Toxic Comments Classification Using Machine Learning

Mr. V. Kandhasamy[1], Mr. K. Sriram[2]
[1, 2]Department of Information Technology
[1, 2] Paavai Engineering College (Autonomous), Namakkal, Tamil Nadu, India

Abstract- *The proliferation of online platforms has facilitated widespread communication but has also led to the emergence of toxic comments, which can propagate hate speech, harassment, and misinformation. In this abstract, the present a study on leveraging machine learning techniques for the classification of toxic comments. We begin by assembling a diverse dataset of comments annotated for toxicity, encompassing various forms of harmful behavior. Utilizing this dataset, we explore a range of machine learning algorithms, including traditional classifiers and deep learning models, to develop effective toxicity detection systems. We investigate feature engineering techniques such as TF-IDF, word embeddings, and contextual embeddings to capture semantic and syntactic nuances in toxic language. Additionally, we address challenges such as dataset imbalance through strategies like oversampling, under sampling, and cost sensitive learning. Through rigorous experimentation and evaluation, we assess the performance of our models in terms of accuracy, precision, recall, and F1-score. Our results demonstrate the efficacy of machine learning approaches in accurately identifying toxic comments across diverse online platforms. By automating the identification of toxic content, our research contributes to fostering healthier online communities, empowering platforms to proactively moderate harmful behavior and promote positive discourse.*

Keywords- Leveraging, TF-IDF, rigorous, syntactic nuances.

I. INTRODUCTION

Social media provides a platform for various discussions, enabling individuals to freely express their opinions, often anonymously. However, this anonymity can be misused by those who strongly oppose a particular viewpoint. The problem of conversational toxicity of the discourages genuine self- expression and the exchange of opinions due to the fear of abuse or harassment. This project aims to utilize deep learning techniques to detect toxic content in text, thereby helping to discourage users from sending potentially hurtful messages, promoting more civil discussions, and evaluating the toxicity of other users' comments.

II. LITERATURE SURVEY

1. The paper “Toxic Comment Classification Using Hybrid Deep Learning Model” discusses about the hybrid models used which are Bidirectional gated recurrent network, convolution neural network and they achieved the accuracy of 98.39% and the f1 score of this model was 79.91%. As much as toxic comment classification is important, toxic span prediction also play the similar role which helps to build more automated moderation systems. The paper “multi-task learning for toxic comment classification and rationale extraction” discusses about the multi-task learning model using the Toxic XLMR for bidirectional contextual embeddings of input text for toxic comment classification and a Bi-LSTM CRF layer for toxic span and rationale identification. The dataset used was curated from Jigsaw and Toxic span prediction dataset. The model has outperformed the single task models on the curated and toxic span prediction models by 4% and 2% improvement for classification.

2. The author of “Impact of SMOTE on Imbalanced Text Features for Toxic Comments Classification using RVVC Model” discusses about the ensemble approach called regression vector voting classifier (RVVC), to identify the toxic comments over the social media platforms. The dataset for this is taken from Kaggle which is a multi-label dataset which contains labels as toxic, severe toxic, threat, insult, and identity hate. The values of the dataset are given as binary values which contains 158,640 comments in total with toxic comments.

3. 3.The paper “Application of Recurrent Neural Networks in Toxic Comment Classification” discusses about this aspect. The dataset used for this methodology is the public dataset which is provided by Conversation AI team. Even Though the previous models have given the accurate toxicity scores, the models still miss classify some texts that share similar patterns as toxic comments which can be reduced using the RNN method. They used the Word2Vec embedding model to train the model to

remove the noise in the data and the methodologies used are the recurrent neural network and done the comparison analysis with GRU. They have successfully employed the word2vec embedding and recurrent neural network in building a toxic comment classification model and achieved high accuracy with low cost.

Even with the existing models have achieved high accuracy, building the model takes more time and complexity of the model will higher. This leads to Automatic detection. The paper “Automatic toxic Comment Detection Using DNN” discusses about the automatic tools which where build using the LSTM and RNN to improve the accuracy and provide the results much faster than the traditional methods.

