**Classification of Toxic Online Comments using Neural Network**

**By**

**Punamkumari Yadav**

Table of Contents

[Introduction 3](#_Toc152855378)

[Related Work 4](#_Toc152855379)

[Dataset 5](#_Toc152855380)

[Dataset Overview: 5](#_Toc152855381)

[Key Attributes and Labels: 5](#_Toc152855382)

[Data Source: https://www.kaggle.com/datasets/julian3833/jigsaw-toxic-comment-classification-challenge (jigsaw-toxic-comment-classification-challenge, n.d.) 6](#_Toc152855383)

[Methods: 6](#_Toc152855384)

[Pre-processing the Data: 6](#_Toc152855385)

[Building the Model: 6](#_Toc152855386)

[Model Compilation: 7](#_Toc152855387)

[Training the Model: 7](#_Toc152855388)

[Visualization 7](#_Toc152855389)

[Evaluation: 7](#_Toc152855390)

[Experimental results: 7](#_Toc152855391)

[Discussion and future work 8](#_Toc152855392)

[Discussion 8](#_Toc152855393)

[Future Work 8](#_Toc152855394)

[Conclusion: 9](#_Toc152855395)

[References 10](#_Toc152855396)

[References 10](#_Toc152855397)

[Appendices 10](#_Toc152855398)

[Appendix 1- Evidence of Implementation: 10](#_Toc152855399)

[Appendix 2- Screenshots and steps: 11](#_Toc152855400)

[Appendix 2- Code 16](#_Toc152855401)

# Introduction

The dynamic development of online communication platforms has brought about a paradigm shift in the interconnectedness of our modern society, fundamentally transforming the way we interact and exchange information in the ever-changing digital realm. But amidst this unparalleled connectivity, a pressing issue has emerged, threatening to undermine the very essence of productive online discourse: toxic comments. Within the fabric of online conversations, a spectrum of harmful expressions, spanning from vitriolic hate speech to damaging harassment, has found fertile ground to spread.

The damaging effects of toxic comments reach far beyond the virtual world, permeating society itself. What were once thought to be spaces for constructive discussions have devolved into breeding grounds for prejudice and hate. The targets of these harmful remarks often endure emotional and psychological turmoil, with some even facing physical dangers. The poisonous nature of these comments corrodes trust within online communities, hindering genuine connections and inhibiting the exchange of ideas.

The impact of toxic statements is significant, leading to serious consequences. Despite being created to promote dialogue, online platforms have transformed into battlefields where toxicity hinders true communication. Rather than fostering inclusivity, the digital realm has become a breeding ground for fear and intimidation. As a result, victims must navigate through a virtual landscape where their voices are drowned out by the overwhelming noise of toxicity, limiting the diversity of perspectives that should enhance online discussions.

Ensuring a healthy and inclusive digital environment means tackling the issue of toxic comments head on. If we let these harmful remarks go unchecked, they can have a detrimental impact, silencing important voices. Not only that, but the ongoing presence of online toxicity presents a significant obstacle for content moderation efforts, forcing platforms to carefully navigate the balance between upholding free speech and safeguarding users from harm.

The pressing issue at hand has sparked a surge of interest from experts and academics, who are leveraging innovative technologies, specifically neural networks, to create effective solutions for identifying and addressing harmful comments. This convergence of cutting-edge tech and social accountability is a pivotal battleground in the constant struggle to promote a digital community that values civility, compassion, and productive communication. As we immerse ourselves in the complexities of developing neural network models for moderating comments, it is essential to recognize the weight of the situation and the immense potential for technology to drive meaningful progress.

In the following pages, we embark on a journey to explore the multifaceted dimensions of toxic comments, dissecting their impact on individuals and society, and examining the role of advanced technologies in mitigating this digital menace. Together, we navigate the delicate balance between fostering open expression and shielding online communities from the detrimental effects of toxicity, seeking to forge a path toward a more enlightened and empathetic digital future.

