

# Context Aware Comment Moderation System

<sup>1</sup>K. Sreeveda, <sup>2</sup>B. Varshitha, <sup>3</sup>P. Abhigna, <sup>4</sup>G. Indu Maithreyee, <sup>5</sup>B. Nashwitha

<sup>1</sup>Asst. Professor, Dept. of CSE (Data Science), G. Narayanamma Institute of Technology and Science (For Women), Hyderabad, India

<sup>2,3,4,5</sup>Dept. of CSE (AI & ML), G. Narayanamma Institute of Technology and Science (For Women), Hyderabad, India

E-mail: <sup>1</sup>[sreeveda.karrepu@gmail.com](mailto:sreeveda.karrepu@gmail.com), <sup>2</sup>[varshithavidyasagar@gmail.com](mailto:varshithavidyasagar@gmail.com), <sup>3</sup>[abhignapeddhagouni@gmail.com](mailto:abhignapeddhagouni@gmail.com),  
<sup>4</sup>[govrojiindumaithreyee@gmail.com](mailto:govrojiindumaithreyee@gmail.com), <sup>5</sup>[nashwitha25@gmail.com](mailto:nashwitha25@gmail.com)

**Abstract** - Aware Comment Moderation System that uses TF-IDF based classifiers, BERT-based language model, and rules. The models used for providing quick prediction include Logistic Regression, SGD Classifier, and Naïve Bayes, and BERT is used for those comments whose meaning depends highly on the context. FastAPI serves as an interface between prediction service and web interface while the database maintains the user, post, comment, score, and moderation. The comment is then classified as Non-Toxic, Mild Toxic, or Severe Toxic before any action is taken against the comment like approving, sending for review, or rejecting.

**Keywords:** Aware Moderation, BERT, Natural Language Processing, TF-IDF, FastAPI, Text Classification.

## I. INTRODUCTION

Commenting on social networks, discussion boards and other online communities is quite common nowadays. In addition to useful comments, there can also be some inappropriate and abusive comments that should be analyzed. It becomes impossible to check everything manually with an increasing amount of comments. Also, filtering by keywords does not always help as the same word can be innocent in one sentence and offensive in another.

The more complex comments are those that use indirection. Sarcasm, back-handed compliments, or negation may be present in such comments; they do not include the words which could have been recognized by simple filters [3]. In this project, the statistical representation of the text, its contextual representation, and the certain language rules are combined together in order to process such comments and simple ones as well.

### 1.1 Problem Statement

Two of the mistakes that a moderation system may be likely to commit include mistaking a harmless comment for an abusive one on the basis of a word, or not recognizing an insult on the basis of lack of keywords. The proposed system will mitigate these two risks and will give a consistent

decision by categorizing comments into Non-Toxic, Mild Toxic, or Severe Toxic categories.

### 1.2 Objectives

The tasks include cleaning of the comments dataset, creation of TF-IDF and context embeddings, building and comparison of logistic regression, SGD classifier, Naïve Bayes and BERT-based classifiers, and rules creation for complex cases of language. The final product is also supposed to provide integration of these classifiers into the web application, so that moderation is performed after entering a comment.

### 1.3 Methodology

The process consists of five steps, which are: Data collection, preparation, selection of useful representations, training the models, evaluation and integration of the whole process with validation. The split ratio for training and testing is 80% and 20%. TF-IDF will be used for the classical models [9], and BERT will provide context to the classification [2]. Their results will be aggregated along with the rule checking results.



Figure 1: Flow Chart of the Proposed Methodology

#### 1.3.1 Data Collection and Preprocessing

Data set contains a combination of public comment data and selected samples of implied and contextual abuse. Prior to training, duplicated or irrelevant entries are filtered out and the text inconsistencies are cleaned. Lowercasing is done, and the necessary symbol handling and tokenizing are done prior to using the BERT tokenizer on the input text.

