



Toxic Comment Classifier


Aryaan Peshoton I049
Lehar Rathore I053
Manya Sahay I055



Overview

O1

Dataset + preprocessing



Acquiring the dataset from kaggle
and using common preprocessing

O2

Feature Extraction

Using tfidf

O3

Building a Neural Network

Defining parameters, vocabulary
etc

O4

Testing

Using unseen data to test the
model

INTRODUCTION

- The proliferation of toxic and harmful comments has become a pressing issue. Social media platforms, discussion forums, and comment sections on various websites often play host to offensive, abusive, and harmful content that causes harm to users.
- With the help of Natural Language Processing we can create a toxic comment filter and other applications which are crucial tools for maintaining safe and constructive online environments.

PROBLEM STATEMENT

Toxic comments, encompassing hate speech, harassment, and offensive language, not only erode the quality of online discussions but also present serious ethical and psychological concerns.

Our project aims to address this pervasive issue by developing and implementing a solution for the identification and categorisation of toxic comments.

The primary objective is to empower digital platforms and online communities with a proactive and scalable mechanism to protect users from harm while preserving the principles of free expression and open discourse.



APPROACH



There are various ways one can explore when developing a toxic comment filtration system using NLP, some are mentioned below:

- Keyword-Based Filtering
- Machine Learning Models
- Ensemble Models
- Rule-Based Systems
- User Feedback Integration

The choice of approach depends on factors like available data, computational resources, the specific requirements of the platform, and the desired level of precision and recall in filtering toxic comments.



APPROACH



Keyword-Based Filtering:

Identify and maintain a list of keywords and phrases commonly associated with toxic comments. Flag or remove comments containing these keywords. Simple, but may have high false-positive rates and limited adaptability.

Ensemble Models:


Combine multiple machine learning models or NLP techniques to improve accuracy. Voting, stacking or boosting methods can be employed for ensembling.

Rule-Based Systems:

Define a set of rules to detect toxic behavior. Rules may be based on specific patterns, syntactic structures, or regular expressions.

User Feedback Integration:

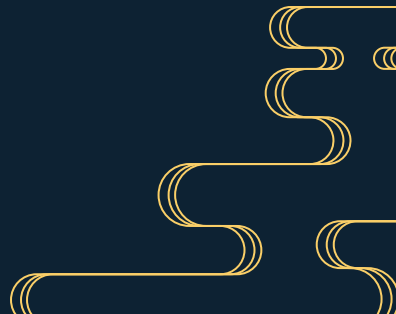
Allow users to flag comments as toxic or non-toxic. Use this feedback to improve the model and filter toxic comments.



OUR APPROACH:

STEPS:

- Data Collection and Labeling: Gather a dataset of comments labeled as toxic or non-toxic.
- Text Preprocessing: Clean and format the text data for analysis.
- Feature Engineering: Extract relevant features from the text data.
- Model Selection: Choose a suitable algorithm for text classification.
- Training: Train the model using the labeled data.
- Evaluation: Assess the model's accuracy and other metrics on a validation dataset.
- Testing and Deployment: Test the model on new data, integrate it into the platform.



Links

<https://colab.research.google.com/drive/1hC2iJPT0xIW9y8NOVX8rJBnBNjP4hx1s?usp=sharing>