# Related Work

Many articles on this topic have recently been published. Toxic Comment Classification by Sara Zaheri, Jeff Leath, and David Stroud from Southern Methodist University in 2020 is one of the notable articles. Sara Zaheri (Sara Zaheri) The study looks on the frequency of cyberbullying on social media and its effects on individuals. It also focuses on the creation of a data science solution for identifying and categorizing toxic comments . The study shows that cyberbullying has detrimental consequences, such as impacting the mental health and suicidal thoughts. It addresses the creation of tools to detect toxic comments by firms such as Jigsaw and Google, as well as the use of algorithms like as Nave Bayes, Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) to enhance categorization. The paper's innovations include using elastic cloud computing resources and simplifying the classification process to binary. Compared to Naïve Bayes, LSTM has a true positive rate that is 20% greater, according to the research. The study emphasizes the need for a scalable data science solution and talks about potential directions for future research, including the use of convolutional neural networks (CNN) and support vector clustering (SVC) to expand the classifiers to multi-label classifiers.

Another notable paper is Convolutional Neural Networks for Toxic Comment Classification by Spiros V. Georgakopoulos, Sotiris K. Tasoulis, Aristidis G. Vrahatis, and Vassilis P. Plagianakos from the University of Thessaly Lamia in Greece. The study article explores the application of Convolutional Neural Networks (CNN) for text classification, specifically in the context of harmful comment categorization in online communication. It shows the difficulties of brief text collections as well as the limits of typical word-frequency-based techniques. The research explores the usefulness of CNN for collecting contextual information utilizing character-level features and recurrent CNN models, as well as the use of CNN models for text encoding and incorporating attention mechanisms for increased performance. Furthermore, it underlines the significance of real-time automatic harmful comment recognition and prediction, as well as the limits of existing algorithms. The study includes a full explanation of the CNN-based technique, text preparation, experimental assessment, and implications for future work. The experimental results show that CNN outperforms previous text mining approaches for harmful comment categorization. (Georgakopoulos & Sotiris K. Tasoulis)

The third notable publication comes from the Vellore Institute of Technology's School of Information Technology in Vellore, India, and is written by Pradeep Kumar Roy, Asis Kumar Tripathy, and Tapan Kumar Das. In this research study, a deep convolutional neural network (DCNN) is used to detect hate speech on Twitter. The research starts out by talking about how difficult it is to identify hate speech on Twitter because of the large number of tweets and the contentious nature of hate speech. The authors draw attention to the frequency of hate speech on Twitter in relation to incidents and the requirement for automated techniques to weed out postings that include hate speech. They provide a DCNN system that uses convolution operations to extract the semantics of tweets using tweet text and GloVe embedding vectors. The suggested model outperformed previous models, achieving high values for accuracy, recall, and F1-score. The DCNN model performed better at recognizing hate speech tweets than other deep learning models, according to the authors' comparison of its performance with other deep learning models and conventional machine learning classifiers. The report also addresses the models' shortcomings and makes recommendations for further research, such as adding visuals to texts and expanding the dataset to include hate speech in a variety of languages. The necessity of a broad framework based on deep learning models and the significance of a sizable and varied training dataset are emphasized in the study's conclusion. (PRADEEP KUMAR ROY, 2020)

# Dataset

The growth of internet platforms in recent years has made it easier to have interactions with people around the world, but it has also led to the emergence of bad behaviours like toxic comments that can sabotage constructive dialogues. Researchers trying to create models that can recognize and categorize hazardous comments efficiently will benefit greatly from the Jigsaw Hazardous Comment Classification Challenge, which is hosted on Kaggle. The Conversation AI team has generously shared this dataset, which presents a unique chance to explore the subtleties of online toxicity and improve our knowledge on how to mitigate these problems.

## Dataset Overview:

The data set consists of 159,572 instances with eight attributes: the text content of the comments, the unique identifier (Id) assigned to each comment, and binary labels denoting obscenity, threats, insults, identity hatred, toxicity, and severe toxicity. Nevertheless, we only utilized about 10,000 instances from this dataset because of a lack of computational resources. Analysing and categorizing comments according to how toxic they are is the aim. This extensive dataset makes it possible to fully explore the difficulties involved in classifying comments by providing insight into the numerous facets of harmful online behaviour. This dataset was made available by The Conversation AI team, an Alphabet research project that was started by Jigsaw and Google with the goal of creating tools to improve online discussions. One aspect of their work is creating models that are accessible to the public, such the Perspective API, which tackles the problem of harmful remarks. But the current algorithms are not perfect, and this dataset provides a test ground to further hone and enhance the precision of toxicity classification.