#### 1.3.2 Feature Selection and Representation

In TF-IDF, each term is assigned a score according to how many times it has been used in the comment as well as

how many times it has been used in the entire corpus. They are fed into the Logistic Regression, Naive Bayes, and SGD classifiers. In BERT, the relationship between the terms in the sentence is taken into account rather than considering them as separate terms.

## 1.4 System Design and Algorithms

### 1.4.1 System Architecture

The system is split into a few clear stages: input handling, preprocessing and feature extraction, model prediction, decision making, storage, and presentation. Once a comment is submitted, it is cleaned and prepared for both the classical models and BERT. Their results are considered together with the rule checks, and the final class and moderation action are then stored and shown in the web interface.

### 1.4.2 Logistic Regression

After the comments have been preprocessed, they get vectorized using TF-IDF vectors and then divided into training and test datasets. Logistic Regression is able to learn the relationship between the features and the classes. While making predictions, it outputs a class along with its probability.

### 1.4.3 SGD Classifier

The SGD classifier trains using the TF-IDF features via Stochastic Gradient Descent (SGD). The strength of this classifier in our project is that it can process a lot of text data without consuming a lot of computational power. This is used along with other signals as input to our decision-making process.

### 1.4.4 Naive Bayes

Multinomial Naive Bayes estimates the probability of each class from the word patterns represented by the TF-IDF vectors. It is fast and straightforward to train. However, its conditional-independence assumption can be limiting when the meaning of a sentence depends strongly on how the words relate to one another.

### 1.4.5 TF-IDF Representation

In TF-IDF, each term is assigned a score according to how many times it has been used in the comment as well as how many times it has been used in the entire corpus [9]. They are fed into the Logistic Regression, Naive Bayes, and SGD classifiers [9]. In BERT, the relationship between the terms in the sentence is taken into account rather than considering them as separate terms.

### 1.4.6 BERT-Based Contextual Classifier

BERT takes a word contextually by bidirectional attention with respect to the entire sentence [2]. The framework uses a pretrained Toxic-BERT framework to calculate a toxicity score [10]. It helps especially in cases where a comment is sarcastic, an indirect insult, has negation or words which are not judged well from their word frequency.

### 1.4.7 Hybrid Decision Layer

The final decision is made by looking into the classical model results, the BERT probability, and the chosen rules for the specific language. The rules find the following features: implicit attacks, sarcasm, negation, and mixed emotions. The result of these factors is then compared against the chosen thresholds, which lead to one of three classifications: NonToxic, Mild Toxic, or Severe Toxic.

### 1.4.8 Training and Evaluation

Classical models act as effective baselines while the BERT model is employed in cases where contextual classification is required. The accuracy, precision, recall, F1-score, and ROCAUC metrics are used in order to evaluate the system. Validation and parameters' tuning are performed during development in order to test prediction stability and avoid overfitting.

## II. IMPLEMENTATION

### 2.1 Dataset Attributes and Context Signals

These include comment id, comment text, label, text length, TF-IDF features, BERT embeddings, context features, insult detection, sarcasm detection, negations, sentiment balance, toxicity score, moderation status, context confidence, agreement among models, comment complexity, language diversity, and final label. The information helps in the prediction task and is useful for reviewing hard cases if required.

### 2.2 Technology Stack

Project implementation is performed with Python. Pandas and NumPy libraries are used for data processing, and scikitlearn provides TF-IDF and classical models [9]. Hugging Face Transformers and PyTorch libraries provide BERT model [10], and Joblib library is used to store trained models. Backend consists of FastAPI and Uvicorn libraries. Front end consists of HTML, CSS, and JavaScript, while MySQL/SQLite stores users, posts, comments, and moderation results.

## 2.3 Moderation Workflow

When the comment has been made by the user, the text is then normalized and input into the prediction service. The classical algorithms will give statistical scores, while BERT adds context, and the rules layer checks that the selected language patterns are correct. All these inputs will result in the score and the class of the output. In case the output is found to be safe, it is accepted, in case it is questionable, it may undergo further manual verification, and in case it is toxic, it is rejected.

