Enhanced Feature-Based Approach for Hate Speech and Offensive Language Detection through NLP Techniques

KEERTHI PATNAIK, University of Texas at Arlington
PREETHI SUBRAMANIAN, University of Texas at Arlington

Identifying and combating online hate speech is crucial for fostering inclusive and secure digital environments, ensuring the well-being and safety of all users. In this paper, we assess the efficacy of various machine learning (ML) models in classifying texts using Natural Language Processing (NLP) techniques and identifying hate speech. The process involves analyzing a dataset, performing text pre-processing, feature engineering, and evaluating various machine learning classification algorithms. Key features such as TF-IDF scores, Sentiment analysis, Doc2vec, and Word2vec vector columns are extracted and compared to enhance the classification models. We evaluated the performance of traditional ML algorithms such as Logistic Regression, Support Vector Machines (SVM), Naive Bayes, Random Forest, and Adaboost. The performance of each algorithm is evaluated using metrics such as accuracy, precision, recall, and F1 score. The results show that differentiating hate speech from offensive language is challenging, but the proposed features provide valuable insights for detecting toxic language online. The project highlights the importance of robust feature extraction methods and addresses existing challenges in hate speech detection on social media. Among the ML models, our experiments show that Adaboost performed well being consistent across all the features with the F1-score of 90.14% when combined with TF-IDF vectorization and other features.

CCS Concepts: • **Computing methodologies → Machine learning; Supervised learning by classification**.

Additional Key Words and Phrases: Hate Speech, Text Classification, Natural Language Processing, Machine Learning, Data Preprocessing, Feature Extraction

ACM Reference Format:
Keerthi Patnaik and Preethi Subramanian. 2024. Enhanced Feature-Based Approach for Hate Speech and Offensive Language Detection through NLP Techniques. 1, 1 (April 2024), 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The project focuses on the challenging task of automatically detecting hate speech on social media, specifically distinguishing hate speech from other instances of offensive language [7]. The significance of this project within the domain of Natural Language Processing (NLP) lies in the need to accurately differentiate between hate speech and offensive language, as hate speech can have serious legal and moral implications. The project builds upon existing work [2], aiming to improve the performance and analysis of hate speech detection. This study utilizes a publicly available dataset provided by CrowdFlower containing hate speech keywords, which are then labeled into three categories: hate speech, offensive language, and neither. By training a multi-class classifier to distinguish between these categories, the project aims to improve the accuracy of hate speech

Authors' addresses: Keerthi Patnaik, University of Texas at Arlington, Arlington, Texas, kxp9181@mavs.uta.edu; Preethi Subramanian, University of Texas at Arlington, Arlington, Texas, pxs9233@mavs.uta.edu.

XXXX-XXXX/2024/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

detection. The results highlight the challenges and nuances involved in accurately classifying hate speech, emphasizing the importance of context and the heterogeneity of hate speech usage [6]. This project contributes to the ongoing efforts in NLP to address the detection and classification of hate speech, providing insights into the complexities of identifying and differentiating between hate speech and offensive language in online communication.

The project proposal aims to address the challenge of distinguishing hate speech from offensive language on social media. The methodology involves using a crowd-sourced hate speech lexicon dataset, which is then labeled into three categories: hate speech, offensive language, or neither. Machine learning models are trained to differentiate between these categories using various features extracted from the tweets.

Overall, this project contributes to the field of NLP by offering a valuable resource for detecting toxic language online and addressing the problem of hate speech in social media. It underscores the significance of utilizing NLP techniques and machine learning algorithms to tackle issues related to harmful content dissemination on digital platforms.

The rest of this paper is organized as follows. In Section 2, we defined the problem statement that the paper aims to address and discuss existing research gaps. In Section 3 we review the related work on hate speech detection and highlight existing approaches' strengths and weaknesses. In Section 4, we describe our datasets and the data preprocessing steps we took for our experiment. In Section 5, we describe our proposed approach of NLP techniques and feature enhancements for text classification models for hate speech detection. We evaluate the models' performance in Section 6 leading to a discussion of result interpretation and findings in Section 7. Finally, in Section 8, we conclude the paper and discuss future directions and advancements for research on hate speech detection.

2 MOTIVATION

The project aims to address the challenge of automatically detecting hate speech on social media platforms, specifically focusing on the differentiation between hate speech and other forms of offensive language. The research question revolves around how to effectively separate hate speech, which targets disadvantaged social groups in a harmful manner, from offensive language that may be offensive but not necessarily hateful.

