North East University Bangladesh

Department of Computer Science and Engineering

Toxic Comment Classification from Social Media with DL

By

Sourov Dey
Reg. No: 200103020026
BSc(Engg) in CSE
4th year 2nd semester

Mitu Paul Bibu
Reg. No: 200103020044
BSc(Engg) in CSE
4th year 2nd semester

Supervised By

Razorshi Prozzwal Talukder
Lecturer
Department of Computer Science and Engineering

29th November, 2023

Abstract

Online conversations can be toxic and subjected to threats, abuse, or harassment. Online conversation toxicity can be defined as rude, disrespectful, make somebody leave a discussion, stop expressing oneself, and even give up on looking for different opinions. To help improving online conversation and analyze the negative online behaviors, the goal of this project is to build a deep learning model that can perform binary classification, determine toxic and non-toxic comment.

Table of Contents

Chapter 1

INTRODUCTION

The dataset is collected by the Conversation AI team, a research initiative founded by Jigsaw and Google organized a Kaggle competition in 2019: Jigsaw Unintended Bias in Toxicity Classification .The dataset was collected from Civil Comments platform from 2015 to 2017. Online discussions can turn harmful with rude behavior, threats, and abuse. Toxicity involves being disrespectful, driving people away, silencing voices, and discouraging diverse opinions. This project aims to create a model using deep learning. It will classify comments into two categories: toxic and non-toxic from a social network, making online spaces safer. Develop a toxic comment classification system using a combination of traditional machine learning model and deep learning method (LSTM).

1.1 Dataset Information

The dataset contains 1.8 million observations and 45 features.The “comment_text” column contains the comment from diverse range of conversations and “target” column indicates how toxic a comment is and target value ≥ 0.5 means comment is toxic. Additional toxicity subtype attributes are severe toxicity, obscene, identity attack, insult, threat, and sexually explicit. Reaction variables are funny, wow, sad, likes, disagree. Identity variables can be classified into five categories.
Gender: male, female, transgender, other gender Sex: heterosexual, homosexual gay or lesbian, bisexual, other sexual orientation.
Religion: Christian, Jewish, Muslim, Hindu, Buddhist, atheist, other religion Race: black, white, Asian, Latino, and other race or ethnicity.
Disability: physical disability, intellectual or learning disability, psychiatric or mental illness, other disability. Toxicity labels were obtained from the human raters. Each comment was rated by 10 to thousand raters. Raters marked each comment as very toxic, toxic, hard to say or no toxic. They also tried to guess the gender targeted in the comment etc. Note: We will work on only two features “comment_text” & “target”.
Here are some examples of the dataset.

Comment: “haha you guys are a bunch of losers.”
Target Label (Toxicity): 0.89
Comment: “I love the idea of upvoting entire articles.”
Target Label (Toxicity): 0.0

1.2 Exploratory Data Analysis

Fig 01 represents the distribution of the target variable which has two categories such as non-toxic (0) and toxic (1). Target variable < 0.5 belongs to non-toxic and target variable ≥ 0.5 belongs to toxic comments. From fig 01, we can see that the dataset is highly imbalanced. Among 1.8M observations, around 92% (1.66M) of the data belongs to non-toxic and only around 8% (0.14M) of the data belongs to toxic comments.

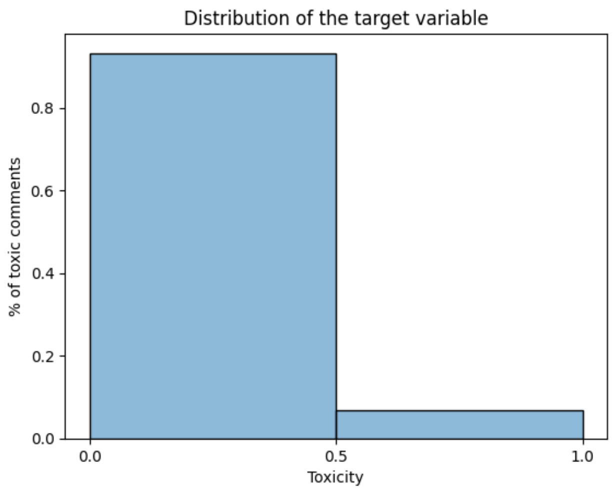


Fig 01

Fig 01. Distribution of the target variable. Target variable < 0.5 belongs to non-toxic (0) and target variable ≥ 0.5 belongs to toxic (1) comments.

