# TOXIC COMMENT FILTERING USING NEURAL NETWORKS

Manya Sahay
*Mukesh Patel School of Technology Management and Engineering*

Lehar Rathore
*Mukesh Patel School of Technology Management and Engineering*

Aryaan Peshoton
*Mukesh Patel School of Technology Management and Engineering*

Dr. Ami Munshi
*Mukesh Patel School of Technology Management and Engineering*

Dr. Vaishali Kulkarni
*Mukesh Patel School of Technology Management and Engineering*

**Abstract - In today's world of social media, rather than just being a platform which facilitates communication, it also becomes a medium for some people to spread hateful comments about other groups or individuals which may cause harm or mental strain to them. To filter out these comments so they can be banned or have action taken against them, the model in this paper filters out such toxic comments using the concepts of deep learning and Natural Language Processing. Our neural network model which can classify a comment as toxic/non-toxic has an accuracy of 94.64% on the test data.**

## I. INTRODUCTION

The internet is an essential part of modern society. It provides abundant resources for almost all types of subject matter which aids in self learning and spreading new information.It also serves as a platform for people of various backgrounds sitting in completely different places to communicate with each other and share ideas.Despite being made for a good cause, some groups or individuals make use of this platform to make hateful comments towards others. The prevalence of these toxic comments creates an unhealthy environment and makes people hesitant to share things online. These toxic comments include - harmful, offensive, or disruptive to a particular community or individual.

They can take many forms, such as hate speech, personal attacks, and bullying. Toxic remarks on the internet should be avoided since they have the potential to negatively affect both people and communities. Toxic remarks, for instance, can cause anxiety, despair, it can also cause them to lose confidence. This would be very common in young people. Their mental health and

general wellbeing may suffer permanently as a result of this. Toxic remarks can occasionally even result in self-harm or suicide. Furthermore, poisonous remarks have the potential to make communities unfriendly and unhealthy. They have the potential to alienate users, inhibit originality and creativity, and degrade a platform's overall quality. This may have a very serious consequence since it would reduce user pleasure and social media engagement

So, in this paper, we have attempted to make a Deep Learning model which can filter out such comments using Sentiment Analysis.

## II. LITERATURE REVIEW

Toxic comment filtering has evolved over time in response to the rising need to keep online communities safe and courteous.

Moderation's Emergence (Late 1990s - Early 2000s):
As online communities started to rise and increased in numbers, various platforms adopted rudimentary moderation mechanisms. To identify and delete inappropriate content, moderators usually relied on user reports. There were no automatic filtering methods in place due to lack of advanced technology, and so moderating was mostly done manually.

Rising Social Media (Mid 2000s):
With the introduction of online platforms and social media sites such as Facebook, Twitter, and YouTube, new issues in dealing with acrimonious remarks arose. Automated content screening systems and streaming platforms like twitch began to arise, although they were crude and sometimes useless.

Machine Learning and Natural Language Processing (Late 2000s - Early 2010s):
Content filtering has improved because of improvements in machine learning and natural language processing (NLP). Platforms began utilizing algorithms based on keywords and patterns to detect and filter hazardous remarks. Due to the dynamic nature of toxic language and it resulting in a false positive or a false negative there were still some challenges left to conquer.

Ongoing Progress (2020s and Beyond):
With breakthroughs in AI and NLP, toxic comment screening continues to evolve. The emphasis continues on enhancing filtering accuracy while resolving ethical and prejudice problems. To summarize, toxic commentfiltering has evolved from a near-nil practice to a hybrid of automatic

algorithms, human moderation, and user customisation.

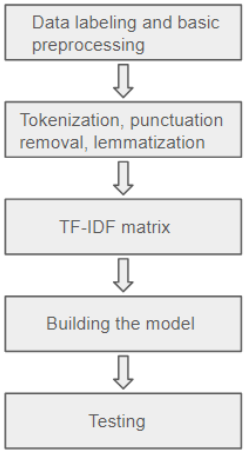
Some of the main reasonswhy NLP approaches are so important in this context:

Machine Translation:
NLP approaches fuel machine translation systems like Language detection and Language Translation systems such as Google Translate, allowing language boundaries to be bridged and worldwide communication to be facilitated.

Sentiment Analysis:
Natural Language Processing (NLP) is used to assess the emotional tone, sentiment, or opinion expressed in text, which can be used to classify a statement or a paragraph as positive or negative. Understanding public image and consumer feedback is useful for businesses and organizations.
Search Engines:
NLP is used by search engines such as Google to comprehend user queries and offer appropriate search results. This enhances the search experience for the user.

