Enhancing Toxic Comment Classification: A Deep Learning Approach with Pre-trained Language Models

Khushi Tejwani[1], Ved Naik[2], Aanya Lari[3], Dhruvin Jhaveri[4]

Department of Computer Engineering, Mukesh Patel School of Technology Management and Engineering, NMIMS Mumbai

[1]khushi.tejwani178@nmims.edu.in

[2]ved.naik109@nmims.edu.in

[3]aanya.lari060@nmims.edu.in

[4]dhruvin.jhaveri@nmims.edu.in

In the digital age, our lives are inundated with text data, especially short text, due to online communication, e-commerce, and digital devices. However, this transformation has also unveiled a darker side - the prevalence of harmful, offensive, and toxic comments, including hate speech and harassment. This toxicity poses serious threats to well-being, societal harmony, and online community integrity. In this study, we explore the effectiveness of two recurrent neural network (RNN) architectures, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), for toxic comment classification. We evaluate these models using key metrics, such as accuracy, precision, recall, and F1-score. Our results show that the LSTM excels in recall, identifying toxic comments at the expense of precision, while the GRU achieves superior precision and overall accuracy. Model selection should align with specific project goals and trade-offs between precision and recall.

Index Terms—Enhancing Toxic Comment Classification, GRU, LSTM, Toxicity, Comment, Function, Deep Learning, RNN

I. Introduction

ONline harassment can have far-reaching consequences, with reduced user participation in online projects being a common outcome [6]. For instance, a 2014 Pew Report highlighted that 73% of adult internet users have witnessed online harassment, with 40% personally experiencing it [2]. Despite efforts to enhance online safety through techniques like crowd-sourced voting schemes and comment reporting, these methods are often inefficient in predicting and preventing potential toxicity [7].

The sheer volume of content online necessitates the adoption of text mining tools that can efficiently handle various document operations in a timely and precise manner. Text classification, defined as the task of associating each document within a given set of documents D with a specific class or label from the set C, is one such operation that is a commonly discussed topic within the field of natural language processing. Significant research efforts have been directed towards the development of automated systems for classifying toxic comments on social media platforms. The objective for these systems is to identify and categorize toxic comments, thereby enabling timely moderation and fostering a healthier environment [4]. Comment toxicity classification, therefore, is instrumental in mitigating the impact of harmful content on individuals and society at large [5]. This process plays a vital role in moderating discussions, organizing information, and safeguarding online communities against harmful content.

A. Motivation

This project is propelled by the critical imperative to combat the proliferation of toxic content in online platforms and social media. The rapid advancement of digital communication, ecommerce, and the pervasive use of digital devices have engendered a voluminous stream of text data in the digital sphere [21]. However, this remarkable digital transformation has also exposed the darker facet of online discourse, characterized by the widespread dissemination of hateful, offensive, and toxic comments.

Toxic comment classification is a fundamental requirement in the digital age, as it has direct implications for individual well-being, societal harmony, and the overall health of online communities [23]. Hate speech, harassment, and abusive language have infiltrated various online platforms, posing threats to user participation and the integrity of online discussions [22]. A deeper understanding of the motivating factors behind toxic comments is imperative, especially as hate speech and abusive language can incite real-world harm.

Existing approaches to toxic comment classification often encounter challenges related to the evolving nature of online toxicity, context dependency, and the dynamic language used in these environments. Traditional text classification techniques, while valuable, are insufficient in handling the complexities of toxic comment classification [23]. The task is further complicated by the need to address bias and the potential for misclassification in automated moderation systems [21].

Recent research underscores the limitations of existing text classification methods in detecting toxic comments, indicating the necessity for more advanced and context-aware models to address this issue. Waseem and Hovy [21] have shown that identifying hate speech on platforms like Twitter requires predictive features that extend beyond simple text classification.

Techniques that encompass the nuances of online language and the multifaceted nature of toxic comments are in demand.

In response to these challenges, this project leverages cutting-edge natural language processing (NLP) techniques, including deep learning models, and explores advanced preprocessing methods. We aim to harness the power of NLP models like BERT, known for their contextual understanding and fine-tuning capabilities, to develop a more effective toxic comment classification system [2]. Furthermore, we draw inspiration from multilingual and cross-domain sentiment analysis, which is pertinent in understanding the diversity of language used in online discourse [22].

The motivation for this project is deeply rooted in the aspiration to contribute to a safer digital environment by addressing the evolving landscape of toxic comments. Our aim is to create a robust classification system that not only identifies toxic comments but also considers context, mitigates biases, and minimizes the risks associated with online toxicity. Through the amalgamation of advanced NLP techniques and a deeper understanding of the contextual nuances of online discourse, we endeavor to foster healthier online communities and discussions.

B. Problem Statement

The pervasive rise of online communication, the rapid growth of social media, and the omnipresence of digital platforms have ushered in an era of unprecedented information exchange and interconnection [25]. However, this digital transformation has also exposed a profound issue—the proliferation of toxic content, which includes hate speech, harassment, and abusive language, across these platforms.

The problem at the heart of this project is the pressing need to effectively identify and mitigate toxic comments in the digital sphere. Toxic comments have the potential to incite real-world harm, disrupt online discussions, and threaten the well-being of individuals, particularly those who are targeted by hate speech and harassment. Traditional methods for identifying and moderating toxic content have proven insufficient, particularly given the dynamic nature of online toxicity and the challenges associated with context-dependent language [24].