Existing literature and research in this area have highlighted the prevalence of toxic comments on social media platforms and the negative impact they can have on users. Various studies have proposed different approaches and techniques, such as using ensemble models, deep learning algorithms, and data re-sampling methods, to classify toxic comments effectively [9]. However, there are still research gaps in terms of improving the accuracy and efficiency of toxic comment classification models, especially when dealing with imbalanced datasets and subtle nuances in language that can indicate toxicity. Some of the research gaps from the existing literatures in this area include:

- Conflation of Hate Speech and Offensive Language: Many studies tend to conflate hate speech with offensive language, leading to challenges in accurately identifying and distinguishing between the two.
- Lexical Detection Challenges: Lexical detection methods have low precision as they may classify all messages containing specific terms as hate speech, even if they are not [4].
- Supervised Learning Limitations: Previous work using supervised learning has failed to effectively differentiate between hate speech and offensive language, highlighting the need for more nuanced approaches [5].

- Contextual Understanding: Understanding the context in which certain words or phrases are used is crucial for accurately identifying hate speech, as language can be used in various ways that may not always be hateful [11].
- Social Biases in Algorithms: There is a need to address and correct social biases that may enter algorithms used for hate speech detection, ensuring fair and accurate classification [3].
- Subjectivity in Hate Speech: There is a need to address and correct social biases that may enter algorithms used for hate speech detection, ensuring fair and accurate classification.

By addressing these gaps and challenges, the project aims to contribute to the development of more accurate and context-aware hate speech detection systems that can effectively differentiate between hate speech and offensive language on social media platforms.

3 LITERATURE REVIEW

In this section, we discuss related work on hate speech detection and highlight existing approaches' strengths and weaknesses. The paper "Automated Hate Speech Detection and the Problem of Offensive Language" [2] highlights the challenge of separating hate speech from other instances of offensive language. It discusses how lexical detection methods tend to have low precision, as they may classify all messages containing particular terms as hate speech. The paper discusses the lack of a formal definition of hate speech but notes a consensus that it targets disadvantaged social groups in a potentially harmful manner. The paper outlines the features used for classification, including linguistic and sentiment features. It also discusses the models tested for hate speech detection, such as logistic regression, naive Bayes, decision trees, random forests, and linear SVM.

The paper "Detecting Hate Speech in Social Media" [7] discusses previous studies on abusive language detection, including research on cyber-bullying, hate speech detection, and racism detection in user-generated content. The paper describes the Hate Speech Detection dataset used in the experiments and outlines the computational approach, features, and evaluation methodology. The authors applied a linear Support Vector Machine (SVM) classifier and used character n-grams, word n-grams, and word skip-grams as features to establish a lexical baseline for discriminating between hate speech and profanity.The results of the experiments showed that the main challenge lies in discriminating between profanity and hate speech. The best accuracy achieved was 78% using a character 4-gram model. The paper also compares the results against a majority class baseline and an oracle classifier to establish the upper limit of performance for the dataset.

The journal "Classification of social media Toxic comments using Machine learning models" [10] discusses the importance of identifying toxic comments on social media platforms and the challenges faced in maintaining a positive online environment. It highlighted the use of Natural Language Processing (NLP) with Deep neural networks to classify toxic comments. The study focused on techniques like Tokenizing, Stemming, and Embedding to classify online comments based on their toxicity levels. The results of the article included the proposal of a neural network model for classifying comments and comparing its accuracy with other models like Long Short Term Memory (LSTM) and Convolutional Neural Network. The study utilized word embeddings in conjunction with recurrent neural networks to achieve high accuracy in toxic comment classification.

The article "Systematic literature: Toxic comment classification" [8] explores the effectiveness of deep learning models compared to machine learning models in classifying toxic comments to combat cyberbullying. The study analyzes the most common algorithms and datasets used by researchers over the past five years. The findings reveal that Long Term Short Memory (LSTM) is the most frequently mentioned deep learning model, consistently achieving high accuracy results above 79% with around 9000 data samples. While some researchers have experimented with hybrid models combining multiple algorithms, these hybrids may not always outperform individual models. Although, Deep learning models perform well, there are few limitations as they can be computationally expensive and require a large amount of data for training. They may also be prone to overfitting if not properly regularized.