## Key Attributes and Labels:

**Id:** The unique identifier assigned to each comment allows for traceability and ensures the dataset's integrity. Researchers can use this attribute to link specific comments to their corresponding analyses, facilitating a granular examination of the dataset.

**Comment Text:** The heart of the dataset lies in the comment text, providing the raw material for analysis. NLP (Natural Language Processing) techniques can be applied to extract meaningful insights, sentiments, and patterns from this textual data.

**Toxic, Severe Toxic, Obscene, Threat, Insult, Identity Hate:** These binary labels serve as the ground truth for classification tasks. Each label indicates whether a comment exhibits toxicity, severe toxicity, obscenity, threats, insults, or identity hate, offering a multi-dimensional perspective on negative online behaviours.

Creating a dataset to combat online toxicity is no easy feat, as demonstrated by the hard work of the Conversation AI team. Despite the progress made by models such as the Perspective API, there are still obstacles to overcome. One major hurdle is the occurrence of false positives and false negatives, which means that some comments may be incorrectly tagged as toxic or non-toxic. Furthermore, the current models do not allow users to target specific types of toxicity for filtering, which could be problematic as different platforms may have different standards for what is considered negative content.

The researchers conducting this study have implemented an array of analytical methods, such as machine learning algorithms, natural language processing, and deep learning techniques, to investigate this dataset. By utilizing these techniques, their goal is to unravel patterns within the text of comments and improve the accuracy of predicting the different toxicity labels.

Understanding and categorizing toxic comments has significant implications for promoting safety in online spaces. This research has far-reaching applications, beyond just building moderation tools, and can also influence content policies on various platforms. By creating targeted models to identify specific types of toxicity, platforms can customize their content filtering systems to align with their community guidelines and user expectations. Overall, the Jigsaw Toxic Comment Classification Challenge dataset is a crucial resource for researchers and data scientists working towards combating online toxicity. The collaboration between Jigsaw, Google, and Kaggle has resulted in a comprehensive dataset that provides valuable insight into the complexities of toxic comments. As machine learning technology continues to advance, this dataset will undoubtedly play a significant role in furthering our understanding of online toxicity.

## Data Source: <https://www.kaggle.com/datasets/julian3833/jigsaw-toxic-comment-classification-challenge> (jigsaw-toxic-comment-classification-challenge, n.d.)

# Methods:

Our objective is to build a robust multilabel classification model that can accurately identify toxic comments from a given dataset. To do so, we will be using TensorFlow, a widely used deep learning library. To capture the sequential patterns in text data, we will be implementing a Bidirectional Long Short-Term Memory (LSTM) network, which has proven to be highly effective.

## Pre-processing the Data:

A crucial aspect of preparing textual data for training is effective pre-processing. To achieve this, we have established key parameters such as max\_words (representing the maximum number of words that can be tokenized) and max\_len (representing the maximum length of input sequences). Utilizing TensorFlow's Tokenizer class, we can effectively tokenize the text and convert it into integer sequences. To ensure consistency, these sequences are then padded to reach a uniform length. To further optimize the training process, we also split the data into separate training and testing sets.

## Building the Model:

When constructing the toxic comment classification model, we pay meticulous attention to the neural network architecture. The selected architecture harnesses a Bidirectional Long Short-Term Memory (LSTM) network, which is a robust version of the traditional LSTM, recognized for its proficiency in handling sequential data. We use the Sequential model from TensorFlow, a popular deep learning library, to create the architecture.