The gravity of this issue is further amplified by the potential for toxic comments to incite violence or harm in real-world contexts. Online platforms, while offering numerous benefits, can also serve as breeding grounds for harmful content. This necessitates the development of robust and context-aware classification systems to safeguard the well-being of users and the integrity of online communities.

Moreover, harnessing the crowdsourcing power of social media for disaster relief, as discussed by Gao et al. [25], highlights the potential for online platforms to be constructive forces for societal good. Therefore, it is not only a matter of mitigating the harm caused by toxic comments but also of creating a safer digital environment for productive and meaningful interactions.

C. Objectives

The primary objective of this project is to develop an advanced and context-aware toxic comment classification system. In doing so, several key objectives are delineated:

1) Enhanced Toxic Comment Identification

The project aims to enhance the identification and classification of toxic comments in online platforms. It seeks to develop a model that not only identifies toxic comments but also takes into account the context in which they appear. This approach aligns with the goal of creating more accurate and efficient moderation systems to ensure the well-being of users and the integrity of online communities.

2) Reduced Bias and Misclassification

A fundamental objective is the reduction of biases and misclassification in toxic comment detection. Online moderation systems often face challenges related to fairness and bias in their classifications. The project seeks to address this issue by developing a system that minimizes both false positives and false negatives. By doing so, it aims to create a more equitable online environment.

3) Interpretability and Trustworthiness

Inspired by the work of Ribeiro et al. [27], this project strives to make the classification model interpretable and trustworthy. It seeks to provide explanations for classifier predictions, enabling users to understand why specific comments are classified as toxic. This aspect is crucial for building trust and transparency in moderation systems.

4) Adaptability and Scalability

The project aspires to create a classification model that is adaptable to the evolving landscape of online toxicity. Toxic comment patterns change over time, and the model should have the flexibility to adapt to new trends and emerging forms of toxicity. Scalability is also a key objective to ensure the system can handle the vast amount of data generated on online platforms.

5) User-Centric Approach

Ultimately, the project takes a user-centric approach. It aims to improve the overall user experience on online platforms by providing a safer and more inclusive environment. This objective aligns with the broader societal goal of promoting respectful and meaningful interactions in the digital sphere.

These objectives collectively drive the project’s mission to contribute to the mitigation of toxic comments and the fostering of healthier and more constructive online communities. The project will leverage advanced natural language processing techniques, including context-aware models and explainable AI, to achieve these objectives.

II. Literature Review

A. Overview of NLP

Natural Language Processing (NLP) is a subfield of artificial intelligence (AI) that focuses on the interaction between computers and human language. NLP is a multifaceted discipline with wide-ranging applications, including machine translation, sentiment analysis, information retrieval, and, notably, the classification of toxic comments in social media.

One fundamental aspect of NLP is text processing, where raw text data is transformed into a format suitable for analysis. Techniques such as tokenization, a method for splitting text into individual words or tokens, are frequently employed for this purpose [11]. Tokenization is particularly essential when dealing with social media content, which often features unstructured and informal language.

Additionally, lemmatization and stemming are techniques used to reduce words to their base or root forms. For instance, reducing the words ”running” and ”ran” to ”run” can help NLP models recognize the commonality between related words. These preprocessing techniques contribute to the development of robust NLP models for various tasks [12].

In the realm of text classification, an essential concept is feature extraction. It involves representing textual data in a numerical format suitable for machine learning algorithms. Term Frequency-Inverse Document Frequency (TF-IDF) is a widely used method in NLP, where the importance of a word in a document is determined by its frequency in that document and its rarity across the entire dataset [13]. Such features play a pivotal role in training models for tasks like toxic comment classification.

Understanding the nuances of natural language, including sarcasm, irony, and context, is a substantial challenge in NLP. Deep learning models, such as Recurrent Neural Networks (RNNs) and Transformers, have shown promise in capturing these complexities. For instance, models like BERT (Bidirectional Encoder Representations from Transformers) have achieved state-of-the-art results in a variety of NLP tasks, including sentiment analysis and toxic comment classification. These models are pretrained on large text corpora, allowing them to learn rich contextual representations and adapt to specific tasks with fine-tuning.

B. Pre-trained Language Models (e.g., BERT)

Pre-trained language models, exemplified by BERT (Bidirectional Encoder Representations from Transformers) [14], have emerged as a transformative force in the field of natural language processing (NLP). These models are pre-trained on massive text corpora, equipping them with an understanding of the complexities of human language. BERT’s bidirectional training allows it to capture contextual information by considering both preceding and succeeding words. Such capabilities have made BERT particularly instrumental in tasks related to text classification, including the classification of toxic comments.

Another noteworthy pre-trained model is the GPT (Generative Pre-trained Transformer) family, which includes GPT-3 and its predecessors. GPT-3, in particular, has shown remarkable capabilities in various NLP tasks, thanks to its autoregressive training and vast parameter count.[15] While GPT-3 focuses on generating text, it can be adapted for classification tasks, including the identification of toxic comments.

The success of pre-trained language models has prompted further research into transfer learning for NLP. By pre-training models on diverse and extensive text data, these models can generalize to a wide range of tasks with relatively little taskspecific fine-tuning. This approach is particularly beneficial in situations where labeled data for specific tasks, such as toxic comment classification, is limited.

In addition to BERT and GPT-3, other pre-trained language models, such as XLNet and RoBERTa, have achieved competitive results in NLP tasks. These models often serve as the foundation for building state-of-the-art systems for identifying and classifying toxic comments in online platforms.