## III. FLOW OF THE PROJECT

The diagram below shows a simple flow of how the project will work.




i. Flow of the paper

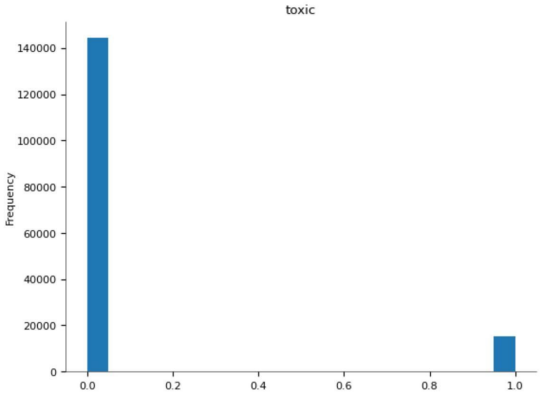
First stage is exploring the dataset and doing some basic preprocessing on it so that it is easier to deal with. After the basic steps tokenization, stop word removal, punctuation removal and lemmatization has been performed. TF-IDF has been used for feature extraction and then a Neural Network model is trained on this data. Finally, sample sentences have been put into the model to test it and the output has been recorded.
All of these steps have been explained in detail in the next section.

## IV. METHODOLOGY

### A. Data source and labeling

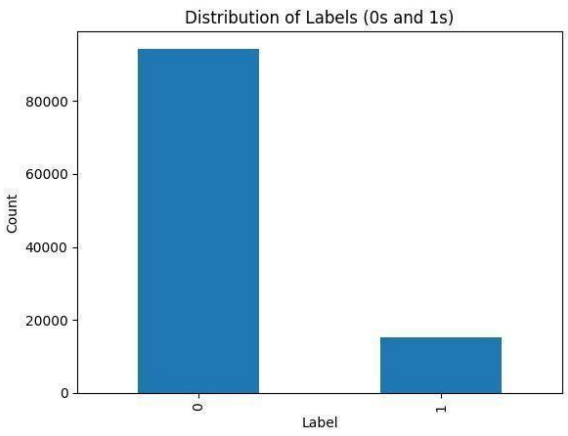
The dataset used to train this model has been taken from the Jigsaw Toxic Comment Classification Challenge on Kaggle. This dataset includes 159571 total comments

with 6 labels defined as - toxic, severe_toxic, obscene, threat, insult, and identity_hate. Each comment can have more than one label i.e a comment can classify as severe_toxic as well as obscene.For example, if a comment is labeled as 100100, it is toxic and a threat.So, 1 is for toxic and 0 is for non-toxic comments.




ii. Distribution of toxic comments in original dataset

To make the labeling simpler, we have created another column named ‘label’ which takes the value of 1 if one or more of the above mentioned 6 labels are 1. Since this dataset contains 150000+ comments, we have reduced the size down to 109571 comments. However, since the number of toxic comments is much less compared to the positive comments, we have randomly dropped 50000 of those comments which have labels 0 only, to retain the number of toxic comments (15294).




iii.Distribution of toxic comments after dropping

## B. Data Preprocessing

As this dataset is a collection of comments made by people, it is unorganized and raw. To make the text pertinent for training the model, preprocessing on this text needs to be performed. For each text, the following steps have been performed -

- Tokenizing the text into words or tokens. Tokenizing refers to splitting up a string or sentence into words or tokens. We have used NLTK’s punkt tokenizer to implement this.
- Conversion of all tokens to lowercase to ensure that the input text is in a consistent state.
- Punctuation removal to remove all the punctuation marks from the input text as they reduce noise, simplify text and maintain consistency throughout the text.
- Performing lemmatization on the tokens. Lemmatization refers to

getting the root word or the base word which is known as the *lemma*. So, for example, running would change to run.

- Finally, joining these tokens back into a string.

