VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi : 590018

A Mini-Project(21CSMP67) Synopsis on

"AI-Based Tool for Detecting Vulgar Content Online"

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering
in
Computer Science and Engineering

by

NISHKA KUMAR	**1AY21CS120**
RAUNAK PRIYADARSHI YADAV	**1AY21CS148**
NAVNEET UJJWAL	**1AY21CS115**
KRISHNANDAN PANDIT	**1AY21CS081**

Under the guidance of
Mr. Vinayak Raju Kage
Designation

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

ACHARYA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belgaum)

2023-2024

ABSTRACT

This project presents an AI-driven tool for detecting and moderating vulgar content on social media. By leveraging advanced natural language processing (NLP) techniques and deep learning models, the system identifies offensive language and imagery with high accuracy. The methodology includes data preprocessing, feature extraction, and model training using annotated datasets, followed by deployment as an API for real-time content moderation. To ensure fairness and accuracy, the system integrates user feedback and bias mitigation mechanisms. This tool enhances the efficiency and scalability of content moderation, addressing ethical considerations such as transparency and privacy. Ultimately, it aims to create safer, more inclusive digital communities by providing a robust and reliable solution for managing offensive content.

CHAPTER 1

INTRODUCTION

1.1. PROBLEM DEFINITION:

The objective of creating an AI based tool for the removal of offensive and vulgar content online is to avoid the human biases and labeling performed and to reduce the psychological effect it can cause people.

Automatic System: It aims to create an automatic system that can perform easy classification and perform unbiased decision without the requirement of manual work.

Reduction of psychological effect: Offensive and vulgar content can cause psychological effects on vulnerable groups of people in society. To avoid this factor the AI-based tool is used to ensure that these contents are never seen or uploaded on the internet.

Support for human activity: Creating this type of AI based tool provides a helping hand for people to perform and complete their daily tasks.

1.2. OVERVIEW OF THE TECHNICAL AREA:

Data Collection and Preprocessing:

The first critical phase in developing an AI tool for detecting vulgar content is data collection and preprocessing. This involves identifying and sourcing a diverse dataset from social media platforms to ensure comprehensive coverage of various types of content, including both vulgar and non-vulgar posts. Once collected, the data undergoes a meticulous labeling process where each piece of content is annotated as offensive or acceptable, which may involve manual labeling or using pre-existing annotated datasets. Following this, data cleaning procedures are implemented to remove noise and irrelevant information. This step includes normalizing text data by converting it to lowercase, removing special characters, tokenization, and eliminating stop words. Properly preprocessing the data is crucial as it prepares the dataset for effective feature extraction and model training, ensuring that the machine learning algorithms can accurately learn and predict patterns indicative of vulgar content.

Machine Learning Model Development:

The next step is the development of the machine learning model. This phase involves selecting appropriate machine learning algorithms that are well-suited for text classification tasks, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), or transformer-based models like BERT. These models are trained on the preprocessed and labeled dataset to learn the distinguishing features of vulgar content. The training process involves optimizing hyperparameters and fine-tuning the model to enhance performance. Rigorous evaluation metrics such as accuracy, precision, recall, and F1-score are used to assess the model’s performance and ensure it can effectively identify offensive content while minimizing false positives and negatives. This phase is iterative, involving continuous refinement and validation to improve the model’s robustness and reliability in detecting vulgar content.

Implementation:

The final phase involves the practical implementation of the AI tool within a working website. This integration involves deploying the trained machine learning models as an API service that can process user-generated content in real-time. When a user submits a post, the AI tool analyzes it for any vulgar content based on the learned patterns. Depending on the platform’s moderation policies, the tool can automatically flag, remove, or escalate the content for further review by administrators. Additionally, implementing feedback mechanisms allows the system to learn from user interactions and continuously improve its accuracy and effectiveness. This integration not only enhances the scalability and consistency of content moderation efforts but also addresses ethical considerations such as fairness, transparency, and user privacy, ensuring the tool remains adaptive and reliable in managing vulgar content on social media platforms.