* **Embedding Layer:** The initial layer of the model is the Embedding layer, acting as the input channel. Its role is integral in transforming words into compact, fixed-sized vectors. Through training, these vectors are custom designed for each word in the dataset. By capturing the semantic connections between words, embeddings enable the model to grasp the nuanced significance of every term used in the toxic comments.
* **Bidirectional LSTM Layer:** At the heart of the model lies the powerful Bidirectional LSTM layer. While traditional LSTMs only consider sequences in chronological order, Bidirectional LSTMs consider both past and future information. This dual-directional approach allows the model to gain a more complete understanding of the context. In the realm of identifying and classifying toxic comments, having a holistic view of a comment's context is crucial in accurately detecting any harmful language and its subtle nuances.
* **Dense Layer with Sigmoid Activation:** After the Bidirectional LSTM layer, we incorporate a Dense layer using a sigmoid activation function to generate the ultimate output of the model. This choice of activation is especially well-suited for handling multilabel classification tasks, as it allows for the possibility that each comment may fall into more than one toxic category at a time. The sigmoid function produces individual probabilities for each toxic category, which are then converted into binary predictions during evaluation using a threshold of 0.5.

## Model Compilation:

Prior to training, we construct our model by specifying the optimizer, loss function, and evaluation metrics. Specifically, we utilize the Adam optimizer and binary cross-entropy loss, which are ideal for multi-label classification tasks. Our chosen evaluation metric is accuracy.

## Training the Model:

Over the course of five epochs and with a batch size of 32, the model undergoes training on pre-processed data. Throughout this training, the model diligently learns to connect input sequences with their appropriate toxicity labels. Furthermore, the addition of validation data allows for careful monitoring of the training process, enabling the model to effectively generalize to new examples.

## Visualization

To assess the model's performance over epochs, we visualize both accuracy and loss on training and validation sets. These plots provide insights into the training dynamics, helping to identify potential overfitting or underfitting issues.

## Evaluation:

After training, we evaluate the model on the test set using metrics such as accuracy. Additionally, we generate a binary prediction for toxicity labels and create a confusion matrix for each category. The confusion matrices are visualized to gain a deeper understanding of the model's classification performance for each toxicity type. To comprehensively assess the model's performance, we generate a classification report, including precision, recall, and F1-score for each toxicity label. Micro-averaged metrics are computed, providing an overall performance summary.

# Experimental results:

In this article, we will critically assess the experimental findings of a multi-label text classification model, providing a comprehensive understanding of its effectiveness on a sample of about 10,000 toxic comments. The dataset was meticulously split into three sets - training, test, and validation - each with a well-balanced distribution of 70%, 30%, and 10% respectively. With the model undergoing five epochs of training, it was able to acquire knowledge and connections within the dataset. This strategic division of the dataset guarantees a thorough evaluation of the model's performance on diverse segments of the dataset.

After careful analysis, the model demonstrates a remarkable accuracy of 91.78%, showcasing its strong capacity to adapt to the entire test dataset. Yet, upon further examination in the classification report, we see how the model fares in identifying six different levels of toxicity: toxic, severe toxic, obscene, threat, insult, and identity hate. Notably, we observe a trade-off between precision and recall, with the model achieving an impressive precision of 83% for toxic comments, signifying its strong ability to accurately predict this category. But The model faces difficulties in several categories, including severe toxic, threat, and identity hate. In these areas, precision, recall, and F1-scores are notably low. For example, precision, recall, and F1-score all equate to zero for the severe toxic category, indicating a considerable challenge. This highlights the need for improvement, as the model struggles to accurately detect and differentiate severe toxic comments.

The compilation of micro-averaged metrics offers a comprehensive evaluation across all categories. Impressively, the micro-averaged precision stands at 79.67%, affirming the model's ability to accurately identify toxic instances on average. However, the micro-averaged recall of 58.29% suggests that the model may overlook a significant number of positive instances. Consequently, the resulting micro-averaged F1-score of 67.32% provides a well-rounded assessment, considering both precision and recall. Along with the detailed classification report, the assessment also includes insights into the model's training progression, presented through accuracy and loss charts plotted over five epochs. These informative visual aids shed light on the model's convergence and potential issues with overfitting or underfitting.

# Discussion and future work

Discussion

The results of the multi-label text classification model's experiments showcase an impressive accuracy rate of 91.78% on the test dataset, solidifying its exceptional performance. By intricately dividing the dataset into training, test, and validation subsets with equal representation, the model underwent a comprehensive assessment of its applicability across various sections of the data. Moreover, with a smooth convergence observed over five epochs, it is evident that the model effectively gained insights and established meaningful connections within the dataset.