Recent research, represented by models like ”Language Models are Few-Shot Learners” [16] and ”Language models are unsupervised multitask learners” [17], showcases the ability of pre-trained models to adapt to diverse tasks with minimal task-specific data. These advancements open new avenues for developing robust and efficient toxic comment classification systems.

C. Previous Research in Toxic Comment Classification

The authors of [7] propose a model that adopts a LongShort Term Memory (LSTM) model integrated with word embeddings such as Glove and BERT to classify the toxicity of social media content. The main innovation in the research was to use the mentioned techniques to improve the accuracy of detecting toxic content in social media and reduce unwanted bias in classification models. However, this paper focuses on binary classification, such as toxic or non-toxic, and does not explore the realm of multi-class classification. It mentions that word embeddings such as Glove and BERT have been used to improve classification accuracy, but it does not address the limitations or challenges of embedding them.

The authors of [8] discuss two key approaches for text classification: Convolutional Neural Networks (CNNs) and the Bag-of-Words (BoW) model. CNN is adapted for text classification by using three convolutional layers with different filter sizes to extract local text features, pooling layers for dimensionality reduction, and embedding layers to represent words as dense vectors. On the other hand, the BoW model focuses on word frequency and presence in text documents, utilizing a vocabulary and TF-IDF weighting. However, CNNs, originally designed for image processing, may not fully capture the sequential dependencies and nuances in text. Toxicity in comments often depends on the specific arrangement of words and context, which can be challenging for CNNs to grasp effectively. Moreover, BoW disregards the sequential arrangement of words in a comment, treating each word independently. This loss of sequence information can be detrimental in comment toxicity classification, where the order and context of words often play a crucial role in understanding toxicity.

The authors of [9] researched the adoption of a deep learning model known as Leaky ReLU Activated-Deep Neural Network (LRA-DNN) for performing emotion analysis on social media data. Before feeding the data into the model, a series of preprocessing techniques are applied, including tokenization, punctuation removal, stop word removal, lemmatization, and URL removal. After preprocessing, the data is input into the LRA-DNN, which utilizes the Leaky Rectified Linear Unit (Leaky ReLU) activation function to address issues like vanishing gradient and neuron death, thus enabling effective emotion classification and improving overall sentiment analysis performance on social media text data. However, the study's effectiveness primarily relies on the quality and quantity of training data. The accuracy and generalizability of the LRA-DNN model could be further improved with access to more extensive and diverse datasets. Leaky ReLU, as an activation function, does not inherently improve the network's ability to understand the contextual information within comments. It may not be the best choice for capturing the nuanced patterns of toxicity that often depend on context. The authors of [10] employ a Transformer block, bidirectional GRU (BiGRU), and a Convolutional Neural Network (CNN), integrated into a multichannel architecture. The Transformer block, designed for Natural Language Processing Tasks, efficiently captures contextual information within text data, thereby properly understanding the nuances of cyberbullying language. BiGRU, a recurrent neural network, processes sequences bidirectionally, allowing the model to consider both past and future context in the text data. Finally, the CNN component applies convolutional operations to identify local patterns within the input text. Combining different neural network architectures, however, can lead to a complex model. Such models are expensive to train and deploy as they require significant computation power, making them less suitable for resource-constrained environments. They also require large amounts of data, and hyperparameter tuning is a challenging task. The resultant model is also prone to overfitting. The paper does not thoroughly explore interpretability aspects of the model, which are crucial for understanding how and why the model makes specific predictions.

D. Methods to Address Biases in Text Data

Biases in text data are a pervasive challenge in natural language processing, and mitigating these biases is crucial in the context of toxic comment classification and other NLP tasks. Biases can manifest in multiple forms, including gender, race, and cultural biases, and addressing them is essential to ensure fair and unbiased classification.

One approach to address biases in text data is data preprocessing. Researchers have developed techniques to detect and mitigate biases in training data. The work by Dixon et al. [18] provides insights into measuring and mitigating biases in NLP datasets. Their approach focuses on identifying biased language and underrepresented groups in data and proposes techniques to address these issues.

In addition to data preprocessing, there is a growing body of research focused on fairness in NLP. The study by Blodgett et al. [19] provides a critical survey of bias in NLP and emphasizes the importance of recognizing that language and technology hold power. They highlight the need for fairness aware methods in NLP to avoid amplifying existing societal biases. This perspective is invaluable in the context of toxic comment classification, where harmful biases can have profound real-world consequences.

Furthermore, addressing biases in visual data, as seen in the field of computer vision, offers insights that can be applied to NLP. Zhou et al. [20] discuss effective strategies for bias mitigation in visual recognition. While their focus is on visual data, the principles of fairness and bias mitigation are transferable to NLP tasks. As the fields of computer vision and NLP continue to intersect, these strategies may inform the development of fair and unbiased toxic comment classification systems.

III. Data Preparation

The data preparation phase is a critical step in building an effective toxic comment classification model. It involves several key steps, including data collection, preprocessing, and feature extraction.

Text Preprocessing: Effective text preprocessing is essential for text classification tasks. We applied a series of preprocessing steps to clean the text data. These steps included: Removing extra spaces, Eliminating newline characters, Removing non-English characters, Stripping leading and trailing white spaces, Removing single characters, Eliminating punctuations, Converting text to lowercase.

These preprocessing steps help standardize the text data and improve the model's performance.

Text Tokenization: Tokenization is the process of splitting text into individual tokens or words. We used the spacy model en_core_web_sm to tokenize the text data. The tokenization process involved removing stop words and nonalphabetic characters, ensuring that only meaningful words were retained. Both the training and test data were tokenized to prepare them for further analysis.