1.3. OVERVIEW OF EXISTING SYSTEM:

Facebook’s AI Moderation:

Facebook utilizes advanced AI moderation tools to manage the vast amount of content uploaded to its platform daily. Despite its sophisticated algorithms, Facebook's system has faced challenges with incorrect predictions and bias selection. These issues arise due to the complexity of natural language and the context-specific nature of content, leading to both false positives and negatives. The bias in training data also affects the fairness of moderation, occasionally resulting in unfair treatment of certain groups or topics.

YouTube’s Content ID and AI Moderation:

YouTube employs Content ID and AI-based moderation to identify and manage copyrighted and inappropriate content. However, the diversity and complexity of video content pose significant challenges. The system sometimes makes incorrect predictions and classifications, failing to account for the nuances in video context and meaning. This leads to inappropriate flagging of content or allowing harmful content to slip through, affecting both content creators and viewers.

Twitter’s Automated Moderation:

Twitter's automated moderation system aims to detect and address harmful content rapidly due to the platform's real-time nature. However, the sheer volume of data and the need for instant moderation create difficulties. The system struggles to accurately detect and respond to offensive content in real-time, often missing harmful posts or incorrectly flagging benign ones. This challenge is exacerbated by the need to handle a wide range of content types and languages.

Instagram’s Automated Filters:

Instagram uses automated filters to detect and manage inappropriate content on its platform. While these filters help in maintaining community standards, they are not without flaws. Bias in training data can lead to unfair predictions, with certain content being disproportionately flagged or ignored. The system's reliance on predefined criteria also limits its ability to adapt to new and evolving types of content, resulting in occasional incorrect moderation decisions.

1.4. OVERVIEW OF PROPOSED SYSTEM:

Immediate Detection and Quick Removal:

The proposed system aims to address the limitations of existing moderation systems by focusing on the immediate detection of vulgar and offensive text posts. Utilizing advanced machine learning techniques, the system can analyze user-generated content in real-time. This enables the quick removal of inappropriate posts and the immediate reporting of these posts to administrators. By automating the detection process, the system ensures that offensive content is swiftly managed, minimizing its exposure and impact on users.

Implementation of an Accurate and Unbiased Model:

To enhance the effectiveness of content moderation, the proposed system emphasizes the development and implementation of a highly accurate model that performs unbiased prediction and selection. The model is trained on a diverse and representative dataset to mitigate biases and ensure fairness. By continuously refining the model through feedback loops and regular updates, the system maintains high accuracy in identifying offensive content while minimizing false positives and negatives. This approach not only improves the reliability of content moderation but also fosters a safer and more inclusive online environment.

CHAPTER 2

LITERATURE REVIEW

2.1. DEF:

S.N	PAPER TITTLE & PUBLICATION DETAILS	NAME OF THE AUTHORS	TECHNICAL IDEAS / ALGORITHMS USED IN THE PAPER & ADVANTAGES	SHORTFALLS/DISADVANTAGES & SOLUTION PROVIDED BY THE PROPOSED SYSTEM
1	"Deep Learning for Hate Speech Detection in Tweets“	Zeerak Waseem	Deep Learning Models-CNN for detecting hate speech and offensive language	Data bias, Interpretability, Scalability
2	"Detecting Offensive Language in Social Media Using Machine Learning Techniques“	Stefanos Angelidis, Mirella Lapata	Supervised Learning-Support Vector Machines(SVMs) and Naïve Bayes Classifiers for predictions.	Model Generalization, Feature Extraction Complexity

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1. FUNCTIONAL REQUIREMENTS:

User Registration and Authentication:

The proposed system includes a robust user registration and authentication feature that allows users to create accounts and log in securely. This ensures that only authorized users can access the platform and submit content, maintaining the integrity and security of user interactions.

Content Submission:

Users are enabled to submit various types of content, including text, images, and videos. This feature ensures a comprehensive platform for user-generated content, accommodating diverse forms of expression and communication.

Content Analysis:

The system integrates an AI-driven content analysis module that analyzes submitted content in real-time. This module is designed to detect offensive and vulgar language, ensuring that inappropriate content is identified and managed immediately upon submission.

Moderation Actions:

Upon detecting offensive or vulgar content, the system automatically labels the content accordingly. This automated labeling helps streamline the moderation process, ensuring that potentially harmful content is flagged for further review or immediate action.