Fig 02. Percent of toxic comments related to different identities using target and population amount of each identity as weights.

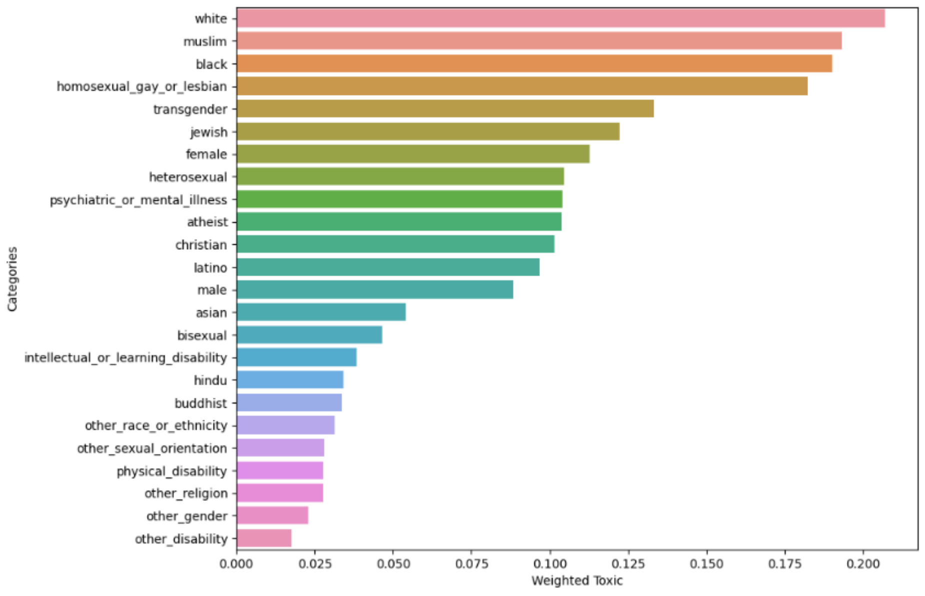


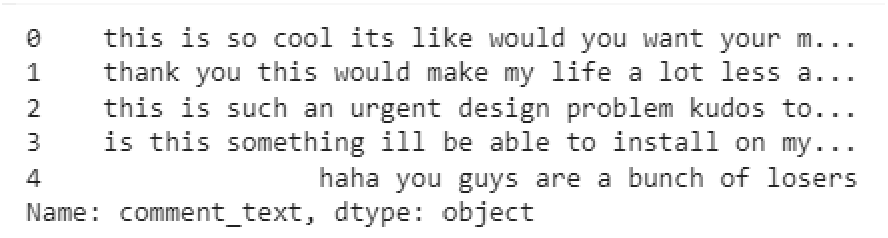
Fig 02

From fig 02, among these 24 identity variables, the most targeted identities are white, black, homosexual gay or lesbian, Muslim, Jewish, female etc. Here, for each observation we have a value of target variable which represents how toxic the comment is. Each identity variable has also a value between 0 to 1 to identify how much they have been targeted. Using these two aspects, we can find which identities are more frequently related to toxic comments. To determine the weighted toxicity, first calculate the sum of the product of each identity variable with the target variable. Then, find the ratio of this sum of product to the number of identity variables which are greater than zero.

1.3 Data Preprocessing

Data Cleaning: In general, we use a lot of punctuations and other words without any contextual meaning. Comments often consists of other user mentions, hyperlink texts, emoticons and punctuations. The Comments are checked for duplicates while pre-processing and in case if there is any comment that is repeating more than once then duplicate comments are removed.

- Firstly, the whole text is converted into lowercase for convenience.
- Removal of hashtag symbols (#), user mentions (@user).
- Removal of hyperlinks or any HTML elements
- Removing numbers
- Removing punctuations



```
0    this is so cool its like would you want your m...
1    thank you this would make my life a lot less a...
2    this is such an urgent design problem kudos to...
3    is this something ill be able to install on my...
4                    haha you guys are a bunch of losers
Name: comment_text, dtype: object
```

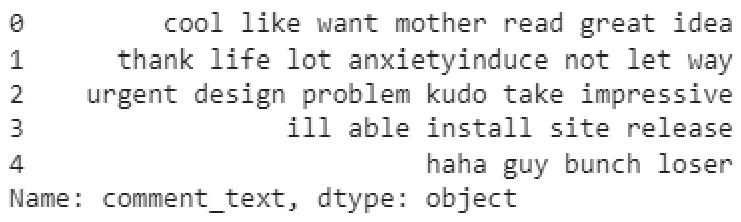
Fig 03 After Cleaning Data

Tokenization: Natural language with which people communicate usually contain, commonly repeated words such as ‘a’, ‘an’, ‘the’, which are commonly referred to as stop words. These words need to be filtered out before going for anymore further processing of the text since they do not greatly add any meaning of the sentence.