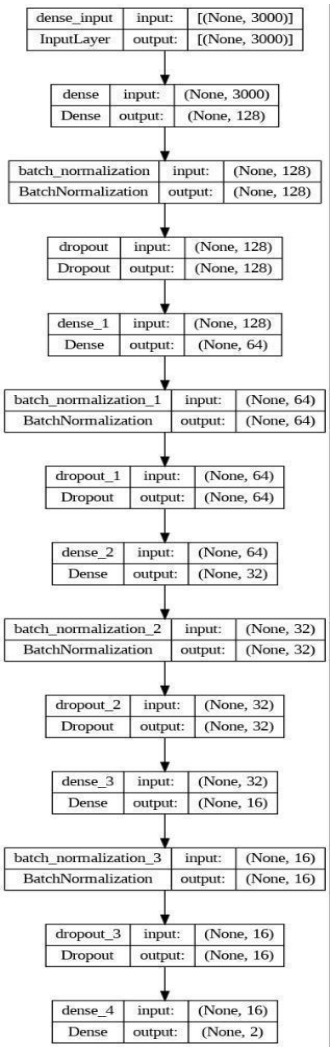
After preprocessing has been performed to clean the data, we have used the Tfidf vectorizer to convert the preprocessed text into a numerical format which is needed so that we can train the Neural Network. TF-IDF i.e. Term Frequency Inverse Document Frequency, is an algorithm to transform text into numbers which can be used to fit a machine algorithm for prediction. After applying, we get the TF-IDF matrix which is converted into a dense Numpy array as the matrix is usually sparse. Each row in this array represents a document and each column is a unique word from the corpus. The Tokenizer was fit on only the 30000 most frequent distinct words in train data, so that we don't need to consider the lesser used words.

After retrieving the features (words) it is stored in a pandas dataframe. After this, we apply one-hot encoding on the output label and fill the null values with 0.

## C. Building the Model

Before training the model, we have split the data into training and testing using sklearn's train_test_split method where 80% of the data is used for training and the remaining 20% is used for testing.

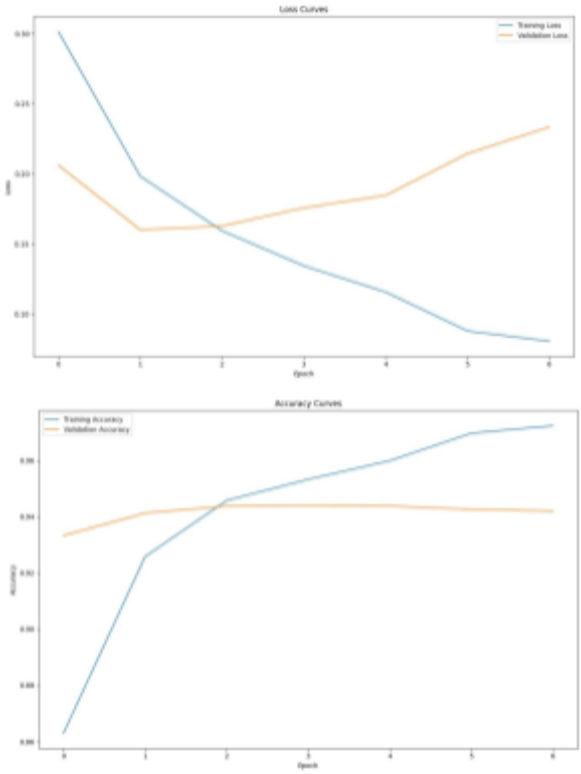
This neural network has 6 layers (including input and output layers). The hidden layers consist of 128, 64, 32 and 16 neurons respectively and the relu activation function has been used for the first three hidden layers whereas the sigmoid activation function has been used for the last hidden layer. As the value for sigmoid ranges from 0 to 1, using it in the precedent layers can cause information loss. For example - for both 10000 and 20000 in a sigmoid function, the output will converge to 1 despite having a 10000 difference which results in information loss. All these layers are dense layers with a dropout of 0.5 on each which drops out random neurons during training and along with this, we have used the L2 regularization as well. Batch regularization in each layer normalizes the values given from the precedent network before performing the activation function on it. We need these to prevent overfitting. The final output layer has 2 neurons (as this is a binary classifier) which will classify the text into toxic or non-toxic.




iv.Structure of the Neural Network

For compiling the model, we have used the adam optimizer, for the loss function we have used the binary cross entropy and accuracy as our evaluation metric.