The paper "Evaluation of Different Machine Learning, Deep Learning and Text Processing Techniques for Hate Speech Detection" [13] evaluates the performance of various machine learning (ML) and deep learning (DL) models for detecting hate speech on three different datasets. Traditional ML algorithms like SVM, Naive Bayes, Decision Trees, Random Forests, and Logistic Regression are compared with DL models like Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM), and the BERT pre-trained transformer model. The results show that BERT outperformed all other models, followed by CNN and LSTM, with SVM performing best among the traditional ML models. The paper provides a comprehensive evaluation of both traditional ML algorithms and deep learning models for hate speech detection. This allows for a thorough comparison of the performance of different approaches.The existing approaches to hate speech detection suffer from several weaknesses, including imbalanced datasets, limited exploration of feature selection techniques, and a lack of discussion on model interpretability. These issues can lead to biased results and hinder the development of effective models for detecting hate speech. Among the traditional ML models as mentioned in this paper, SVM performed best with the highest F1-score of 75.6%.

4 METHODOLOGY

In this section, we briefly present the dataset from Crowdflower that we used for our experiment. Additionally, we go over the preprocessing, feature engineering and classification models we implemented to make the data suitable for training and testing. A minimum of three annotators have annotated 14,509 English tweets in the sample. The individuals responsible for the annotation of this dataset were requested to assign a class to each tweet and provide an annotation for each one as shown below:

- **Hate**- Contains hate speech.
- **Offensive** - Contains offensive language but no hate speech.
- **Neither**- Contains no offensive content at all.

Each instance in this dataset contains the text of a tweet along with one of the three aforementioned labels. The distribution of the texts across the three classes is shown in Table 1.

Table 1. Distribution of tweets and categories in the Hate Speech Dataset

Category	Text Count
Total tweet	24783
Hate Speech Tweet	1430
Offensive Tweet	19190
Neither Tweet	4163

4.1 Text Pre-processing

The collected data undergoes text pre-processing techniques to clean and prepare the text for analysis. This involves steps like removing punctuation, tokenizing the text, removing stopwords, stemming, and eliminating URLs and mention names.

4.2 Feature Engineering

After pre-processing, the text data is passed through a feature engineering stage. Various features are extracted from the text data, including:

- TF-IDF weights: These are used to represent the importance of words in the text.
- Sentiment polarity scores: These scores indicate the sentiment (positive, negative, neutral) of the text.
- Doc2Vec vector columns: These are used for representing the text in a vector space.
- Word2Vec vector columns: These are used as word-level embeddings and helps in identifying the granularity of text.

4.3 Classification Models

Different classification algorithms are applied to the feature sets to classify the text into categories like hate speech, offensive language, or neither. The algorithms used include: Logistic Regression that works consistently well with most feature sets. Random Forest shows good performance, especially with TF-IDF scores included. Naive Bayes performs less significantly compared to other algorithms. SVM shows consistent performance across different feature sets. Adaboost performs consistently well among the traditional ML models.

4.4 Evaluation

The classification models are evaluated based on accuracy and F1-scores for different feature sets. The results are analyzed to understand the performance of each algorithm and feature set in classifying hate speech.

4.5 Modifications and Refinements

Throughout the project, modifications and refinements are made based on the analysis of results. The project aims to improve existing works in hate speech detection by identifying gaps and proposing solutions to enhance classification accuracy.

4.6 Frameworks and Tools

The implementation of the project involves using Python programming language along with libraries such as scikit-learn for machine learning algorithms, NLTK for natural language processing tasks, and Jupyter Notebook for code development and analysis.

Overall, the project follows a systematic approach from data collection to model evaluation, with a focus on improving hate speech detection using NLP techniques and machine learning algorithms. The methodology includes thorough analysis, feature engineering, and model evaluation to achieve the project's objectives.

5 EXPERIMENTAL SETUP

In this section, we go over the experiment setup, feature extraction, and classifiers (ML models) we used in our research, as well as the evaluation metrics.