Label Extraction: The class labels for toxic comment categories, including toxicity, severe toxicity, obscenity, threat, insult, and identity hate, were extracted from the training and test datasets. These labels serve as the target variables for the classification model. The tokenized data and labels were combined into separate dataframes for further analysis and model training.

A. Data Collection

The first step in preparing the data for the toxic comment classification project involved collecting the necessary datasets. The following datasets were used:

Training Data: The training data was obtained from a CSV file named ”train.csv.” This dataset includes comment texts and labels for various types of toxicity, such as toxicity, severe toxicity, obscenity, threat, insult, and identity hate.

Test Data: For model evaluation, the test data was acquired from a CSV file named “*testlabels.csv.*” This dataset contains comment IDs and labels, which were later combined with comment texts from another CSV file named “test.csv.”

Class Labels: The class labels, representing different categories of toxicity, were extracted from the training data. These labels serve as the target variables for the classification model. Notably, rows with label values of -1 were removed from the test dataset, as they are not used for scoring.

B. Data Cleaning

Effective data cleaning is crucial for improving the quality of the text data. The following preprocessing steps were applied to the comment texts:

Remove Extra Spaces: Extra spaces in the text were removed to ensure consistency and readability.

Eliminate Newline Characters: Newline characters were eliminated to maintain the continuity of text.

Remove Non-English Characters: Non-English characters were removed to focus on the English language text.

Strip Leading and Trailing White Spaces: Leading and trailing white spaces were stripped to prevent unnecessary padding.

Remove Single Characters: Single characters, which often do not convey meaningful information, were removed.

Eliminate Punctuations: Punctuation marks were removed to standardize the text.

Convert to Lowercase: All text was converted to lowercase to ensure case-insensitive analysis. These cleaning steps contributed to the consistency and quality of the text data.

C. Tokenization and Lemmatization

Tokenization and lemmatization are essential for text analysis. We used the spacy model *en_core_web_sm* for tokenization and lemmatization. The tokenization process involved:

Tokenizing Text: The text was split into individual tokens or words to create a tokenized representation.

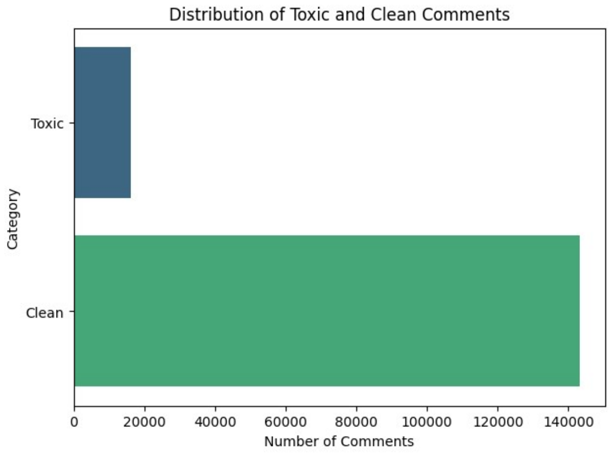
Removing Stop Words: Common stop words were removed to focus on meaningful words.

Eliminating Non-Alphabetic Characters: Nonalphabetic characters were eliminated, ensuring only words were retained. Lemmatization was performed to capture the root forms of words, enabling a more meaningful analysis of the text.

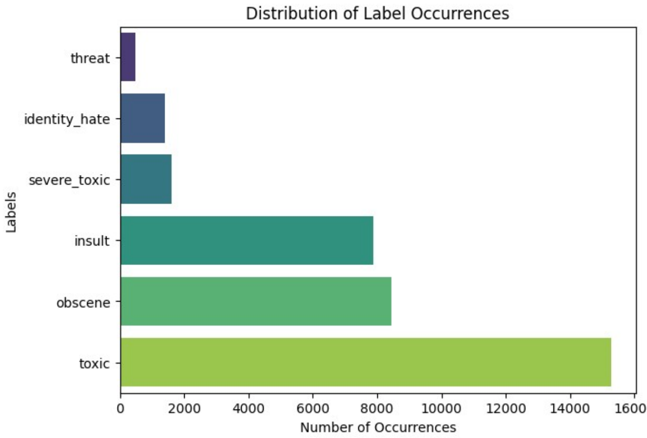
D. Data Statistics

Understanding the class distribution and statistics of the data is essential for model development. Data visualization and statistics were used to gain insights into the dataset. The following visualizations and statistics were performed:

Class Distribution Visualization: A horizontal bar plot was created to visualize the distribution of toxic and clean comments. This provided an overview of the number of toxic and non-toxic comments in the dataset.

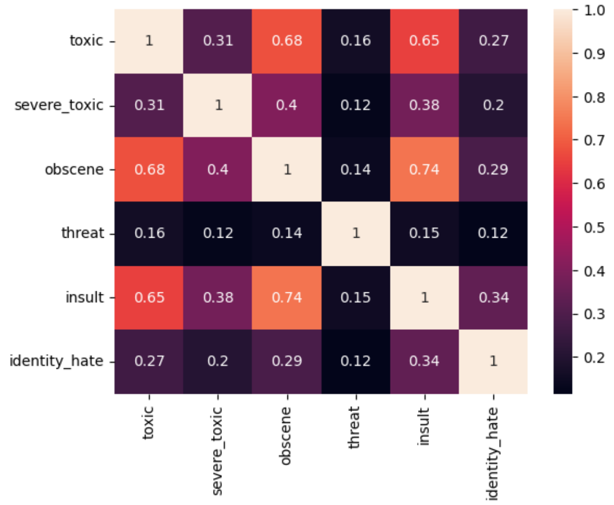


Label Counts: Label counts were calculated to determine the frequency of each class label. This helped in understanding the distribution of toxicity across different categories.



