Toxic Comment Classification Using Machine Learning on Social Media Data

Tram Tran, An Nguyen, Thu Nguyen, Tam Tran, Viet Tran
Department of Computer Science
Cal Poly Pomona

***Abstract*—In the digital age, as online platforms continue to grow, they face major challenges in moderating harmful content. Such harmful contents even have a huge negative impact on users' real lives. To protect users from offensive language, social media companies have implemented to flag comments or block abusive users. This project aims to develop a machine learning model that can classify harmful comments on social media. These online toxic comments are categorized into six types, including toxic, severely toxic, obscene, threat, insult, and identity hate [1]. Using publicly available data from the Jigsaw/Google Toxic Comment Classification Challenge on Kaggle, we will explore various natural language processing (NLP) techniques to process textual data and extract meaningful features. We will also experiment with multiple classification models, such as Decision Tree, K-Nearest Neighbors (KNN), and Logistic Regression. Model performance will be evaluated using metrics including accuracy, precision, recall, and F1-score. The goal of this project is to provide efficient and interpretable classification to assist content moderation systems in promoting a safer and healthier online environment.**

I. Introduction

A. Problem Description

The scale and speed of online interactions have grown significantly since email-based communication gave way to modern social media. Moderation of content has become much more difficult as a result of this evolution, particularly when it comes to offensive or toxic comments. Because of the enormous number of posts that are created every day, it is nearly impossible to manually identify abusive content. Automated systems that can effectively classify and moderate such content are therefore desperately needed. The goal of this project is to create a safer online environment by identifying and filtering toxic comments online using machine learning techniques that we specifically studied in class: Decision Tree, K-Nearest Neighbors (KNN), and Logistic Regression.

B. Proposed Solution

To address the problem of identifying toxic comments on social media, this project proposes a machine learning-based solution that applies classification models studied in class, including Decision Tree, K-Nearest Neighbors (KNN), and Logistic Regression. The dataset used will be the publicly available Jigsaw/Google Toxic Comment Classification dataset from Kaggle, which contains thousands of user comments labeled across six toxicity categories: toxic, severely toxic, obscene, threat, insult, and identity hate.

The solution begins with data preprocessing, including steps such as converting text to lowercase, removing punctuation, stopwords, and unnecessary characters, and applying stemming or lemmatization. We will then extract features from the cleaned text using methods like bag-of-words and `TF-IDF` to convert textual data into numerical form suitable for machine learning algorithms. Each classification model will be trained using the same preprocessed dataset to allow for fair performance comparison. Since a single comment can belong to multiple categories, a multi-label classification approach will be adopted using appropriate methods for each model. Performance will be evaluated using metrics such as accuracy, precision, recall, and F1-score.

By comparing results across the three models, we aim to determine which classifier provides the most effective balance between accuracy and interpretability, ultimately helping automate content moderation in a scalable and responsible way.

II. Dataset

This project utilizes the publicly available dataset from the Toxic Comment Classification Challenge hosted on the Kaggle platform. The dataset comprises user-generated comments sourced from Wikipedia's talk pages, each annotated for multiple types of toxicity. The classification task is multi-label in nature, where each instance may be associated with one or more of the defined toxicity categories.

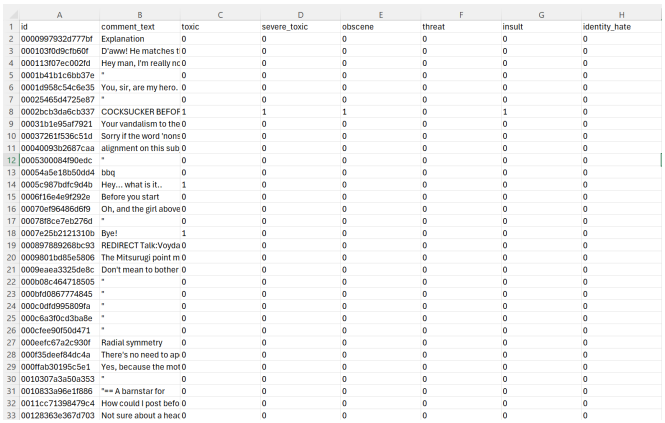
The following files were employed in the course of this project:

A. train.csv

The `train.csv` file serves as the primary training set and includes a total of 159,571 samples. Each row represents a unique comment, identified by an `id`, along with the raw text under the `comment_text` field. Corresponding to each comment are six binary columns indicating the presence (1) or absence (0) of the following toxicity labels: `toxic`, `severe_toxic`, `obscene`, `threat`, `insult`, and `identity_hate`.