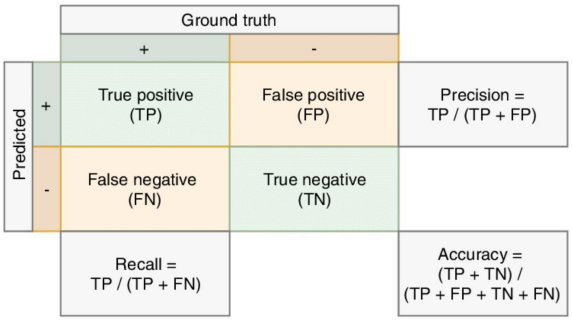
We have made use of several callback methods and after running our model, the training accuracy came out to be 93.32% and the validation accuracy came out to be 94.01%.



v.Loss and accuracy curves

## D. Result

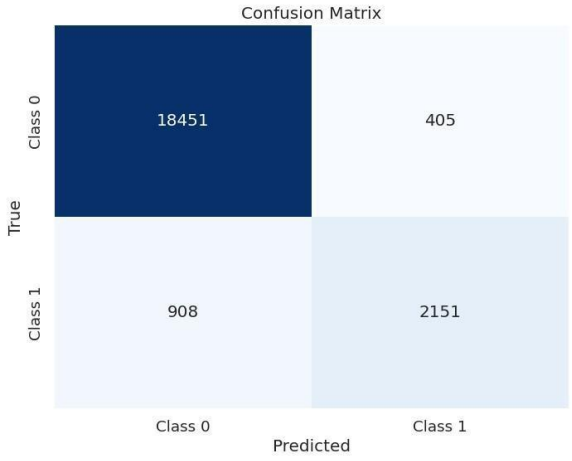
A confusion matrix is a table for understanding how well our model has performed. Each cell labels a number which can be – True positive, True Negative, False Positive and False Negative. In our, for example, true positives are the inputs which were labelled as toxic in the dataset and our model also correctly predicted it as toxic. False negatives are those inputs which are toxic but our system incorrectly labelled as non-toxic.




vi.Confusion Matrix

The accuracy label gives us the percentage of observations which were labelled correctly but we don't use this for text classification tasks as it doesn't work well when the classes are unbalanced.

Precision measures the percentage of the True Positive inputs. Recall measures the percentage of items actually present in the input that were correctly identified by the system.




vii. Confusion matrix for our model

As we can see in this confusion matrix, most of the values were labelled correctly.

Accuracy on test data : 94.01%

| Class | Precision | Recall | F1 score | Support |
|---|---|---|---|---|
| 0 | 0.95 | 0.98 | 0.97 | 18856 |
| 1 | 0.84 | 0.70 | 0.77 | 3059 |

All these scores were calculated with : epoch = 20, batch_size = 128.

Output of some sample texts which were put into our model :

Input sentence : go kill your self u absolute pathetic sob story

Predicted Class : 1 (toxic) Input sentence : It was really nice meeting you yesterday

Predicted Class : 0 (non-toxic)

Input sentence : I hate you so much it physically hurts to be near you

Predicted Class : 1 (toxic)

## V. CONCLUSION

Toxic comments can cause emotions of worry, despair, and poor self-esteem, particularly among young individuals who are still building their sense of identity. It can have serious impacts on their confidence level. This can have long-term consequences for their mental health and well-being. The solution to this problem is to develop an efficient model capable of detecting the amount of toxicity in remarks such as threats, vulgarity, insults, racism, and so on. So, in this paper, we have attempted to make a model which will filter out such comments and make the online environment more positive.

We conclude the following :

1. This model is able to classify a comment as toxic or non toxic based on the 6 labels provided in the original dataset.
2. The accuracy is 94.82%, the recall is 70.36%, and the F1 score is 76.61%.

## VI. REFERENCES


1. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit by Steven Bird Ewan Klein Edward Loper
2. Speech and Language Processing An Introduction to Natural Language Processing,Computational Linguistics, and Speech Recognition Daniel Jurafsky Stanford University James H. Martin University of Colorado at Boulder
3. Natural Language Understanding by James Allen
4. The Handbook of Computational Linguistics and Natural Language Processing by Alexander Clark, Chris Fox, and Shalom Lappin
5. Handbook of Natural Language Processing Machine Learning & Pattern Recognition Series by Nitin Indurkhya and Fred J. Damerau
6. Natural Language Processing: History, Evolution, Application, and Future Work by Prashant Johri Sunil Kumar Khatri
7. The Evolution of NLP from 1950 to 2022 Analytics Vidya
8. Evolution of Natural Language Processing Fresh Gravity
9. Toxic Comment Detection using LSTM Krishna Dubey Rahul Nair Mohd. Usman Khan