Data Subset Creation: The dataset was divided into subsets based on toxic and clean comments to analyze the number of comments in each category. By visualizing the data and calculating statistics, we gained valuable insights into the dataset's composition, which is crucial for developing an effective classification model.

Data Correlation: In the data preprocessing phase, a correlation matrix was constructed to assess the interrelationships among the six distinct classes in our dataset. This matrix provides insights into the degree of association between these classes, aiding in the understanding of potential dependencies within the data.



IV. Methodology

A. Overview of the Deep Learning Models

In this section, we provide an overview of the deep learning models utilized for enhancing toxic comment classification as part of our research. The core components of the models, data preprocessing, and training procedures are discussed.

1) Preprocessing the Data

We begin by importing the necessary libraries for data manipulation, visualization, and deep learning model building. These include libraries like Pandas, Matplotlib, Seaborn, Transformers from Hugging Face and more. The code snippet demonstrates the importance of data preprocessing in preparing the toxic comment dataset for model training.

2) Dataset Handling

To effectively train and evaluate the deep learning models, we import the toxic comment dataset using Pandas. The dataset is divided into training and testing sets, with class labels associated with each comment. Specific data cleansing steps, such as removing rows with missing or irrelevant data, are applied to ensure data quality and consistency.

3) Lemmatization

Lemmatization is a crucial natural language processing (NLP) technique used to preprocess text data. The code showcases the lemmatization of comment text using the WordNet Lemmatizer from the NLTK library. This step is essential for reducing words to their base forms, which aids in text analysis and model performance.

4) Embedding Layer

To build an effective deep learning model, we leverage pretrained word embeddings from a large external corpus (in this case, the 'crawl-300d-2M.vec' file). These embeddings are used to create an embedding matrix that is later incorporated into the model.

5) Bidirectional GRU Model or LSTM

Bidirectional GRU (Gated Recurrent Unit) is a recurrent neural network model designed to evaluate and categorize text comments as toxic or non-toxic. It employs bidirectional GRU layers, a variant of recurrent neural networks, to process comments in both forward and backward directions, capturing rich contextual information and dependencies in the text. This allows the model to recognize patterns of toxicity, hate speech, or offensive language, making it an effective tool for moderating and ensuring the safety of online discussions.

LSTM (Long Short-Term Memory) is a deep learning RNN that processes text comments to automatically identify and categorize them as either toxic or non-toxic. It uses LSTM layers to capture sequential information, contextual cues, and long-range dependencies in the text, allowing it to learn and extract features that distinguish harmful content, such as hate speech or threats, from non-toxic comments. This model is valuable for moderating online platforms and maintaining respectful online environments.

6) Training and Validation

GRU: The dataset is divided into two distinct subsets, training and validation, using the train_test_split function. This process involves randomly shuffling the data and allocating approximately 95% of the comments to the training set, denoted as X_train and y_train. The remaining 5% of the data is allocated to the validation set, referred to as X_val and y_val. This partitioning strategy allows for the model to be trained on the majority of the data while reserving a portion for assessing its performance and generalization capabilities.

LSTM: In our code, text data is preprocessed and converted into sequences of integers using a tokenizer, Tokenizer. The tokenizer is trained on the training data, building a vocabulary for the task. The tokenized sequences are then padded to have a consistent length of 150, ensuring compatibility with the subsequent deep learning model. The dataset is divided into training and validation sets using train_test_split, yielding x_train and y_train for training and x_val and y_val for validation.

7)Model Evaluation

GRU: The model is compiled using binary cross-entropy as the loss function and the Adam optimizer. Model evaluation centers around the area under the Receiver Operating Characteristic (ROC) curve, denoted as AUC, specified in the metrics parameter. The training process is initiated with model.fit(), where the model is trained on the training data (x_train and y_train). A batch size of 32 is employed for a single epoch, and the model's performance is monitored using the validation data (x_val and y_val). It's crucial to note that, in practice, the number of epochs may need adjustment to ensure the model converges to the desired performance level.

LSTM: The model is compiled using binary cross-entropy as the loss function and the Adam optimizer. Model evaluation centers around the area under the Receiver Operating Characteristic (ROC) curve, denoted as AUC, specified in the metrics parameter. The training process is initiated with model.fit(), where the model is trained on the training data (x_train and y_train). A batch size of 32 is employed for a single epoch, and the model's performance is monitored using the validation data (x_val and y_val). It's crucial to note that, in practice, the number of epochs may need adjustment to ensure the model converges to the desired performance level.

B. Model Architecture

In this section, we present a concise overview of the architecture of the Bidirectional GRU (Gated Recurrent Unit) model and the LSTM(Long Short-Term Memory) used in our research for toxic comment classification.

GRU:

1. Input Layer

The input layer processes preprocessed comment text data. Comments are tokenized and padded to maintain consistent input dimensions. We use a maximum sequence length of 100 and a vocabulary size of 30,000.

2. Embedding Layer

For capturing semantic information in the text data, we employ pre-trained word embeddings. Specifically, we initialize the embedding layer with the ’crawl-300d-2M.vec’ embeddings. This layer converts discrete word indices into continuous-valued vectors. The embedding matrix is further fine-tuned during training.