B. test.csv

The `test.csv` file contains 153,164 unlabeled comments used for inference and prediction generation. Each entry comprises an `id` and a `comment_text` field. The true labels for this file were not released during the competition period.



	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
2	0000997932d777bf	Explanation	0	0	0	0	0	0
3	000103f0d9cfb60f	D'aww! He matches t	0	0	0	0	0	0
4	000113f07ec002fd	Hey man, I'm really nc	0	0	0	0	0	0
5	0001b41b1c6bb37e	"	0	0	0	0	0	0
6	0001d958c54c6e35	You, sir, are my hero.	0	0	0	0	0	0
7	00025465d4725e87	"	0	0	0	0	0	0
8	0002bcb3da6cb337	COCKSUCKER BEFOF	1	1	1	0	1	0
9	00031b1e95af7921	Your vandalism to the	0	0	0	0	0	0
10	00037261f536c51d	Sorry if the word 'non	0	0	0	0	0	0
11	00040930268672ca	alignment on this sub	0	0	0	0	0	0
12	0005300084f90edc	"	0	0	0	0	0	0
13	00054a5e18b50dd4	bbq	0	0	0	0	0	0
14	0005c987bdfc9d4b	Hey... what is it..	1	0	0	0	0	0
15	0006f16e4e9f292e	Before you start	0	0	0	0	0	0
16	00070ef96486d6f9	Oh, and the girl above	0	0	0	0	0	0
17	00078f8ce7eb276d	"	0	0	0	0	0	0
18	0007e25b2121310b	Bye!	1	0	0	0	0	0
19	000897889268bc93	REDIRECT Talk:Voyda	0	0	0	0	0	0
20	0009801bd85e5806	The Mitsurugi point m	0	0	0	0	0	0
21	0009eaea3325de8c	Don't mean to bother	0	0	0	0	0	0
22	000b08c464718505	"	0	0	0	0	0	0
23	000bfd0867774845	"	0	0	0	0	0	0
24	000c0dfd995809fa	"	0	0	0	0	0	0
25	000c6a3f0cd3babe	"	0	0	0	0	0	0
26	000cfee90f50d471	"	0	0	0	0	0	0
27	000eefc67a2c930f	Radial symmetry	0	0	0	0	0	0
28	000f35deef84dc4a	There's no need to ap	0	0	0	0	0	0
29	000ffab30195c5e1	Yes, because the mot	0	0	0	0	0	0
30	0010307a3a50a353	"	0	0	0	0	0	0
31	0010833a96e1f886	"== A barnstar for	0	0	0	0	0	0
32	0011cc71398479c4	How could I post befo	0	0	0	0	0	0
33	00128363e367d703	Not sure about a heac	0	0	0	0	0	0

Fig. 1. Sample rows from the `train.csv` file, showing the comment text and associated toxicity labels.

C. sample_submission.csv

This file provides the expected format for model submissions. It contains the `id` field and six toxicity-related columns. The values are floating-point numbers between 0 and 1, representing the predicted probability of each label being applicable to the corresponding comment in the test set.

D. test_labels.csv

Following the conclusion of the Kaggle competition, the organizers released `test_labels.csv`, which contains the ground truth labels for the test set. The structure mirrors that of `train.csv`, comprising the same six binary toxicity labels. However, some entries are annotated with a value of `-1`, signifying that those specific samples were excluded from the competition's scoring phase. For academic purposes, this file provides a means for evaluating model performance on the previously unseen test data.

III. Methodology

The proposed Toxic Comment Classification System follows the Knowledge Discovery in Databases (KDD) process. The methodology is divided into the following stages: data selection, preprocessing, transformation, data mining (modeling), and evaluation.

A. Data Selection

We used the publicly available dataset from Kaggle's Toxic Comment Classification Challenge. To optimize performance and reduce memory usage, we randomly sampled 30,000 comments per run from the full dataset. Each comment is labeled across six toxicity categories: `toxic`, `severe_toxic`, `obscene`, `threat`, `insult`, and `identity_hate`.

B. Data Preprocessing

We applied several text cleaning techniques to prepare the comments for analysis. These steps included converting text to lowercase, removing URLs, punctuation, and stopwords, and applying lemmatization or stemming to reduce words to their root forms. The cleaned version of each comment was stored in a new column named `clean_text`, which was then used for the transformation phase.