The data for hate speech detection in social media was collected from a publicly available dataset provided by CrowdFlower. The dataset was used to train and test the hate speech detection model. Text preprocessing techniques were applied to clean the dataset before training the model. Preprocessing steps included removing punctuation, tokenizing, removing stopwords, stemming, and eliminating URLs and mention names. Various features were extracted from the preprocessed text data, including TF-IDF vectorization, sentiment analysis, Doc2vec vector columns, and Word2vec vector columns. Different sets of features were created and used for training and testing the classification models. Machine learning classification algorithms such as Logistic Regression, Random Forest, Naive Bayes, SVM and Adaboost were used for training the hate speech detection model. The models were trained on the extracted features and evaluated based on accuracy and F1-scores.The experimental setup likely required a machine with sufficient computational resources to handle the training of machine learning models. Software tools such as Python, Jupyter Notebook, and relevant libraries for natural language processing (NLP) and machine learning were used. The classification models' performance was evaluated using metrics like accuracy and F1-scores. Precision, recall, and F1-scores were calculated for each class (hate, offensive, neither) to assess the model's performance in detecting hate speech and offensive language. The experiments involved comparing the performance of different models trained on various feature combinations. This comparative analysis helped in identifying the most effective combination of features and models for hate speech detection.

Overall, the experimental setup involved data collection, preprocessing, feature extraction, training and testing of machine learning models, and evaluation of model performance using appropriate metrics. The hardware and software configurations likely included standard machine learning tools and libraries, and the evaluation was based on accuracy and F1-scores to assess the hate speech detection model's effectiveness.

6 RESULT AND ANALYSIS

In this section, we discuss the results of our experiments. The evaluation results from the five models on Dataset are shown in Table 2. Based on the evaluation results of different models trained on various combinations of features in classifying hate speech, offensive language, and neutral comments on the dataset, we can provide the following result analysis:

Table 2. Presenting the results obtained from experiments conducted during the project

Feature or Models	Logistic Regression	Random Forest	Naive Bayes	Linear SVC	Adaboost
TF-IDF	89.77	90.52	64.92	89.27	90.05
TF-IDF + Sentiment Analysis	89.83	89.37	65.02	89.17	90.01
TF-IDF + Sentiment Analysis + Doc2Vec	90.01	88.99	65.02	89.27	89.67
TF-IDF + Sentiment Analysis + Doc2Vec + Enhanced Features	81.34	88.18	66.25	89.49	90.05
TF-IDF + Sentiment Analysis + Doc2Vec + Enhanced Features + Word2Vec	85.19	88.62	66.25	83.86	90.14

Logistic Regression achieved an accuracy score ranging from 85.19% to 90.01% across different feature combinations. It consistently performed well in classifying the data, especially when combined with multiple features like Sentiment Analysis, Doc2Vec, Enhanced Features, and Word2Vec. Random Forest also showed good performance with accuracy scores ranging from 88.18% to 90.52%.It demonstrated stable performance across different feature combinations, making it a reliable choice for classification tasks. Naive Bayes had the lowest accuracy scores among the models, ranging from 64.92% to 66.25%.It struggled to classify the data accurately compared to other models, indicating limitations in handling complex relationships within the data.Linear SVC achieved accuracy scores between 83.86% and 89.49% across different feature combinations.It showed consistent performance but slightly lower accuracy compared to Logistic Regression and Random Forest. AdaBoost performed well with accuracy scores ranging from 89.67% to 90.14%.It showed competitive performance, especially when combined with multiple features, making it a strong contender for classification tasks. AdaBoost model's performance was consistently better for all different combinations for features when compared with other models.

Overall, the AdaBoost and Random Forest models stand out as top performers in terms of accuracy and overall performance across different feature combinations. Adaboost model's performance was consistent across all the combinations of features when experimented on different classifiers. Logistic Regression and Linear SVC also demonstrate stable and reliable performance. Naive Bayes, on the other hand, shows lower accuracy and performance metrics compared to other models. It is essential to consider the trade-offs between accuracy, precision, recall, and F1-score when selecting the best model for the hate speech detection task.

Since AdaBoost's performance was overall better, we analyzed its accuracy using confusion matrix as shown in Fig 1. The confusion matrix shows the proportions of correctly and incorrectly classified instances relative to the total number of predictions.