Next, lemmatization is applied on the text which determines the root word which belongs to the same language unlike stemming which only performs word reduction. Since lemmatization is typically more informative, it is opted over stemming. When lemmatization is applied to the words (‘eat’, ‘ate’, ‘eaten’), it reduces the words to a common lemma which is ‘eat’.

- Removal of Stop words

➢ Lemmatization of text



```
0          cool like want mother read great idea
1       thank life lot anxietyinduce not let way
2     urgent design problem kudo take impressive
3                    ill able install site release
4                            haha guy bunch loser
Name: comment_text, dtype: object
```

Fig 03 After Tokenization

TFIDF Vectorizer: Term Frequency - Inverse Document Frequency (TF-IDF) is a widely used statistical method in natural language processing and information retrieval. It measures how important a term is within a document relative to a collection of documents (relative to a corpus). Words within a text document are transformed into importance numbers by a text vectorization process. There are many different text vectorization scoring schemes, with TF-IDF being one of the most common.

As its name implies, TF-IDF vectorizes/scores a word by multiplying the word’s Term Frequency (TF) with the Inverse Document Frequency (IDF).

Term Frequency: TF of a term or word is the number of times the term appears in a document compared to the total number of words in the document.

TF= number of times the term appears in a document/ total number of words in the document.

Inverse Document Frequency: IDF of a term reflects the proportion of documents in the corpus that contain the term. Words unique to a small percentage of documents (e.g., technical jargon terms) receive higher importance values than words common across all documents (a, the, and).

IDF= log(number of the documents in the corpus/ number of documents in the corpus contain the term)

The TF-IDF of a term is calculated by multiplying TF and IDF scores.

TF-IDF= TF*IDF

Chapter 2

PROPOSED METHOD

2.1 Logistic Regression

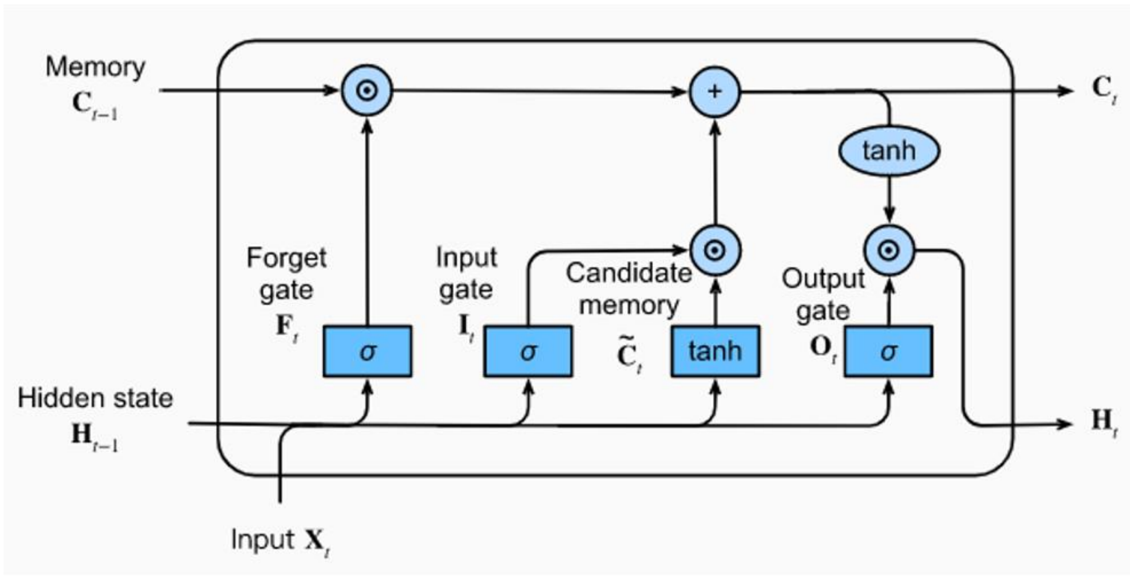
In the Logistic Regression step, we use a mathematical model to analyze and predict outcomes, particularly for binary classification problems like ours (toxic or non-toxic comments). The model examines the relationships between the input features (our transformed text data) and the likelihood of a comment being toxic. It assigns probabilities and classifies comments based on whether they are more likely to be toxic or non-toxic.