C. Data Transformation

We used the `TF-IDF` (Term Frequency-Inverse Document Frequency) vectorizer to convert the cleaned text into numerical features. This technique helped represent each comment by its most informative words, emphasizing terms that are important in a particular comment but not common across all comments. We limited the vocabulary size to the top 30,000 most frequent terms to reduce the feature space and improve efficiency. The result was a sparse matrix of `TF-IDF` scores, which served as input features for the model. The target labels `(y)` were structured for multi-label classification, containing binary indicators for each of the six toxicity types.

D. Data Mining

The following machine learning models will be applied to the transformed dataset:

- **Decision Tree Classifier**: Serves as a baseline model for comparison.
- **K-Nearest Neighbors (KNN)**: Classifies based on the most common label among the nearest neighbors in the feature space.
- **Artificial Neural Networks (ANNs)**: Capable of modeling complex patterns in text data.
- **Naive Bayes**: Probabilistic classifier based on Bayes' Theorem; performs well on high-dimensional text data and is efficient for baseline comparisons.
- **Genetic Algorithms (GA)**: Used for feature selection and model optimization.

Each model was trained on the `TF-IDF` features, except for Genetic Algorithms, which utilized a reduced feature space generated via Truncated Singular Value Decomposition (`SVD`). We recorded both the training time and model accuracy to generate evaluation reports.

E. Evaluation

The model will be evaluated using these performance metrics: accuracy, precision, recall, and F1 score. Moreover, the final models and insights will be presented through visualizations, graphs, and tables.

IV. Results

	Model	Accuracy (%)	F1 Score (Weighted)	Precision (Weighted)	Recall (Weighted)	Training Time (min)
0	Decision Tree	83.57	0.54	0.45	0.67	2.23
1	KNN	90.39	0.23	0.74	0.14	0.00
2	Naive Bayes	90.32	0.21	0.85	0.12	0.88
3	ANNs	85.95	0.57	0.54	0.61	45.34
4	Genetic Algorithm	85.59	0.54	0.53	0.56	23.32

Fig. 2. Performance Comparison of Machine Learning Models on Toxic Comment Classification

Among all models, KNN achieved the highest accuracy (90.39%), closely followed by Naive Bayes (90.32%) and Artificial Neural Networks (ANNs) (80.95%). However, (ANNs) yielded the best F1 score (0.57) and demonstrated strong overall precision and recall, indicating balanced performance across all classes despite the long training time (45.34 minutes).

While Naive Bayes and KNN had high accuracy and precision, their low F1 scores and recall suggest poor handling of imbalanced or minority classes. In contrast, the Decision Tree model provided relatively balanced metrics with moderate training time, making it a practical choice in time-sensitive scenarios.

Genetic Algorithms (GAs), although not traditional classifiers, were used here to optimize linear weights. The GA model achieved competitive accuracy and precision but had a lower F1 score due to weaker recall.

V. Related Work

Several previous studies have explored the problem of toxic comment detection. Google's Perspective API provides a real-time toxicity scoring system, using deep learning techniques trained on large datasets [2]. In the Jigsaw Kaggle competition, many top teams used ensemble methods combining LSTM, GRU, and attention mechanisms to improve accuracy [1]. Unlike these deep models, our work focused on lightweight models like KNN, Naive Bayes, and Decision Tree to balance performance with interpretability and training efficiency.

VI. Conclusion

In this project, we developed a text classification system to detect toxic comments using natural language processing and machine learning techniques. We applied through text preprocessing, visualized patterns in toxic vs. non-toxic language, and trained multiple classifiers, including Decision Tree, Naive Bayes, Genetic Algorithm, ANNs, and KNN. The models were trained and evaluated on a random sample of 30,000 (instead of 150,000+) comments from the full dataset to reduce memory usage and training time.

The model was trained and evaluated using `TF-IDF` (except GA) features extracted from cleaned text, with visualization insights confirming that toxic comments tend to use more aggressive language and are slightly longer in length.

The result gave us an idea that choosing a model depends on the situation and what is the goal of your program. Each model has its own strengths and weaknesses, like ANNs, which have a good F1 score, but they take too long to train.

Project Repository

The code and additional resources are available at:
https://colab.research.google.com/drive/1nGiu4fT4Q0YMd6sc4hdwi015c4-BMosn?authuser=1

References

[1] "Toxic Comment Classification Challenge," *Kaggle.com*, [Online]. Available: https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data

[2] ""Perspective API" Jigsaw" *Google*, [Online]. Available: https://www.perspectiveapi.com/