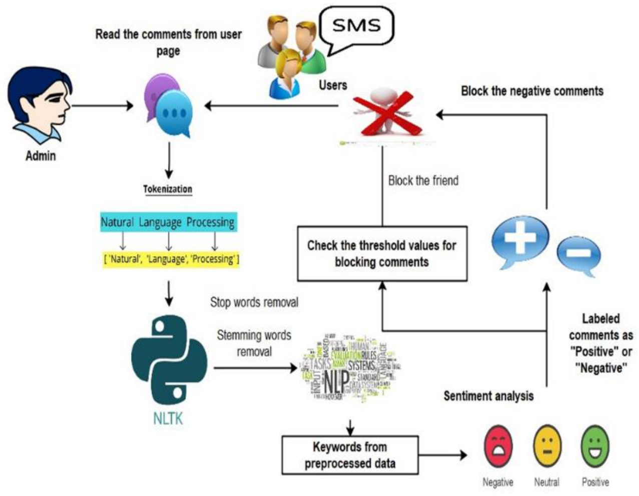
4. n (Ibrahim et al., 2019), the authors analyzed the data imbalance problem of the toxicity dataset and proposed a novel solution. The study shows the entire dataset consists of less than 7% of toxic comments. The skewed distribution might create a bias towards a majority label; thus, augmentation is applied to the data. The methods applied include removal of duplicates from minority classes, creation of new comments using random 20% content of the original text and creation of new comments but with some words replaced for their synonyms. The method was tested using a convolutional neural network, Bidirectional LSTM and GRU models. The results indicate that each method provided an improvement over the not augmented data. The best F measure (0.88) was recorded utilizing the CNN ensemble model and data processed using all three aforementioned methods.

5. Husnain (Husnain et al., 2021) proposed a different approach to the pre-processing of toxic comments to address the classification problems. After the data cleaning process that involved removal of stop words steming and tokenization authors extracted features according to the word length. The analysis of the features indicates that bigrams or words composed of two tokens are giving better results. The created training set was tested on a binary classification problem of detection of toxicity in the text and a multi-label classification problem. The algorithms used included Logistic Regression, Naïve Bayes, and Decision Tree classifiers. The results presented show over 95% accuracy of all models in a binary task and ~90% accuracy in multilabel problems. The best performing model was logistic Regression in both case scenarios.

III. PROBLEM IDENTIFICATION

Ensuring the dataset is properly labeled and balanced, with representative samples of toxic and non-toxic comments. Biases in the data can lead to biased models. Machine learning models can amplify biases present in the training data. It's crucial to regularly assess and mitigate bias during model training and evaluation. Ensuring the model generalizes well to unseen data, including comments with different languages, dialects, or cultural contexts. Understanding why a model makes certain predictions is important, especially in sensitive applications like toxic comment classification. Lack of interpretability can lead to mistrust and misapplication. Implementing the model in real-world applications requires considerations such as latency, scalability, and ethical implications. Toxic comment classification models need to adapt to evolving language trends and new forms of toxicity. Implementing mechanisms for continuous learning is essential.

IV. SYSTEM DESIGN


Read the comments from user page
SMS
Users
Admin
Block the negative comments
Tokenization
Block the friend
Natural Language Processing
['Natural', 'Language', 'Processing']
Check the threshold values for blocking comments
Stop words removal
Stemming words removal
Labeled comments as "Positive" or "Negative"
NLP
NLTK
Sentiment analysis
Keywords from preprocessed data
Negative
Neutral
Positive

V. PROPOSED SYSTEM

Online Social Networks (OSNs) are now one of the most popular interactive mediums for communicating, sharing, and disseminating a significant amount of human life data. One of the most important issues in today's On-line Social Networks (OSNs) is giving users the opportunity to control the messages posted on their own private area in order to prevent the appearance of undesired content. Until date, OSNs have been unable to meet this condition. To fill the void, we propose in this paper a mechanism that gives OSN users direct control over the messages put on their walls. This is accomplished via a flexible rule-based system that allows users to design the filtering criteria that will be applied to their walls, as well as a Machine Learning-based soft classifier that labels messages automatically to provide control based filtering. Deep learning (DL) is a text classification technique that uses machine learning to assign each brief text message to one of several categories based on its content.

The extraction and selection of a set of characterising and discriminating features is the focus of most efforts in developing a robust back propagation algorithm. A database of categorised terms is created here, which is then used to check the words for any inappropriate words. If the communication contains any vulgar terms, the message will be submitted to the Blacklists, which will filter those words out. Finally, as a result of the content-based-filtering technique, a message free of obscene terms will be posted on the user's wall. A system uses blacklists to automatically filter unwanted messages based on both message content and message creator relationships and characteristics. A revised semantics for filtering rules to better fit the considered domain, to assist users with Filtering Rules (FRs) formulation, and the extension of the collection of features examined in the classification process are some of the major differences. Finally predict the friends who are posted continues unwanted messages on user pages with alert system.