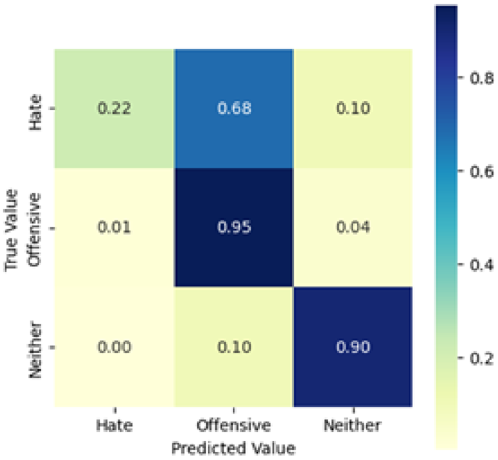

Fig. 1. Adaboost Model: True Category Vs Predicted Category

Based on the colors and the values provided in each cell of the matrix, we can see that the

- **Top-left cell (Hate, Hate)**: This represents the True Positive (TP) rate for "Hate speech". Here, 22% of all predictions were correctly identified as "Hate speech".
- **Middle cell (Offensive, Offensive)**: This is the TP rate for "Offensive language", showing that 95% of "Offensive language" was correctly identified, which is quite high.
- **Bottom-right cell (Neither, Neither)**:This indicates the TP rate for "Neither" category, with 90% correctly identified
- **Top-left cell (Hate, Hate)**: This represents the True Positive (TP) rate for "Hate speech". Here, 22% of all predictions were correctly identified as "Hate speech".
- The off-diagonal cells represent the errors
- **Top-left cell (Hate, Hate)**: This represents the True Positive (TP) rate for "Hate speech". Here, 22% of all predictions were correctly identified as "Hate speech".
- **Middle row (Offensive, Hate and Offensive, Neither)**: This shows that only 1% of "Offensive language" was misclassified as "Hate speech" and 4% as "Neither".
- **Bottom row (Neither Hate and Neither, Offensive)**: This indicates that "Neither" was misclassified as "Hate speech" 0% of the time and as "Offensive language" 10% of the time.

The color gradient (from blue to light green) indicates the percentage of predictions in each category, with darker shades typically representing higher percentages or frequencies. In this matrix, we can see that most "offensive language" instances are correctly classified (dark blue), while there is considerable confusion between "Hate speech" and "Offensive language" (lighter blue in "Hate, Offensive" cell).

The model is best at identifying “Offensive language” correctly. There is significant confusion between “Hate speech” and “Offensive language”, with more “Hate speech” being classified as “Offensive language” than correctly identified. “Neither” is mostly classified correctly, but there is some confusion with “Offensive language”.

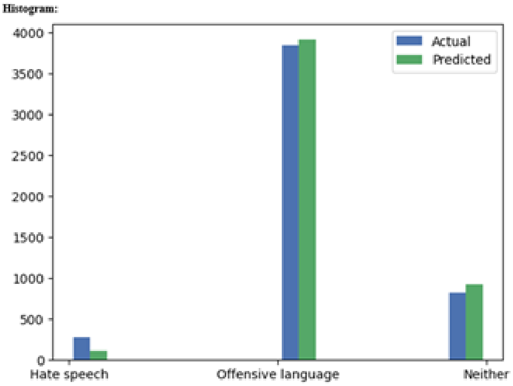

Fig. 2. Histogram: Actual Target Vs. Predicted Target

Finally, we also analyzed the histogram plot for actual target vs predicted target as shown in Fig 2. The discrepancies in the height of the bars between the actual and predicted histograms point out overprediction and underprediction issues. The model overpredicts “Offensive language” and “Neither” as indicated by a higher bar in the prediction’s histogram than in the actual distribution. It seems to underpredict “Hate speech” based on a lower bar in the prediction’s histogram.

7 DISCUSSION

In this section, we discuss the results of our experiments and their interpretation

- Hate Speech: With only 22% correctly identified, the model struggles to identify this category.
- Offensive Language: The high TP rate (95%) indicates that AdaBoost performs very well with this category.
- Neither: An 90% TP rate is quite good, though there is some room for improvement.
- False Negatives and False Positives: There is a substantial misclassification of "Hate speech" as "Offensive language" (68%) and to a lesser extent as "Neither" (10%). "Offensive language" misclassified as other categories is very low (1% as "Hate speech" and 4% as "Neither"). "Neither" is mostly well-identified, but there is a 10% rate of being misclassified as "Offensive language," which suggests some features overlap between these two categories in the model’s decision-making process.