2.2 LSTM

In the LSTM step, we leverage a type of deep learning model called Long Short-Term Memory networks. These networks are excellent for understanding and predicting sequences, like the order of words in our text data. LSTM helps the model capture long-term dependencies in language, making it effective for tasks like classifying toxic comments. It allows the model to learn patterns and relationships in the data over extended periods, enhancing its ability to make accurate predictions.

Key components of an LSTM cell include:

1. Cell State (Ct): This represents the long-term memory of the network. It runs straight down the entire chain of LSTM units and is carefully regulated by gates to add or remove information.
2. Hidden State (ht): This is the output of the LSTM unit at a specific time step. It is a filtered version of the cell state that only exposes relevant information.
3. Forget Gate: Decides what information to discard from the cell state.
4. Input Gate: Determines what new information to store in the cell state.
5. Output Gate: Controls the exposure of the information in the cell state, producing the output based on the current input and the memory.



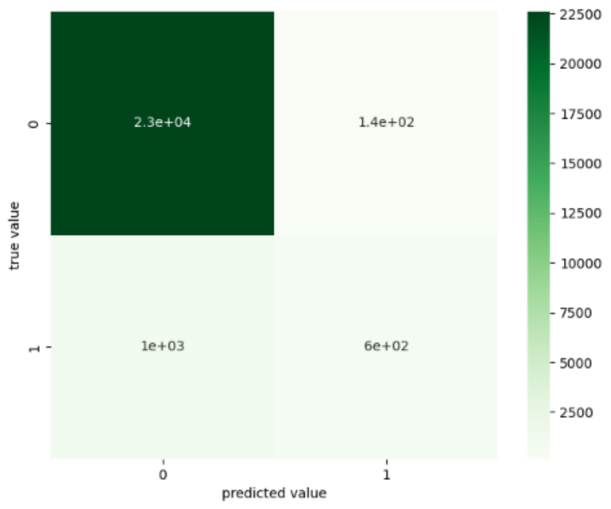
Memory $\mathbf{C}_{t-1}$
$\mathbf{C}_t$
tanh
Forget gate $\mathbf{F}_t$
Input gate $\mathbf{I}_t$
Candidate memory $\tilde{\mathbf{C}}_t$
Output gate $\mathbf{O}_t$
σ
σ
tanh
σ
Hidden state $\mathbf{H}_{t-1}$
$\mathbf{H}_t$
Input $\mathbf{X}_t$

Chapter 3

RESULTS AND DISCUSSION

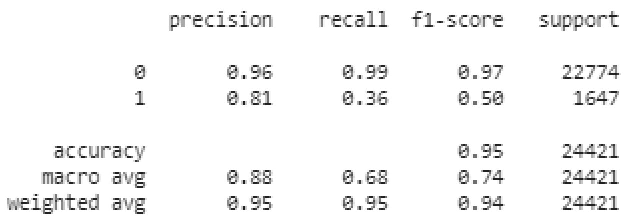
3.1 Logistic Regression

For logistic regression out of 81404 data we have used 70% data for training and 30% data for testing.



True Positives (TP): 599 instances were correctly predicted as positive cases. False Negatives (FN): 1048 positive cases were incorrectly predicted as negative. False Positives (FP): 144 negative cases were incorrectly predicted as positive. True Negatives (TN): 22630 instances were correctly predicted as negative cases. This breakdown highlights the performance of a binary classification model: Positive Cases: The model correctly identified 599 instances as positive (True Positives). However, it failed to identify 1048 positive instances (False Negatives). Negative Cases: The model accurately predicted 22630 instances as negative (True Negatives). It misclassified 144 negative instances as positive (False Positives). The model performs well in identifying negative cases but has higher errors in predicting positive instances, especially in false negatives.