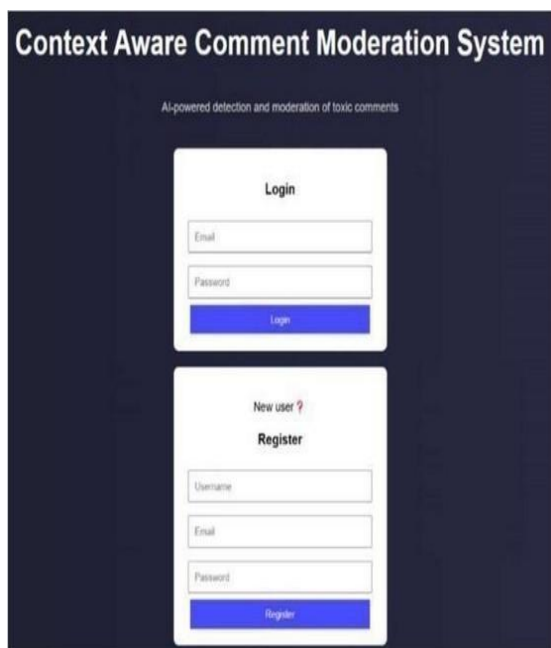


Figure 2: Login and Registration Interface

User Registration Page. The page through which new users register themselves by inputting their personal information such as name, email, and password.

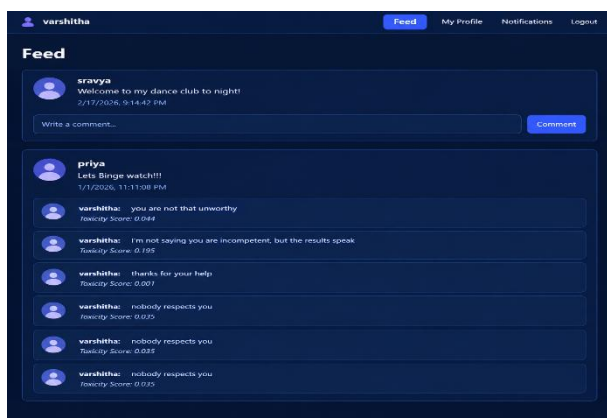


Figure 3: Feed and Comment Interface

Social media feed user interface showing user posts and the toxic scores of their comments.

## III. RESULTS AND DISCUSSION

The completed interface indicates how the interaction between the web application and moderation service takes place. A user has the possibility to register, log in into his account, post messages, and write comments. Once the comment is processed, its class and score are displayed via the interface.

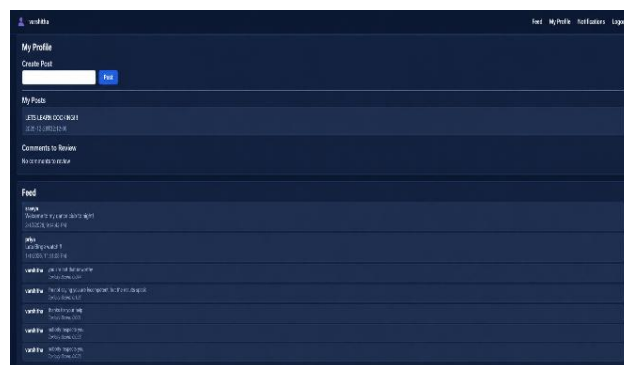


Figure 4: User Profile Dashboard

User profile dashboard allowing the creation of posts, personal content viewing, and interaction management all within one interface.