VI. EXISTING SYSTEM

In content-based filtering each user is assumed to operate independently. As a result, a content-based filtering system selects information items based on the correlation between the content of the items and the user preferences as opposed to a collaborative filtering system that chooses items based on the correlation between people with similar preferences. While electronic mail was the original domain of early work on information filtering, subsequent papers have addressed diversified domains including newswire articles, Internet “news” articles, and broader network resources. Documents processed in content-based filtering are mostly textual in nature and this makes content-based filtering close to text classification. The activity of filtering can be modeled, in fact, as a case of single label, binary classification, partitioning incoming documents into relevant and non-relevant categories. More complex filtering systems include multi-label text categorization automatically labeling messages into partial thematic categories. Content-based filtering is mainly based on the use of the ML paradigm according to which a classifier is automatically induced by learning from a set of pre-classified examples.

VII. FEASIBILITY STUDY

The purpose of this chapter is to introduce the reader to feasibility studies, project appraisal, and investment analysis. Feasibility studies are an example of systems analysis. A system is a description of the relationships between the inputs of labour, machinery, materials and management procedures, both within an organization and between an organization and the outside world.

Technical feasibility:

Technical Feasibility assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems.

Operational feasibility:

Operational Feasibility is depended on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements analysis phase of system development. Operational feasibility reviews the willingness of the organization to support the proposed system. This is probably the most difficult of the feasibilities to gauge. In order to determine this feasibility, it is important to understand the management commitment to the proposed project.

Economical feasibility:

Economic feasibility analysis is the most commonly used method for determining the efficiency of a new project. It is also known as cost analysis. It helps in identifying profit against investment expected from a project. Cost and time are the most essential factors involved in this field of study. For any system if the expected benefits equal or exceed the expected costs, the system can be judged to be economically feasible. In economic feasibility, cost benefit analysis is done in which expected costs and benefits are evaluated. Economic analysis is used for evaluating the effectiveness of the proposed system.

VIII. MODULES DESCRIPTION

This section explores our recommended approach, which includes two elements of the proposed software solution. To successfully execute the final solution, an extra module focused on extracting data from a specific social media profile would be required.

- Module 1: Dataset
- Module 2: Text processing
- Module 3: Data split
- Module 4: Model selection
- Module 5: Evaluation

Dataset:

To train the toxic detection model, we utilize a publicly available dataset specifically designed for toxicity classification. This dataset was curated by the Conversation AI team for an NLP challenge on Kaggle. It comprises six distinct labels: toxic, severe toxic, obscene, threat, insult, and identity hate. The original dataset is already divided into train and test sets, but for the purpose of our research, we combine both sets. In total, there are 223,549 comments, with the majority (201,081) being clean texts. The toxic label is the most prevalent, with many comments solely labeled as toxic. Interestingly, a small subset of 45 comments possesses all six toxic labels. The original dataset is sourced from Kaggle's toxic, severe toxic, obscene, threat, insult, and identity hate categories. Non-toxic comments belong to one class, while the remaining comments selected for analysis are those labeled as severe toxic, obscene, threat, or insult.

Text processing:

The initial step in text cleaning involved replacing short versions of words such as 's, 're, 'll, 'd, etc. Subsequently, all comments were converted to lowercase to ensure that capitalized words are not treated differently by the model, which could potentially impact accuracy. To facilitate the tokenization process, we utilized the Tokenizer function from the TensorFlow library. Tokenization involves breaking down a piece of text into smaller parts, which can be characters, words, or sub-words. For word tokenization, we opted for the default delimiter characters provided by the TensorFlow Tokenizer function. We conducted several tests with different settings, and the most favorable outcomes were obtained when the maximum number of features was set to 10,000. Drawing from existing research and our own experiments, we made the decision to exclude comments exceeding 150 characters. This was done to mitigate the impact of lengthy comments on the processing and training time of the algorithms.

Data split:

The data was divided into three sections: training, validation, and test. The test section includes only 100 text samples, which are then combined with images using various fonts. This division is carried out in a stratified manner to ensure an equal representation of toxic and non-toxic comments. The remaining data is split in an 80:10:10 ratio for training, validation, and test sets, as our research has shown that this setup produces the best outcomes. As a result, the processed data used in the experiments consists of 20,942 training samples, 2,617 validation comments, and 2,617 texts for testing. Each section contains an equal number of toxic and non-toxic comment samples.