Classification Report:



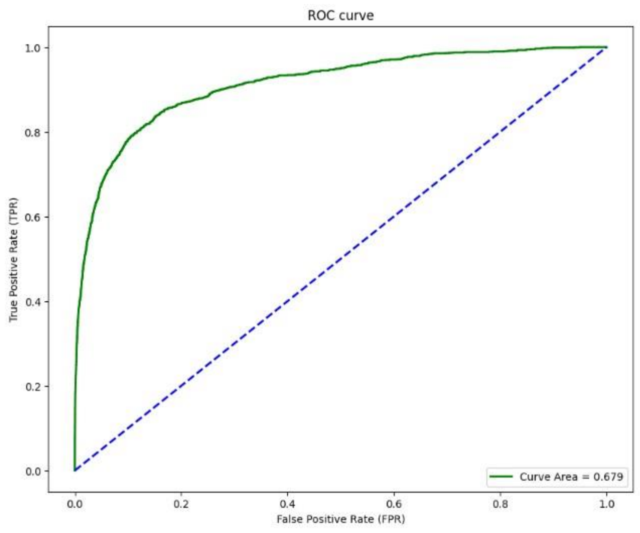
```
              precision    recall  f1-score   support

           0       0.96      0.99      0.97     22774
           1       0.81      0.36      0.50      1647

    accuracy                           0.95     24421
   macro avg       0.88      0.68      0.74     24421
weighted avg       0.95      0.95      0.94     24421
```

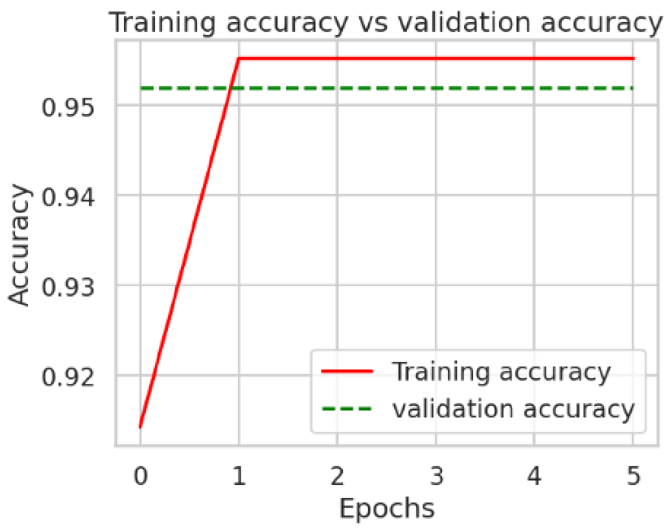
We can see high precision and recall for negative instances (class 0) but lower recall and F1-score for positive instances (class 1). This indicates that the model is performing well in identifying negative cases but has limitations in correctly identifying positive cases.

ROC CURVE:



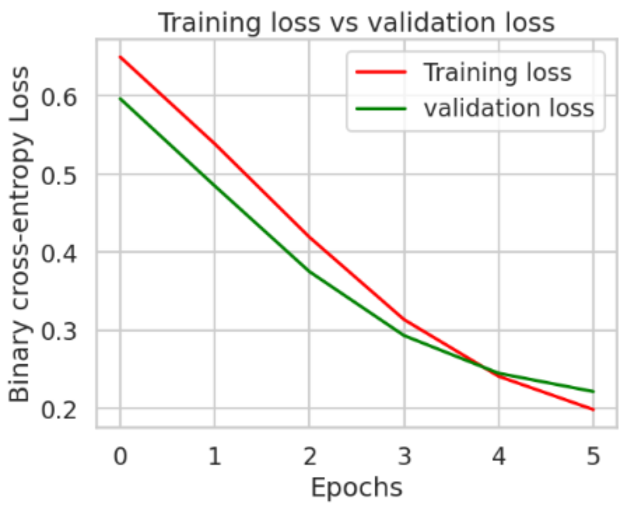
3.2 LSTM

Training Accuracy VS Validation Accuracy:



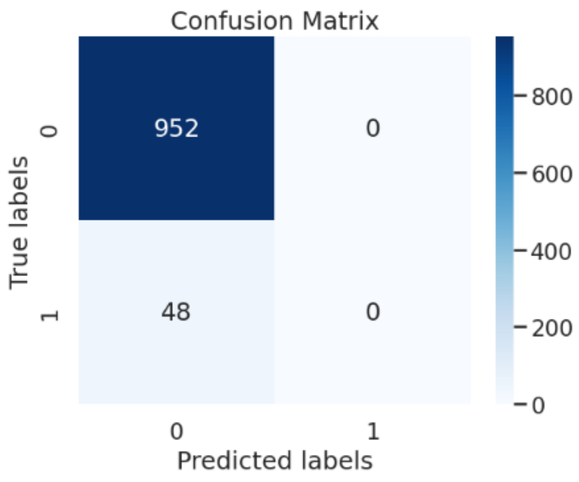
We can see training and validation accuracy both are above 95% .Validation accuracy remain same for all the epochs and training accuracy jumped a little bit after first epoch.