Upon further analysis of the classification report, it is evident that there are difficulties in accurately determining the varying levels of toxicity. Although the model has a commendable precision rate of 83% for identifying toxic comments, it faces challenges in distinguishing categories such as severe toxic, threat, and identity hate. The limitations in achieving favourable precision, recall, and F1-scores in these categories point to the necessity for improvement, specifically in accurately identifying and differentiating severe toxic comments.

## Future Work

To improve the results, Future research efforts should prioritize improving the model's ability to identify severe toxic, threat, and identity hate comments by balancing precision and recall. Fine-tuning the model architecture, experimenting with new algorithms, or enhancing the complexity of the model to capture subtle nuances within these categories might help for this.

We can also greatly enhance the effectiveness of our model, by increasing the dataset and incorporating a more diverse range of toxic levels. Many toxic instances, particularly those in the severe toxic, threat, and identity hate categories, are currently lacking in quantity. However, this issue can be addressed through the implementation of data augmentation techniques, which would provide our model with a more robust understanding of these instances.

The model's performance could also be significantly enhanced, by incorporating contextual information, such as surrounding words or phrases in a comment. Context-aware models can also be used as they have the potential to better capture the subtleties of toxic language and improve the distinction between closely related categories, resulting in improved precision and recall. Exploring the effects of adjusting hyperparameters could also lead to enhanced performance of the model. This is evident from the accuracy and loss charts observed during training, where tweaking parameters such as learning rate, batch size, and dropout rates has shown potential in addressing overfitting and underfitting issues.

Exploring the use of ensemble methods, which involve combining predictions from multiple models, could be a valuable approach to harnessing the unique abilities of various architectures. By employing ensemble learning, we may experience improved generalization and resiliency, potentially tackling the specific toxicity categories that challenge our model. As the nature of toxic language constantly evolves, it is essential to continuously monitor the model's performance on new data. Regularly updating and retraining the model with the latest data can help ensure its effectiveness in adapting to the ever-changing patterns of online toxic behaviour.

# Conclusion:

Overall, the results of our experimental evaluation on a sample of 10,000 toxic comments demonstrate the strengths and areas for improvement of our multi-label text classification model. We meticulously divided the dataset into training, test, and validation sets and observed the model's successful convergence over five epochs, indicating its robustness and adaptability. Most notably, the model achieved a commendable accuracy of 91.78% on the test dataset, showcasing its ability to effectively identify toxic language.

Upon thorough examination of the classification report, it is evident that there are complex challenges when it comes to accurately identifying specific levels of toxicity. While the model performs well in predicting toxic comments, there is a noticeable trade-off between precision and recall in categories like severe toxic, threat, and identity hate. This struggle to achieve high precision, recall, and F1-scores in these categories highlights the need for further refinement to effectively detect and distinguish instances of severe toxicity. To address these challenges and improve the model's performance, future work will focus on specific considerations. These include strategies such as enhancing model specificity, augmenting, and balancing the dataset, incorporating contextual information, and fine-tuning hyperparameters. Ultimately, these efforts aim to provide more accurate and comprehensive detections of severe toxicity.

The extensive discussion of micro-averaged metrics, such as precision, recall, and F1-score, adds depth to the evaluation of the model's effectiveness across all categories. Although the micro-averaged precision impressively reaches 79.67%, demonstrating the model's capability to accurately identify toxic instances on average, the corresponding recall of 58.29% highlights the possibility of overlooking positive instances. As a result, the micro-averaged F1-score of 67.32% serves as a well-rounded assessment, considering both precision and recall and indicating scope for enhancement. Moreover, the inclusion of accuracy and loss charts spanning the five training epochs offers valuable insights into the model's convergence and potential issues with overfitting or underfitting. These visual aids enhance the overall comprehension of the model's performance and contribute to a comprehensive evaluation.

Overall, the model demonstrates admirable capabilities in detecting toxic comments. However, there are obstacles within certain categories of toxicity that require continual improvement. To achieve this, the suggested next steps offer a well-planned approach to bolstering the model's precision, recall, and accuracy. This will ultimately result in a more efficient and nuanced multi-label text classifier, better equipped to identify varying degrees of toxicity in online toxic comments.