7.1 Implications

From the results of hate speech detection in social media, we can see that incorporating various features such as TF-IDF, sentiment analysis, Doc2Vec, enhanced readability features, and Word2Vec has led to improvements in the accuracy of the classification models. Also, using AdaBoost slightly improves the model’s performance. The results highlight the importance of incorporating diverse features and techniques for hate speech detection in social media, as it is a complex and context-dependent task. The findings suggest that a combination of linguistic, sentiment, and semantic features can improve the accuracy and robustness of hate speech classification models.

The implications extend to the broader field of natural language processing and social media analysis, emphasizing the need for nuanced approaches to handle sensitive and harmful language online.

7.2 Strengths

The use of multiple features has enhanced the performance of the models, with AdaBoost consistently achieving the highest accuracy across different feature combinations. The inclusion of sentiment analysis and readability features has provided additional context and linguistic information to the models, improving their ability to classify hate speech accurately. The combination of different feature sets has allowed for a more comprehensive analysis of the text data, capturing both semantic and syntactic information effectively.

7.3 Weaknesses

The Naive Bayes model consistently performed the worst among the models, indicating that the assumption of independence between features may not hold true for hate speech classification. The Linear SVC model also showed lower accuracy compared to other models, suggesting that the linear separation of classes may not be optimal for this task. The models may still struggle with the nuances and subtleties of hate speech language, leading to misclassifications or inaccuracies in certain cases.

7.4 Findings and Challenges

The lower performance of Naive Bayes and Linear SVC models was unexpected and indicates the limitations of these traditional classification algorithms for hate speech detection. Challenges were encountered in balancing the trade-off between model complexity and interpretability, and in optimizing the hyperparameters for each model and feature combination.

7.5 Future Work

Considering the potential avenues for future research and improvements based on the project outcomes, Future research could focus on exploring more advanced deep learning models such as LSTM or BERT for hate speech detection, which may capture the contextual nuances better. Investigating the impact of different pre-processing techniques, feature engineering methods, and model ensembling strategies could further enhance the performance of hate speech classification systems.

8 CONCLUSION

Based on the evaluation of multiple models trained on hate speech detection using different feature combinations, we can see that Adaboost outperformed all other models, including Logistic Regression, Random Forest, Naive Bayes, and Linear SVC, in terms of accuracy across different feature sets and models. Adaboost consistently achieved the highest accuracy scores compared to the other models when trained on various feature sets, including TF-IDF, TF-IDF with Sentiment Analysis, TF-IDF with Sentiment Analysis and Doc2Vec, TF-IDF with Sentiment Analysis, Doc2Vec, and Enhanced Features, and TF-IDF with Sentiment Analysis, Doc2Vec, Enhanced Features, and Word2Vec. Adaboost showed better performance in terms of precision, recall, and F1-score for all three classes (Hate, Offensive, Neither) compared to the other models. The confusion matrix visualization for Adaboost also indicates a better distribution of predicted values across the true values, showing a more balanced and accurate classification. Therefore, based on the provided evaluation results and conclusion, it is evident that Adaboost consistently outperformed all other models in terms of accuracy and overall performance across different feature sets and classification tasks.

The main findings and contributions of this paper are as follows:

- **Development of a hate speech detection model**: The project involves the creation of a machine learning model using Python to detect hate speech in social media data. This model can analyze text data and classify it as hate speech or non-hate speech based on certain features and patterns.
- **Evaluation of model performance**: The project evaluates the performance of the hate speech detection model using metrics such as accuracy, precision, recall, and F1 score. This helps in understanding how well the model can identify hate speech instances in social media content.
- **Implementation of natural language processing techniques**: The project utilizes natural language processing techniques such as tokenization, word embeddings, and text preprocessing to enhance the model's ability to detect hate speech effectively.

In today's digital age, social media platforms are increasingly being used as channels for spreading hate speech, which can have harmful consequences on individuals and communities. By developing an automated hate speech detection model, this project offers a proactive approach to identifying and mitigating hate speech online. This not only contributes to the advancement of NLP research but also has practical implications for online platforms and social media companies looking to implement effective content moderation systems [15]. This can help promote a safer and more inclusive online environment, ultimately contributing to the larger goal of combating online hate speech and fostering positive interactions on social media platforms. This project is significant in advancing knowledge and addressing real-world challenges in the field of Natural Language Processing (NLP) as follows:

- **Advancing Knowledge**: By developing a hate speech detection model, this project contributes to the advancement of knowledge in natural language processing. It involves the application of machine learning algorithms and text processing techniques to identify and classify hate speech in social media data. This research can lead to a better understanding of the language patterns and characteristics of hate speech, which can further enhance the development of more sophisticated algorithms for text analysis [12, 14].
- **Addressing Real-World Challenges**: Hate speech is a prevalent issue in social media platforms, leading to harmful consequences such as cyberbullying, discrimination, and incitement to violence. By creating a hate speech detection system, this project aims to address the real-world challenge of combating online hate speech. Detecting and filtering out hate speech can help create a safer and more inclusive online environment for users, thereby contributing to the promotion of positive social interactions and the prevention of online harassment [1].
- **Ethical Considerations**: Detecting hate speech and offensive language is crucial for maintaining ethical standards in online communication. By developing accurate and efficient hate speech detection systems, we can promote responsible and respectful online interactions.
- **Technological Advancement**: The project utilizes NLP techniques and machine learning algorithms to analyze and classify text data. By implementing and optimizing these models, we contribute to the advancement of NLP technology and its applications in real-world scenarios.

9 DEMO VIDEO AND SOURCE CODE

- **Demo Video link**: https://mavsuta-my.sharepoint.com/:v:/g/personal/kxp9181_mavs_uta_edu/EYo1yTBTpIREs0eAccZpWJEB5LhH3g2Yb3xeXVK0NGg29A?e=LwYzUu
- **Source Code**: https://github.com/IamPreethi-S/Enhanced-Feature-Based-Approach-for-Hate-Speech-Detection-through-NLP-Techniques

REFERENCES

[1] Mohammed Ali Al-Garadi, Mohammad Rashid Hussain, Nawsher Khan, Ghulam Murtaza, Henry Friday Nweke, Ihsan Ali, Ghulam Mujtaba, Haruna Chiroma, Hasan Ali Khattak, and Abdullah Gani. 2019. Predicting cyberbullying on social media in the big data era using machine learning algorithms: review of literature and open challenges. *IEEE Access* 7 (2019), 70701–70718.

[2] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. 11, 1 (2017), 512–515.

[3] Mai Elsherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth M. Belding-Royer. 2018. Hate Lingo: A Target-based Linguistic Analysis of Hate Speech in Social Media. (2018). https://api.semanticscholar.org/CorpusID:4809781

[4] Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. 2015. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering* 10, 4 (2015), 215–230.

[5] Harsh Kajla, Jatin Hooda, Gajanand Saini, et al. 2020. Classification of online toxic comments using machine learning algorithms. (2020), 1119–1123.

[6] György Kovács, Pedro Alonso, and Rajkumar Saini. 2021. Challenges of hate speech detection in social media: Data scarcity, and leveraging external resources. *SN Computer Science* 2, 2 (2021), 95.

[7] Shervin Malmasi and Marcos Zampieri. 2017. Detecting Hate Speech in Social Media. (11 2017), 467–472. https://doi.org/10.26615/978-954-452-049-6_062

[8] Felix Museng, Adelia Jessica, Nicole Wijaya, Anderies Anderies, and Irene Anindaputri Iswanto. 2022. Systematic literature review: Toxic comment classification. (2022), 1–7.

[9] Kiran Babu Nelatoori and Hima Bindu Kommanti. 2023. Multi-task learning for toxic comment classification and rationale extraction. *Journal of Intelligent Information Systems* 60, 2 (2023), 495–519.

[10] K Poojitha, A Sai Charish, M Reddy, and S Ayyasamy. 2023. Classification of social media Toxic comments using Machine learning models. *arXiv preprint arXiv:2304.06934* (2023).

[11] Anna Schmidt and Michael Wiegand. 2017. A Survey on Hate Speech Detection using Natural Language Processing. (2017). https://api.semanticscholar.org/CorpusID:9626793

[12] Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. (2017), 1–10.

[13] Nabil Shawkat. 2023. Evaluation of Different Machine Learning, Deep Learning and Text Processing Techniques for Hate Speech Detection. (2023).

[14] Chrysoula Themeli, George Giannakopoulos, and Nikiforos Pittaras. 2019. A study of text representations for Hate Speech Detection. (2019), 424–437.

[15] Kehan Wang, Jiaxi Yang, and Hongjun Wu. 2021. A survey of toxic comment classification methods. *arXiv preprint arXiv:2112.06412* (2021).