Admin Dashboard:

The proposed system features an admin dashboard that offers an intuitive interface for administrators to review flagged content. This dashboard enables admins to efficiently manage moderation actions, review user submissions, and ensure compliance with community guidelines and standards.

3.2. NON-FUNCTIONAL REQUIREMENTS:

Performance:

The proposed system is designed to deliver optimal performance in terms of response time and throughput. The response time, which is the time taken to analyze and label content, is kept minimal to ensure swift moderation actions. Additionally, the system is capable of handling a high volume of content submissions efficiently, maintaining high throughput to accommodate the needs of a large user base.

Reliability:

Reliability is a key focus of the system. It is built to remain operational even in the face of errors or faults. This ensures continuous and uninterrupted service, providing users with a dependable platform for content submission and moderation.

Security:

User data security is paramount in the proposed system. All user data, including account information and content submissions, are securely stored and transmitted. Robust security measures are implemented to protect against unauthorized access and data breaches, ensuring user privacy and data integrity.

Usability:

The system is designed with usability in mind, ensuring that it is easily navigable and understood by users. A user-friendly interface is provided to facilitate seamless interaction, allowing users to effortlessly create accounts, log in, submit content, and navigate the platform.

Accuracy:

Accuracy in detecting offensive and vulgar content is critical for the system’s effectiveness. The AI model is trained to maintain high accuracy, minimizing false positives and negatives. This ensures that inappropriate content is correctly identified and managed, fostering a safer online environment.

3.3. SOFTWARE REQUIREMENTS:

Deployment Requirements:

The proposed system is developed and tested in a Windows operating system environment. This ensures compatibility and ease of development during the initial phases of the project. The system is designed to be flexible enough to be deployed on various operating systems for production use.

Database:

For structured data storage, the system utilizes MySQL. This relational database management system provides efficient storage, retrieval, and management of user data, content submissions, and moderation records. MySQL ensures data integrity and supports the system's scalability needs.

Application Server:

An application server is required to run the application. This server handles client requests, processes them, and delivers the necessary responses. It ensures that the application functions smoothly and efficiently, supporting high user loads and providing a reliable platform for content submission and moderation.

Programming Language:

Python is chosen as the primary programming language for the system. It is used for both AI model development and frontend development. Python's extensive libraries and frameworks make it an ideal choice for building robust and scalable applications.

Frameworks:

Django is employed for both backend and frontend development. This high-level Python web framework enables rapid development and clean, pragmatic design. Django simplifies the creation of complex web applications and provides built-in features for user authentication, database management, and content management.

AI/ML Libraries:

The system leverages TensorFlow for machine learning model training and deployment. TensorFlow provides a comprehensive suite of tools for building and deploying AI models. Additionally, the Natural Language Toolkit (NLTK) is used for natural language processing tasks, enhancing the system's ability to analyze and interpret text data effectively.

3.4. HARDWARE REQUIREMENTS:

CPU: Multi-core processor, preferably Intel i5 or better, to ensure efficient processing of tasks and support for parallel operations.

RAM: At least 16GB to handle multiple processes simultaneously and ensure smooth operation during peak usage times.

Storage: SSD with at least 256GB space to provide fast read/write speeds and sufficient storage for user data, content submissions, and system logs.

GPU: Dedicated GPU to accelerate machine learning model training and deployment, improving performance for tasks involving large datasets and complex computations.

Network: Reliable internet connection to ensure seamless data transfer, user interactions, and real-time content analysis.

3.5. TECHNOLOGIES USED:

Programming Languages:

Python: Used for AI model development, backend development, and some frontend tasks. Python’s extensive libraries and frameworks make it ideal for building robust AI solutions and web applications.

Frameworks:

Django: A high-level Python web framework used for backend and frontend development. Django facilitates rapid development, secure data handling, and seamless integration of various components within the application.

Database:

MySQL: A relational database management system used for structured data storage. MySQL ensures efficient data management and retrieval, supporting the storage needs of user data, content submissions, and system logs.

AI/ML Libraries:

TensorFlow: An open-source machine learning library used for training and deploying machine learning models. TensorFlow provides powerful tools and functionalities to build and optimize models for detecting offensive and vulgar content.