# References

# References

*Build a Comment Toxicity Model with Deep Learning and Python*. (n.d.). Retrieved from Youtube: https://www.youtube.com/watch?v=ZUqB-luawZg&t=1141s

*CommentToxicity*. (n.d.). Retrieved from Github: https://github.com/nicknochnack/CommentToxicity

Georgakopoulos, S. V., & Sotiris K. Tasoulis, A. G. (n.d.). Convolutional Neural Networks for Toxic Comment Classification. Retrieved from ACM Digital Library: https://dl.acm.org/doi/abs/10.1145/3200947.3208069

*How to compute precision, recall, accuracy and f1-score for the multiclass case with scikit learn?* (n.d.). Retrieved from Stackoverflow: https://stackoverflow.com/questions/31421413/how-to-compute-precision-recall-accuracy-and-f1-score-for-the-multiclass-case

*jigsaw-toxic-comment-classification-challenge*. (n.d.). Retrieved from Kaggle: https://www.kaggle.com/datasets/julian3833/jigsaw-toxic-comment-classification-challenge

*Keras - Plot training, validation and test set accuracy*. (n.d.). Retrieved from stackoverflow.com: https://stackoverflow.com/questions/41908379/keras-plot-training-validation-and-test-set-accuracy

PRADEEP KUMAR ROY, A. K.-Z. (2020). A Framework for Hate Speech Detection Using Deep Convolutional Neural Network. *IEEE*.

Sara Zaheri, J. L. (n.d.). https://scholar.smu.edu/. Retrieved from SMU Scholar: https://scholar.smu.edu/cgi/viewcontent.cgi?article=1134&context=datasciencereview

# Appendices

## **Appendix 1- Evidence of Implementation:**

This project was developed Google Collab since it is one of the easiest IDEs to use and has many features for constructing Machine Learning Models.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## **Appendix 2- Screenshots and steps:**

1. **Importing and exploring the data:**

**A screenshot of a computer

Description automatically generated**

****

1. **Preprocessing:**

****

1. **Building the Model:**

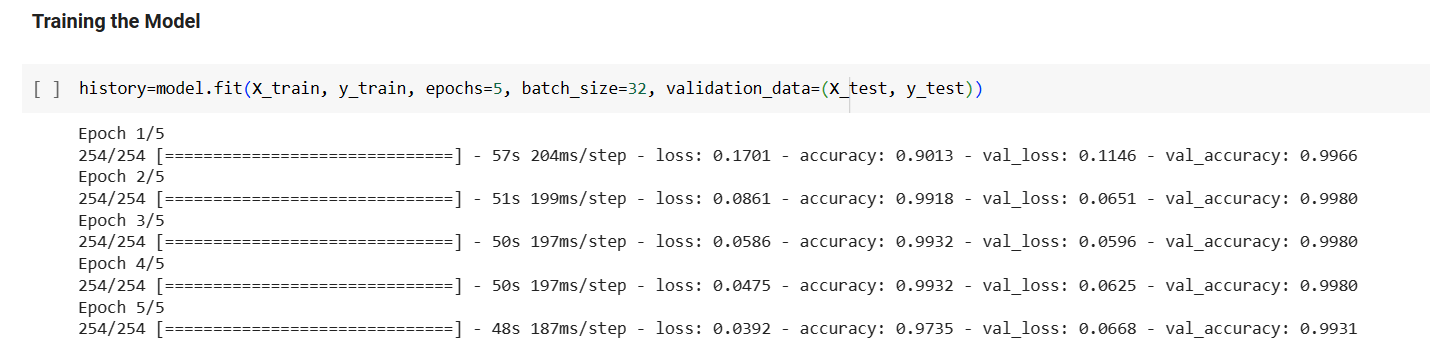
**A screenshot of a computer

Description automatically generated**

**A close-up of a white paper

Description automatically generated**

1. **Training the Model:**

****

1. **Evaluating the Model:**

**A white background with colorful text

Description automatically generated with medium confidence**

**A graph of a graph

Description automatically generated with medium confidence**

**A close-up of a person

Description automatically generated**

**A graph of loss of a model

Description automatically generated with medium confidence**

**A white box with black text

Description automatically generated**

**A computer code with text

Description automatically generated with medium confidence**

**A screenshot of a graph

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A computer code with many colored text