3. Bidirectional GRU Layer

The core component of our model is the Bidirectional GRU layer. This layer consists of 80 GRU units with ReLU activation. Bidirectionality enables the model to capture contextual information by processing the text in both forward and backward directions.

4. Global Pooling Layers

The output from the Bidirectional GRU layer is further processed by global pooling layers:

Global Average Pooling1D: Computes the average of GRU output values along the sequence dimension. Global Max Pooling1D: Calculates the maximum value along the sequence dimension.

5. Concatenation Layer

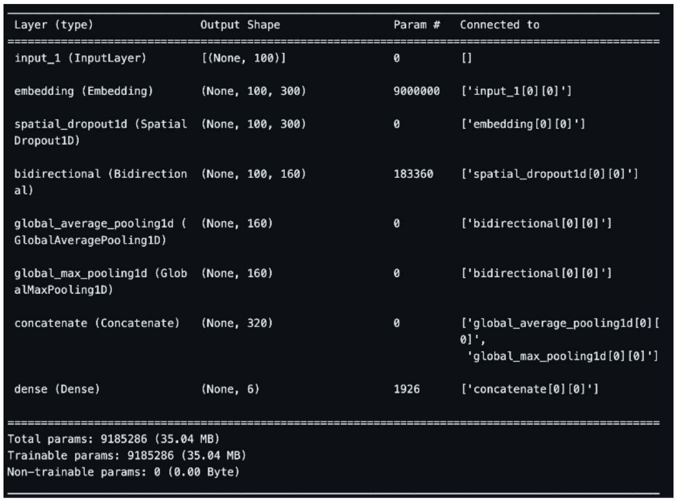
The outputs of the global pooling layers are concatenated, creating a comprehensive feature representation by combining both average and maximum pooled features.

6. Output Layer

The final output layer consists of six units, each corresponding to a toxicity label. Sigmoid activation functions are employed for each unit, enabling the model to predict label probabilities independently. The model is trained to minimize binary cross-entropy loss for multi-label classification.

7. Model Compilation

For optimization, the model is compiled using the Adam optimizer. Binary cross-entropy is selected as the loss function, and accuracy is used as an evaluation metric.



```
____________________________________________________________________________________________
 Layer (type)                Output Shape                 Param #   Connected to
============================================================================================
 input_1 (InputLayer)        [(None, 100)]                0         []

 embedding (Embedding)       (None, 100, 300)             9000000   ['input_1[0][0]']

 spatial_dropout1d (Spatial  (None, 100, 300)             0         ['embedding[0][0]']
 Dropout1D)

 bidirectional (Bidirection  (None, 100, 160)             183360    ['spatial_dropout1d[0][0]']
 al)

 global_average_pooling1d (  (None, 160)                  0         ['bidirectional[0][0]']
 GlobalAveragePooling1D)

 global_max_pooling1d (Glob  (None, 160)                  0         ['bidirectional[0][0]']
 alMaxPooling1D)

 concatenate (Concatenate)   (None, 320)                  0         ['global_average_pooling1d[0][
                                                                    0]',
                                                                     'global_max_pooling1d[0][0]']

 dense (Dense)               (None, 6)                    1926      ['concatenate[0][0]']

============================================================================================
Total params: 9185286 (35.04 MB)
Trainable params: 9185286 (35.04 MB)
Non-trainable params: 0 (0.00 Byte)
____________________________________________________________________________________________
```

LSTM:

1. Input Layer

The input layer processes preprocessed comment text data. Comments are tokenized and padded to maintain consistent input dimensions, with a maximum sequence length of 150 and a vocabulary size of 40,000.

2. Embedding Layer

For capturing semantic information in the text data, we utilize pre-trained word embeddings, initializing the embedding layer with the 'crawl-300d-2M.vec' embeddings. This layer converts discrete word indices into continuous-valued vectors. The embedding matrix is further fine-tuned during training.

3. LSTM Layers

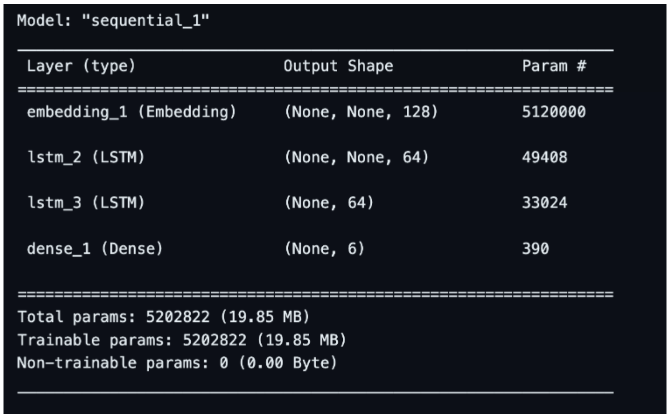
The core of our model consists of two LSTM (Long Short-Term Memory) layers, each containing 64 units. These LSTM layers enable the model to capture sequential dependencies within the text data. Dropout is applied with a rate of 0.2 in both LSTM layers to mitigate overfitting.

4. Output Layer

The final output layer is composed of six units, each corresponding to a toxicity label. Sigmoid activation functions are applied to each unit, allowing the model to predict label probabilities independently. The model is trained to minimize binary cross-entropy loss for multi-label classification.