Model selection:

The proposed method involves using Recurrent Neural Networks (RNN) for the task. The initial RNN type considered is Long short-term memory (LSTM), which is widely used for processing sequential data and achieves excellent performance. LSTM was specifically designed to address the problems of vanishing and exploding gradient in RNN networks by introducing an additional output cell with four gates. The forget gate determines which data should be discarded from memory units, while the input gate decides which data should be accepted. The update gate updates the memory, and the output gate returns the new long-term memory. Another RNN that will be used is the Gated Recurrent Unit (GRU), which is similar to LSTM but has only two gates: reset and update. The reset gate combines inputs with previous data, while the update gate determines how much of the previous memory should be retained. GRU has a simpler architecture, resulting in faster training times. Although its accuracy is usually slightly lower than LSTM on certain datasets, GRU may outperform LSTM in some cases. Both models achieve excellent performance in time series and NLP data, so it is expected that they will have similar accuracy scores. While traditional RNN networks learn from left to right, recent advancements allow for training in both directions. This type of network considers two sequences, with the input being fed in both forward and backward directions. Essentially, two models are trained simultaneously. This approach provides additional context and enables the model to learn more from the combined information. Bi-directional RNNs excel in speech recognition and NLP, making them a potentially better solution for this task. Figure 2 illustrates the architecture of a Bidirectional RNN that can be applied to both LSTM and GRU networks.

Evaluation:

The performance of each classifier is assessed using four key metrics: accuracy, precision, recall, and F1-score.

Accuracy is the proportion of correct predictions made by the model, providing an overall indication of its performance. However, due to class imbalances, other metrics should also be considered as accuracy can be negatively affected. The formula for accuracy is shown in equation

1. Where True Positive (TP) represents correct positive predictions, True Negative (TN) represents correct negative predictions, False Positive (FP) represents incorrect positive predictions, and False Negative (FN) represents incorrect negative predictions Precision, as defined in equation.
2. Focuses on the number of correctly predicted positive values, which is crucial in binary detection tasks like identifying toxic messages without misclassifying normal ones.
3. Measures the proportion of actual positive values correctly identified by the model. F1-score calculated using equation.
4. Combines precision and recall into a single metric that balances both aspects. During model training, performance was evaluated based on training accuracy, loss, validation loss, and validation accuracy.

IX. FUTURE ENHANCEMENT

We plan to use similar strategies to infer BL rules and FRs in the future. We can enhance the framework in the future to implement this approach in a variety of languages with higher accuracy. Also included is the semi-supervised technique to un-labeled data analysis.

X. CONCLUSION

The effectiveness of various machine learning models in classifying toxic comments is assessed in this study, along with the introduction of a new ensemble approach called Lstm-Cnn. The study conducts extensive experiments to examine the influence of imbalanced and balanced datasets on the performance of the models, using random under-sampling and oversampling techniques. Two feature extraction methods, TF-IDF, are utilized to obtain feature vectors for training the models. The results indicate that the models perform poorly on imbalanced datasets but show a significant improvement in classification accuracy when balanced datasets are used.

REFERENCES

[1] E. Aboujaoude, M. W. Savage, V. Starcevic, and W. O. Salame, ‘‘Cyberbullying: Review of an old problem gone viral,’’ J. Adolescent Health, vol. 57, no. 1, pp. 10–18, Jul. 2015.

[2] How Much Data is Created on the Internet Each Day? Accessed: Jun. 6, 2020. [Online]. Available:https://blog.microfocus.com/howmuch data-iscreated- onthe-internet-each-day/

[3] World Internet Users and 2020 Population Stats.Accessed:Jun.6,2020.[Online].Available:https://www.intern etworldstats.com/stats.htm.

[4] M. Duggan, ‘‘Online harassment,’’ Pew Res. Center, Washington, DC, USA, Tech. Rep., 2014.[Online].Available:https://www.pewresearch.org/internet/wpcontent/uploads/sites/9/2017/07/PI_2017.07.11_OnlineHarassment_FINAL.pdf

[5] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, ‘‘Deep learning for hate speech detection in tweets,’’ in Proc. 26th Int. Conf. World Wide Web Companion, 2017, pp. 759– 760.

[6] Man Jailed for 35 years in Thailand for Insulting Monarchy on Facebook. Accessed: Jun.6,2020.[Online].Available:https://www.theguardian. com/world/2017/jun/09/man-jailed-for-35-years in Thailand for insulting monarchy-on-facebook.