Description automatically generated with medium confidence**

## Appendix 2- Code

!pip install tensorflow numpy pandas

"""\*\*Importing the Dataset\*\*"""

import pandas as pd

from google.colab import drive

import pandas as pd

drive.mount('/content/gdrive')

file = '/content/gdrive/MyDrive/ToxicComments.csv'

data = pd.read\_csv(file)

data.head()

data.info()

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Embedding, LSTM, Dense, Bidirectional

from tensorflow.keras.preprocessing.text import Tokenizer

from tensorflow.keras.preprocessing.sequence import pad\_sequences

from sklearn.model\_selection import train\_test\_split

import numpy as np

"""\*\*Pre-processing Data\*\*"""

max\_words = 10000

max\_len = 100

tokenizer = Tokenizer(num\_words=max\_words)

tokenizer.fit\_on\_texts(data['comment\_text'])

sequences = tokenizer.texts\_to\_sequences(data['comment\_text'])

X = pad\_sequences(sequences, maxlen=max\_len)

y = data[['toxic', 'severe\_toxic', 'obscene', 'threat', 'insult', 'identity\_hate']].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

"""\*\*Building the Model\*\*"""

model = Sequential()

model.add(Embedding(max\_words, 128, input\_length=max\_len))

model.add(Bidirectional(LSTM(64)))

model.add(Dense(6, activation='sigmoid'))

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

model.summary()

""" \*\*Training the Model\*\*"""

history=model.fit(X\_train, y\_train, epochs=5, batch\_size=32, validation\_data=(X\_test, y\_test))

from sklearn.metrics import multilabel\_confusion\_matrix, classification\_report, accuracy\_score, precision\_score, recall\_score, f1\_score

import matplotlib.pyplot as plt

import seaborn as sns

"""\*\*Evaluating the Model\*\*"""

plt.figure(figsize=(8, 5))

plt.plot(history.history['accuracy'])

plt.plot(history.history['val\_accuracy'])

plt.title('Model Accuracy')

plt.ylabel('Accuracy')

plt.xlabel('Epoch')

plt.legend(['Train', 'Test'], loc='upper left')

plt.show()

plt.figure(figsize=(8, 5))

plt.plot(history.history['loss'])

plt.plot(history.history['val\_loss'])

plt.title('Model loss')

plt.ylabel('Loss')

plt.xlabel('Epoch')

plt.legend(['Train', 'Test'], loc='upper left')

plt.show()

"""\*\*Evaluating the Model using Test Set\*\*

"""

loss, accuracy = model.evaluate(X\_test, y\_test)

print(f'Test Accuracy: {accuracy \* 100:.2f}%')

y\_pred = model.predict(X\_test)

y\_pred\_binary = (y\_pred > 0.5).astype(int)

conf\_matrix = multilabel\_confusion\_matrix(y\_test, y\_pred\_binary)

labels = ['toxic', 'severe\_toxic', 'obscene', 'threat', 'insult', 'identity\_hate']

fig, axes = plt.subplots(2, 3, figsize=(7,5))

for i, ax in enumerate(axes.flatten()):

    sns.heatmap(conf\_matrix[i], annot=True, fmt='d', cmap='Blues', cbar=False, ax=ax)

    ax.set\_title(f'Confusion Matrix - {labels[i]}')

    ax.set\_xticklabels(['Not Predicted', 'Predicted'])

    ax.set\_yticklabels(['Not Actual', 'Actual'])

plt.tight\_layout()

plt.show()

print("Classification Report:\n", classification\_report(y\_test, y\_pred\_binary, target\_names=labels))

precision = precision\_score(y\_test, y\_pred\_binary, average='micro')

recall = recall\_score(y\_test, y\_pred\_binary, average='micro')

f1 = f1\_score(y\_test, y\_pred\_binary, average='micro')

accuracy\_micro = accuracy\_score(y\_test, y\_pred\_binary)

print(f'Micro-Averaged Metrics:')

print(f'Precision: {precision:.4f}')

print(f'Recall: {recall:.4f}')

print(f'F1-score: {f1:.4f}')

print(f'Accuracy: {accuracy\_micro:.4f}')