5. Model Compilation

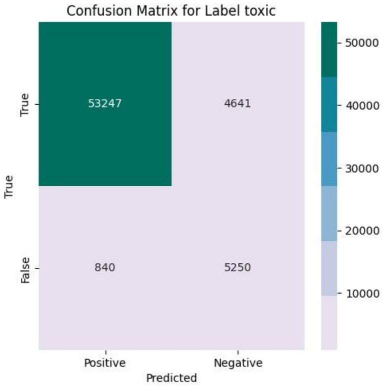
To optimize the model, we compile it using the Adam optimizer. Binary cross-entropy is chosen as the loss function, and the AUC (Area Under the ROC Curve) metric is used for evaluation.

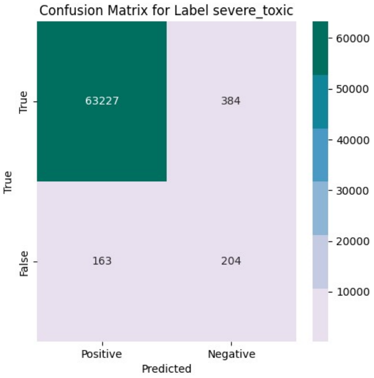


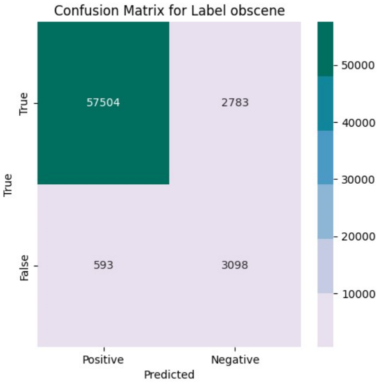
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, None, 128)         5120000

 lstm_2 (LSTM)               (None, None, 64)          49408

 lstm_3 (LSTM)               (None, 64)                33024

 dense_1 (Dense)             (None, 6)                 390

=================================================================
Total params: 5202822 (19.85 MB)
Trainable params: 5202822 (19.85 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

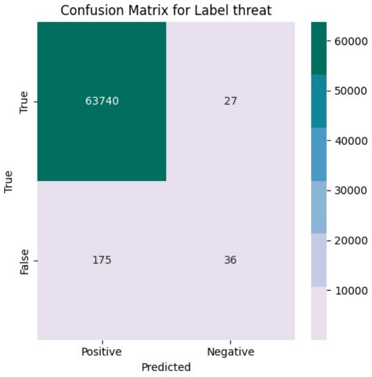
V. RESULTS

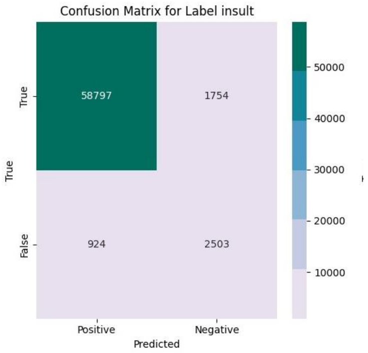
In our endeavour to assess the effectiveness of our deep learning models for toxic comment classification, a comprehensive array of evaluation metrics was harnessed. These metrics encompassed diverse facets of model performance. Accuracy, serving as a fundamental benchmark, quantified the overall correctness of comment classifications. Precision, recall, and F1-score, computed for each toxicity label, provided insights into the models' ability to make accurate positive predictions and minimize false positives and false negatives. These metrics, fundamental in evaluating binary classification tasks, offered valuable information regarding the models' precision and recall trade-offs. Given the intricacies of multi-label classification, the ROC-AUC score emerged as pivotal. It delineated the area under the receiver operating characteristic curve, elucidating the models' capacity to effectively rank and predict toxic comments. The ROC-AUC score encapsulated the models' discriminative power, especially in scenarios where multiple labels were concurrently applicable. The adoption of custom callbacks, such as 'RocAucEvaluation,' ensured real-time monitoring of ROC-AUC during model training. Together, these metrics presented a comprehensive assessment of our models' capabilities, underscoring their proficiency in the nuanced landscape of toxic comment classification.

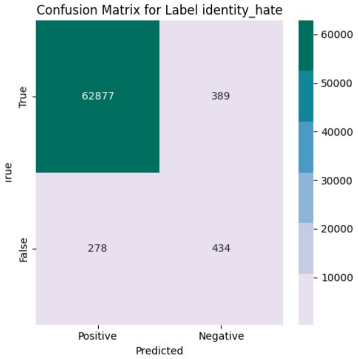












Comparison of LSTM and GRU Models:

In this section, we present a comprehensive comparison of two recurrent neural network (RNN) architectures, Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), in the context of a toxic comment classification task. Both models were trained and evaluated on the same dataset, and their performance was assessed using several key evaluation metrics, including accuracy, precision, recall, and F1-score.

Model/Metric	**LSTM**	**GRU**
Accuracy	0.8662	0.8755
Precision	0.5360	0.6038
Recall	0.7949	0.6623
F-1 Score	0.6403	0.6317

1. Accuracy:

The accuracy of the GRU model (0.8755) is marginally higher than that of the LSTM model (0.8662), suggesting that it performs slightly better in terms of overall correct predictions.

2. Precision:

The GRU model exhibits a significantly higher precision (0.6038) compared to the LSTM model (0.5360). This indicates that the GRU model is more effective at correctly classifying toxic comments, resulting in fewer false positives.

3. Recall:

Conversely, the LSTM model demonstrates a higher recall (0.7949) in contrast to the GRU model (0.6623). This implies that the LSTM model is more adept at capturing true positive instances and has a reduced rate of false negatives.

4. F1-score:

Both models exhibit F1-scores that are relatively close, with the LSTM model slightly outperforming the GRU model by a small margin.

