., Wei, Z. and Yang, Y.-Q. 2018. Research of Text Classification Based on Improved TF-IDF Algorithm. In: *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*. pp. 218–222. Available at: https://ieeexplore.ieee.org/abstract/document/8492945 [Accessed: 8 August 2024].

Mueller, J.P. 2019. *Deep learning*. 1st edition. Hoboken, New Jersey: For Dummies.

Naidu, G., Zuva, T. and Sibanda, E.M. 2023. A Review of Evaluation Metrics in Machine Learning Algorithms. In: Silhavy, R. and Silhavy, P. eds. *Artificial Intelligence Application in Networks and Systems*. Cham: Springer International Publishing, pp. 15–25. doi: 10.1007/978-3-031-35314-7_2.

Niemann, M., Riehle, D.M., Brunk, J. and Becker, J. 2020. What Is Abusive Language?: Integrating Different Views on Abusive Language for Machine Learning. In: Grimme, C., Preuss, M., Takes, F. W., and Waldherr, A. eds. *Disinformation in Open Online Media*. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 59–73. Available at: http://link.springer.com/10.1007/978-3-030-39627-5_6 [Accessed: 25 June 2024].

Pei, Z., Sun, Z. and Xu, Y. 2019. Slang Detection and Identification. In: Bansal, M. and Villavicencio, A. eds. *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, pp. 881–889. Available at: https://aclanthology.org/K19-1082 [Accessed: 26 July 2024].

Raaijmakers, S. 2022. *Deep learning for natural language processing*. [First edition]. Shelter Island, New York: Manning Publications Co. LLC.

Rahman, S., Talukder, K.H. and Mithila, S.K. 2021. An Empirical Study to Detect Cyberbullying with TF-IDF and Machine Learning Algorithms. In: *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*. pp. 1–4. Available at: https://ieeexplore.ieee.org/abstract/document/9641251 [Accessed: 8 August 2024].

Rothe, M., Lath, R., Kumar, D., Yadav, P. and Aylani, A. 2023. Slang language Detection and Identification In Text. In: *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. pp. 1–5. Available at: https://ieeexplore.ieee.org/document/10308036/?arnumber=10308036 [Accessed: 26 July 2024].

Saleem, H.M. 2021. *Abusive Language through the Lens of Online Communities*. ProQuest Dissertations & Theses. Available at: https://www.proquest.com/docview/2701132119/abstract/?pq-origsite=primo [Accessed: 29 July 2024].

Srinivasa Rao, A.S.R. 2023. *Deep learning*. Amsterdam, Netherlands: Elsevier.

*Toxic Comment Classification Challenge*. [no date]. Available at: https://kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge [Accessed: 31 August 2024].

Vaswani, A. et al. 2023. Attention Is All You Need. Available at: https://arxiv.org/abs/1706.03762 [Accessed: 5 August 2024].

Waldron, J. 2012. *The harm in hate speech*. Cambridge, Mass: Harvard University Press. doi: 10.4159/harvard.9780674065086.

Wang, Z. and Zhang, B. 2021. Improved Bi-GRU Model for Imbalanced English Toxic Comments Dataset. New York, NY, USA: ACM, pp. 24–29. doi: 10.1145/3508230.3508234.

Zhou, Z.-H. 2021. *Machine learning*. Gateway East, Singapore: Springer.