NLTK (Natural Language Toolkit): A library for natural language processing (NLP) tasks. NLTK is used to preprocess text data, tokenize content, and perform various NLP tasks essential for analyzing and understanding textual content.

CHAPTER 4

PROPOSED METHODOLOGY

4.1. OVERVIEW OF PROPOSED METHODOLOGY:

Data Collection and Preparation:

The first step in developing an AI tool for detecting vulgar content is data collection and preparation. This involves gathering a comprehensive dataset of social media posts that includes a diverse range of both vulgar and non-vulgar content. This dataset is crucial for training the machine learning model, as it provides the examples needed for the model to learn to distinguish between acceptable and unacceptable content. Data cleaning and preprocessing are also essential at this stage to ensure that the dataset is free of noise and inconsistencies. This may involve removing duplicates, correcting errors, and standardizing formats.

Feature Extraction:

Once the dataset is prepared, the next step is feature extraction. This involves identifying and extracting meaningful features from the data that can be used by the machine learning model to make predictions. In the context of text data, features could include various linguistic and syntactic properties such as word frequencies, n-grams (combinations of words), part-of-speech tags, and semantic embeddings. These features help the model understand the nuances of the text and improve its ability to detect vulgar content.

Model Selection:

Selecting the appropriate machine learning model is a critical step in the project. Various models can be considered, including traditional models like logistic regression, support vector machines (SVM), and more advanced models like deep learning neural networks (e.g., LSTM, CNN). The choice of model depends on factors such as the size and nature of the dataset, the complexity of the features, and the desired accuracy and performance. It’s important to understand how different models impact the project outcomes and to select the one that best fits the specific requirements of the project.

Training and Evaluation:

The next step is training the machine learning model. This involves splitting the dataset into training and validation sets to evaluate the model's performance. During training, the model learns to associate the extracted features with the corresponding labels (vulgar or non-vulgar). Defining the training parameters, such as learning rate, batch size, and number of epochs, is crucial for optimizing the model's performance. Evaluation metrics like accuracy, precision, recall, and F1-score are used to assess the model's effectiveness and make necessary adjustments.

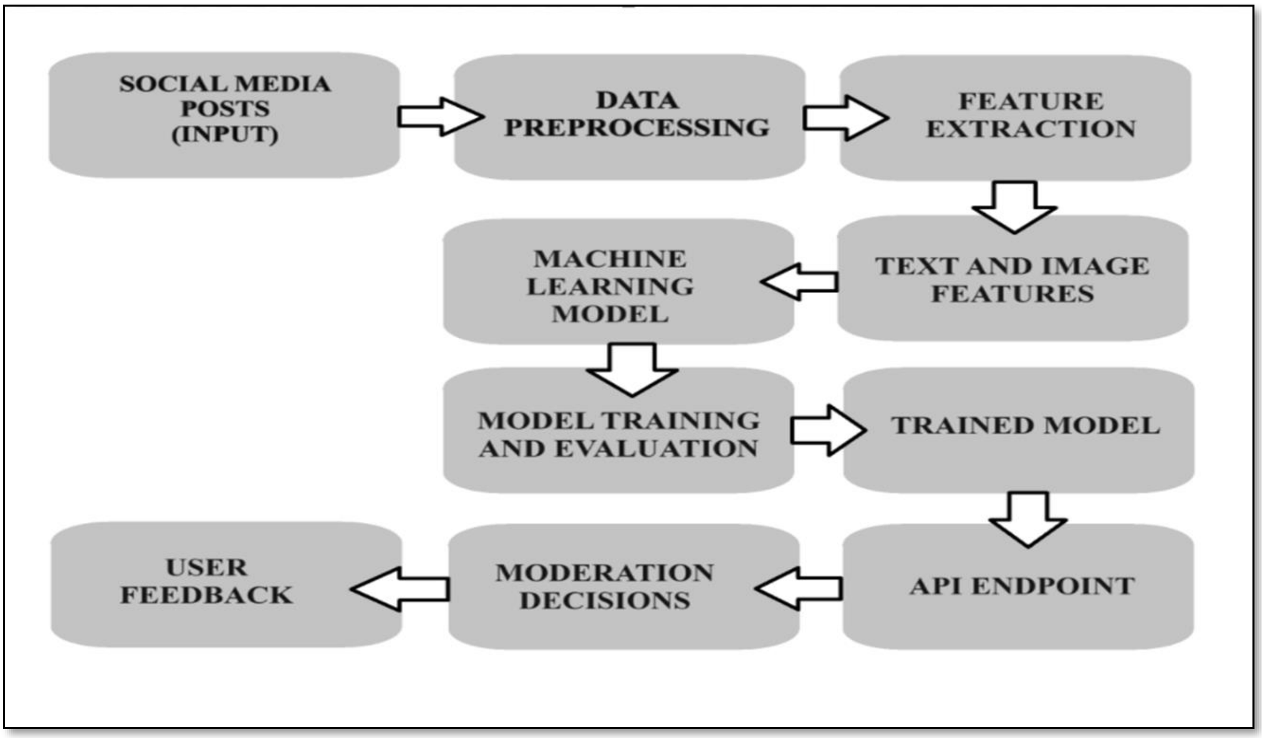
Deployment and Integration:

Once the model is trained and evaluated, the next step is deployment. This involves integrating the model into the application developed for real-time content analysis. The deployment process includes setting up the infrastructure to run the model, ensuring it can handle live data inputs, and optimizing it for performance and scalability. This step also involves implementing any necessary APIs and interfaces to allow the application to communicate with the model and utilize its predictions effectively.

Algorithm Overview:

After deployment, the final step is to observe the outcome of the project and assess the effectiveness of the model used. This involves continuous monitoring of the model's performance in a real-world setting, collecting feedback, and making improvements as needed. The effectiveness of the model is assessed based on its accuracy in detecting vulgar content, the speed of predictions, and its ability to generalize across different types of content. Regular updates and retraining may be necessary to maintain the model's performance and adapt to evolving content trends.

SYSTEM ARCHITECTURE FLOW DIAGRAM: -



CHAPTER 5

CONCLUSION & FUTURE SCOPE

5.1. CONCLUSION:

The primary aim of this project is to develop an AI tool capable of detecting offensive and vulgar posts or content posted by multiple users online. The objective is to automate the detection and filtering of such content to prevent it from remaining on the internet, thus enhancing the overall quality of online interactions and safeguarding users from exposure to harmful material. This automation saves considerable time and resources compared to manual moderation. Implementing this AI-driven solution not only ensures a more consistent and scalable approach to content moderation but also improves the user experience by maintaining a safer online environment. The integration of this AI tool into social media platforms can significantly reduce the presence of offensive content, fostering a more positive and respectful digital community.

5.2. FUTURE SCOPE:

Looking ahead, this project has the potential for several impactful enhancements. One promising direction is the extension of the AI tool's capabilities to filter out unwanted and fake user accounts. By incorporating advanced techniques such as anomaly detection and behavioral analysis, the tool can identify and remove fraudulent accounts, thereby improving the overall integrity of user interactions. Additionally, the AI tool can be further developed to prevent the upload of vulgar media content, including images and videos, by leveraging advanced computer vision techniques. This expansion would enable the tool to handle a broader spectrum of user-generated content, ensuring comprehensive moderation across various media types. Future iterations of the tool could also include multilingual support, enabling it to detect offensive content in multiple languages, thus widening its applicability and effectiveness on a global scale. Through these enhancements, the AI tool can evolve into a robust and versatile solution for maintaining a safe and respectful online environment.

REFERENCES

[1]. "Detecting Offensive Language in Social Media Using Machine Learning Techniques“, Stefanos Angelidis, Mirella Lapata, Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI-15), 2015.

[2]. "Deep Learning for Hate Speech Detection in Tweets“, Zeerak Waseem, Presented at the First Workshop on Abusive Language Online, co-located with the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017).

[3]. https://www.kaggle.com/datasets/thedevastator/hate-speech-and-offensive-language-detection

[4]. https://www.w3schools.com/ai/ai_tensorflow_intro.asp

[5]. https://www.geeksforgeeks.org/natural-language-processing-overview/