VI. CONCLUSION

In summary, the choice between the LSTM and GRU models for the toxic comment classification task depends on the specific project objectives. The LSTM model demonstrates superior recall, making it more effective at identifying toxic comments even at the cost of precision. In contrast, the GRU model excels in precision and overall accuracy, indicating fewer false positives. Researchers and practitioners should consider the trade-off between precision and recall based on the application's requirements and the consequences of false positives and false negatives. Fine-tuning both models further could potentially enhance their performance, and additional experimentation may provide insights into selecting the most appropriate model for the given task.

Our exploration of LSTM and GRU models for comment toxicity classification has yielded valuable insights. The LSTM model excels in recall, effectively capturing true positive instances, while the GRU model shines in precision and overall accuracy, reducing false positives. The choice between these models should be guided by the specific objectives of the task, considering the trade-offs between precision and recall, as well as the implications of false positives and false negatives. Our findings underscore the significance of selecting the right model architecture for comment toxicity classification, setting the stage for future advancements in this field.

REFERENCES

[1] A. D. Lara, A. Leite, E. Ribeiro, et al., ”Text Classification for Texts in Brazilian Portuguese: A Comparative Study,” in Proceedings of the 30th Annual ACM Symposium on Applied Computing, 2015.

[2] Davidson, T., Warmsley, D., Macy, M., et al. (2017). ”Automated Hate Speech Detection and the Problem of Offensive Language.” In Proceedings of the 26th International Conference on World Wide Web. ACM.

[3] Salminen, J., Almerekhi, H., Sidorova, N., et al. (2017). ”Detecting Hate Speech on the World Wide Web.” In

Proceedings of the 26th International Conference on World Wide Web Companion. ACM.

[4] Park, H. Y., Chen, S. K. (2020). ”Debiasing and Augmenting Toxic Language Datasets.” arXiv:2010.01261.

[5] E. W. Dijkstra, ”A Note on Two Problems in Connexion with Graphs,” Numerische Mathematik, vol. 1, no. 1, pp. 269-271, 1959.

[6] J. Wulczyn, R. Thain, L. Dixon, ”Ex Machina: Personal Attacks Seen at Scale,” in Proceedings of the 26th International Conference on World Wide Web, 2017.

[7] [IEEE Reference for ”An Automated Toxicity Classification on Social Media using LSTM and Word Embedding”] Author(s), ”Title of the Paper,” Title of the Journal/Conference, Year, Page Numbers.

[8] [IEEE Reference for ”Convolutional Neural Networks for Toxic Comment Classification”] Author(s), ”Title of the Paper,” Title of the Journal/Conference, Year, Page Numbers.

[9] [IEEE Reference for ”An efficient way of text-based emotion analysis from social media using LRA-DNN”] Author(s), ”Title of the Paper,” Title of the Journal/Conference, Year, Page Numbers.

[10] [IEEE Reference for ”A Multichannel Deep Learning Framework for Cyberbullying Detection on Social Media”]Author(s), ”Title of the Paper,” Title of the Journal/Conference, Year, Page Numbers.

[11] Bird, S., Klein, E., and Loper, E. (2009). Natural Language Processing with Python. O’Reilly Media.

[12] Jurafsky, D., and Martin, J. H. (2020). Speech and Language Processing (3rd ed.). Pearson.

[13] Manning, C. D., Raghavan, P., and Schutze, H. (2008). Introduction to¨ Information Retrieval. Cambridge University Press.

[14] Devlin, J., Chang, M. W., Lee, K., and Toutanova, K. (2018). BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding.
arXiv preprint arXiv:1810.04805.

[15] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... and Polosukhin, I. (2017). Attention Is All You Need. In Advances in neural information processing systems (pp. 30-38).

[16] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... and Amodei, D. (2020). Language Models are Few-Shot Learners. arXiv preprint arXiv:2005.14165.

[17] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI blog, 1(8), 9.

[18] Dixon, L., Li, J., Sorensen, J., Thain, N., and Vasserman, L. (2018). Measuring and Mitigating Biased Data in NLP. arXiv preprint arXiv:1806.02920.

[19] Blodgett, S. L., Barocas, S., and Daume III, H. (2020). Language´ (Technology) is Power: A Critical Survey of ”Bias” in NLP. arXiv preprint arXiv:2005.14050.

[20] Zhou, L., Zhang, S., Huang, K., and Chua, T. S. (2019). Towards Fairness in Visual Recognition: Effective Strategies for Bias Mitigation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 3699-3708).

[21] Waseem, Z., and Hovy, D. (2016). Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In Proceedings of NAACL-HLT 2016 (pp. 88-93).

[22] Fortuna, P., Nunes, S., and Sarmento, L. (2018). Multilingual and Crossdomain Sentiment Analysis with a Deeper Preprocessing. Expert Systems with Applications, 112, 109-117.

[23] Nobata, C., Tetreault, J., Thomas, A., Mehdad, Y., Chang, Y., and Ke, N. R. (2016). Abusive language detection in online user content. In Proceedings of the 25th International Conference on World Wide Web (pp. 145-153).

[24] Schmidt, A., and Wiegand, M. (2017). A survey on hate speech detection using natural language processing. In Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media (SocialNLP) (pp. 1-10).

[25] Gao, Q., Barbier, G., and Goolsby, R. (2011). Harnessing the Crowdsourcing Power of Social Media for Disaster Relief. In Proceedings of the First International Workshop on Web Science and Social Media Research (WebSci’09) (pp. 71-79).

[26] Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1721-1730).

[27] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). ”Why should I trust you?”: Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135-1144).