Training loss VS Validation loss:



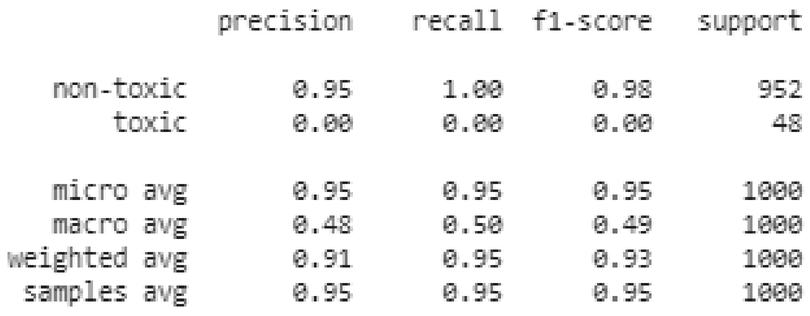
We can clearly see that loss is training and validation loss are reducing over epochs. Here the loss function is binary cross entropy loss.

Confusion Matrix:



True Positives (TP): 952 instances were correctly predicted as belonging to class 0. False Negatives (FN): 0 instances of class 0 were incorrectly predicted as class 1. False Positives (FP): 48 instances of class 1 were incorrectly predicted as class 0. True Negatives (TN): 0 instances were correctly predicted as belonging to class 1. Class 0: The model correctly identified a large number of instances (952) belonging to class 0. There were no instances of class 0 incorrectly classified as class 1 (false positives). Class 1: The model incorrectly predicted 48 instances of class 1 as class 0 (false positives). However, it did not correctly predict any instances as class 1 (true positives) in this specific scenario. While the model has a high number of true positives for class 0, it fails to correctly predict any instances of class 1, resulting in a notable false positive rate for class 1.

Classification Report:



```
               precision    recall  f1-score   support

    non-toxic       0.95      1.00      0.98       952
        toxic       0.00      0.00      0.00        48

    micro avg       0.95      0.95      0.95      1000
    macro avg       0.48      0.50      0.49      1000
 weighted avg       0.91      0.95      0.93      1000
  samples avg       0.95      0.95      0.95      1000
```

The model performs exceptionally well in identifying non-toxic instances (Class 0), achieving high precision, recall, and F1-score. However, it completely fails to identify instances of the toxic class (Class 1), resulting in extremely low precision, recall, and F1-score for this class.

Chapter 4

LACKING & LIMITATION

Data Size and Memory Issues:

- Problem: Limited dataset size and memory constraints.
- Impact: Couldn't use a larger dataset or comprehensive text representation, limiting the model's ability to capture nuanced patterns.

Imbalanced Target Levels:

- Problem: Significant class imbalance, especially between non-toxic and toxic instances.
- Impact: Challenges in training, biased learning patterns, and difficulty in accurately predicting instances of the minority class.

Handling Large-Scale Embedding:

- Problem: Memory constraints prevented the use of sophisticated embedding layers.
- Impact: Hindered the model's capacity to capture intricate semantic relationships and nuances in the text corpus.

Chapter 5
CONCLUSION

Online discussions can turn harmful with rude behavior, threats, and abuse. Toxicity involves being disrespectful, driving people away, silencing voices, and discouraging diverse opinions.

While the logistic regression model demonstrated commendable performance within its limitations, the LSTM model's potential was interrupted due to constraints in utilizing large-scale embedding.

References

1. **Introduction to Long Short-Term Memory (LSTM):** https://medium.com/analytics-vidhya/introduction-to-long-short-term-memory-lstm-a8052cd0d4cd
2. **Essential Text Pre-processing Techniques for NLP!:** https://www.analyticsvidhya.com/blog/2021/09/essential-text-pre-processing-techniques-for-nlp/
3. **Let's learn about AUC ROC Curve!:** https://medium.com/greyatom/lets-learn-about-auc-roc-curve-4a94b4d88152