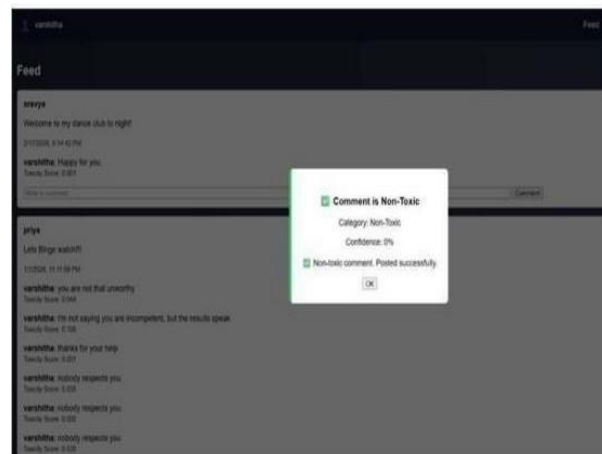


Figure 5: Moderation System Interface

Upon clicking the submit button, the prediction result will show up below with the output being either Non-Toxic, Mild Toxic, or Severe Toxic, along with a toxicity rating.

The system will pop up a message saying that the comment has been identified as non-toxic and posted successfully.

### 3.1 Visual Performance Analysis

The heatmap shows the accuracy, precision, recall and F1score scores reported for the compared models. Naive Bayes has the highest precision score of the classical models

mentioned, but its recall is comparatively low. The SGD Classifier lies somewhere in the middle of the other baseline models, while the contextual part is responsible for handling the semantics of the sentences.

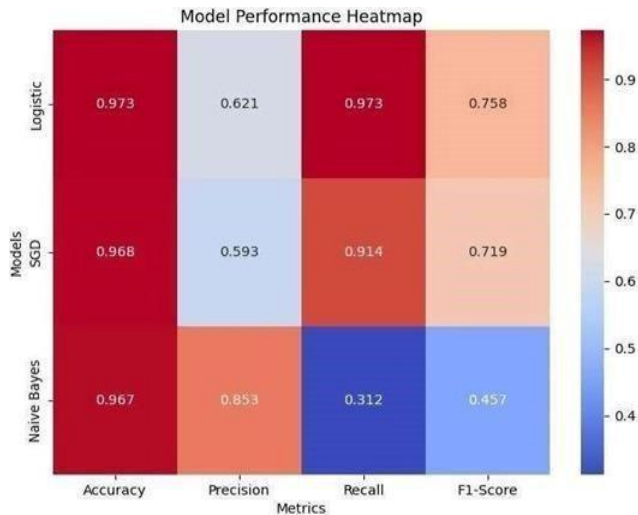


Figure 6: Performance Heatmap

### 3.2 Real-Time Moderation and Usability

The FastAPI is responsible for being the service layer upon submission of comments to the system. Users are given feedback while within the application itself, thus, it becomes easier to tell whether the comments have been moderated. The fact that the moderation records are stored also enables the system to review previous cases.

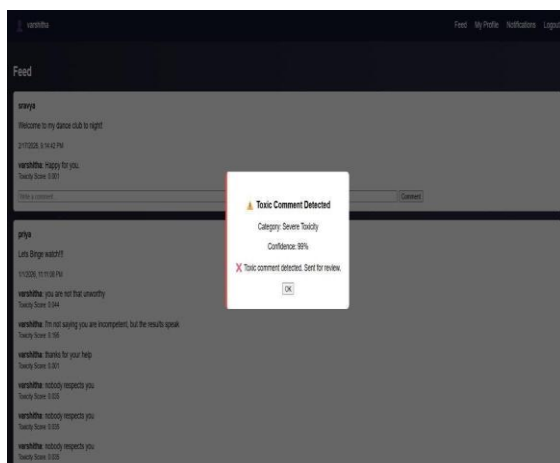


Figure 7: Comment Confirmation and Moderation Pop-up

A pop-up alert that there is a high level of toxicity in the comment with high confidence.

### 3.3 Limitations and Practical Considerations

The contextual model is more resource-intensive than the classical classification models, and thus its application requires more computational power. Comments could also be

difficult to understand when they are very ambiguous or employ terminology that was not part of the training dataset.

### 3.4 Future Enhancements

Expansion could take place in several ways such as multilingual moderation, improved context models, learning from moderators, user behavioral characteristics, adaptive thresholds, incorporation in platforms, visualization, and monitoring. All these aspects would help the model keep upto-date with new vocabulary being used on the Internet.

### 3.5 Model Performance and Comparison

#### 3.5.1 Reported Model Performance

Table I: Reported Model Performance

| Model               | Accuracy | Precision | Recall | F1    |
|---------------------|----------|-----------|--------|-------|
| Logistic Regression | 0.973    | 0.621     | 0.973  | 0.758 |
| SGD Classifier      | 0.968    | 0.593     | 0.914  | 0.719 |
| Naive Bayes         | 0.967    | 0.853     | 0.312  | 0.457 |
| Hybrid Model        | 0.973    | 0.621     | 0.973  | 0.758 |

It is obvious that the models carry out different tasks. The traditional methods are fast and can be considered a simple filter to discriminate between the comments. The data in Table I are cited from the project documentation and were not verified again for this report.

#### 3.5.2 Model Comparison

Table II: Model Comparison

| Feature                    | Logistic Reg. | Naive Bayes          | SGD                   | BERT                |
|----------------------------|---------------|----------------------|-----------------------|---------------------|
| Context understanding      | Limited       | Very limited         | Limited               | Strong              |
| Sarcasm / implicit meaning | Poor          | Poor                 | Poor                  | Very good           |
| Training speed             | Fast          | Very fast            | Very fast             | Slow                |
| Computational efficiency   | Efficient     | Highly efficient     | Highly efficient      | Requires high power |
| Moderation role            | Good baseline | Fast basic filtering | Large-scale filtering | Nuanced toxicity    |

Classical methods are suitable for fast statistics-based filtering, but BERT is more costly from a computational

perspective while having a higher capability of understanding the context. This is particularly true when the meaning of a comment is determined by the relationships between words.

#### IV. CONCLUSION

However, the obtained results prove that the classical algorithms can be used when there is a need for quick processing, but the advantages of each of them are quite different. Thus, Naive Bayes provides very precise results but at the same time has low recall, while Logistic Regression and SGD Classifier demonstrate higher baseline results. However, it is particularly crucial at the contextual stage in case there is no sufficient data in words to evaluate the comment's quality.

Speaking of future possible improvements, one can name support of several languages, improvement of the algorithm of context analysis, moderator's feedback, adaptive threshold, users' information, integration into other systems, and monitoring.

#### REFERENCES

- [1] "Toxic Comment Classification Challenge," *Kaggle*.
- [2] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional

- Transformers for Language Understanding," *NAACL-HLT*, 2019.
- [3] A. Schmidt and M. Wiegand, "A Survey on Hate Speech Detection using Natural Language Processing," *SocialNLP*, 2017.
- [4] T. Davidson, D. Warmley, D. Macy, and I. Weber, "Automated Hate Speech Detection and the Problem of Offensive Language," *ICWSM*, 2017.
- [5] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep Learning for Hate Speech Detection in Tweets," *WWW*, 2017.
- [6] Z. Zhang, D. Robinson, and J. Tepper, "Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network," *ESWC*, 2018.
- [7] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," *EMNLP*, 2014.
- [8] A. Vaswani *et al.*, "Attention Is All You Need," *NeurIPS*, 2017.
- [9] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, 2011.
- [10] T. Wolf *et al.*, "Hugging Face Transformers: State-of-the-art Natural Language Processing," *EMNLP*, 2020.

#### Citation of this Article:

K. Sreeveda, B. Varshitha, P. Abhigna, G. Indu Maithreyee, & B. Nashwitha. (2026). Context Aware Comment Moderation System. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 10(9), 164-168. Article DOI <https://doi.org/10.47001/IRJIET/2026.109017>

\*\*\*\*\*