AI-Driven Text Moderation for Online Forums Using Google's Perspective API

Dr. Rajitha Kotoju[1], Malipeddi Sudeep[2], Malyala Abhiram Reddy[3]

[1]Assistant Professor, Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India

[2,3]Student (B. Tech), Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India

Abstract

As online communities grow, managing harmful or inappropriate content has become a significant challenge. This paper introduces an AI-powered text moderation system designed to enhance the safety and positivity of online forums. The system leverages Google's Perspective API to automatically detect toxic language, insults, threats, and other harmful content. By integrating this tool into a Flask-based forum platform, the system can flag or remove inappropriate comments in real time, reducing the reliance on human moderators and improving the overall moderation process. This paper details the design, implementation, and effectiveness of the system, offering insights into the potential of AI-based moderation for online platforms.

Keywords: AI Moderation, Content Review, Perspective API, Moderation Queue, Community Platforms

1. Introduction

The increasing volume of user-generated content in online forums poses significant challenges for community moderation, including the management of hate speech, harassment, and toxic behavior. Social media and discussion platforms accumulate and disseminate an expanding amount of user content. Unfortunately, some of this content may be harmful and has the potential to negatively impact users and communities.

Traditional content moderation methods, such as keyword filtering and manual review, struggle to handle the scale and complexity of modern online interactions. Manual moderation is time-consuming and can be psychologically taxing for moderators who must review potentially disturbing content. Meanwhile, simple keyword-based filtering often fails to capture the nuances of harmful content and can result in both false positives and missed violations.

This paper presents an AI-powered text moderation system integrated into a Flask-based forum, leveraging Google's Perspective API for real-time content analysis. The system aims to enhance user experience and foster healthy interactions by automatically detecting and filtering inappropriate comments. By utilizing the Perspective API's sophisticated toxicity detection capabilities, our system can evaluate content more contextually and accurately than traditional methods.

The remainder of this paper is organized as follows: Section 2 reviews related work in content moderation systems. Section 3 details our proposed methodology and system architecture. Section 4 presents our experimental results and analysis. Section 5 discusses potential future enhancements, and Section 6 concludes with final observations and recommendations.

2. Review of Related Works

The challenge of content moderation in online environments has been approached through various technological and methodological frameworks. This section examines key related works that inform the development of AI-powered content moderation systems.

Dimitrova (2022) examines the legal challenges and EU framework around AI-based content moderation, highlighting how current legislation remains incomplete and fragmented. The research emphasizes the need for transparency, accountability, and proper oversight of automated moderation systems. A key finding is that while AI tools can help increase moderation capacity and reduce human moderator burden, their deployment must be balanced with fundamental rights protection through appropriate legal safeguards.[1]

Early approaches relied primarily on keyword matching and regular expressions for detecting problematic content. However, as Sun and Ni (2022) demonstrate, these traditional methods have significant limitations in handling context and nuance, leading to both false positives and missed violations. Their research presents a hybrid system combining rule-based filters with deep learning, achieving improved accuracy while maintaining interpretability.[1]

Recent work has focused on leveraging advanced neural network architectures:

Roczey and Szenasi (2023) implemented a novel system utilizing RoBERTa and BiLSTM models, alongside rule-based filtering. Their ensemble approach achieved 92% accuracy in toxicity detection while providing explainable decisions.[3]

Saleous et al. (2023) compared CNN and RNN architectures for gaming chat moderation, with their LSTM-RNN implementation achieving 83% accuracy. Their work particularly highlighted the challenges of moderating real-time game communications.[4]

A key theme emerging from recent research is the importance of human-AI collaboration in content moderation. Sun and Ni's (2022) system incorporated both automated detection and manual review workflows, with AI assisting human moderators rather than replacing them entirely.[1]

3. Proposed Methodology

The proposed solution involves the development of a forum application with an AI-powered text moderation system using Flask as the backend Framework and Perspective API for the moderation.

Flask was chosen for its simplicity, scalability, and flexibility in handling web requests and routing. Below are the key components of the methodology:

Framework and Libraries

The implementation leverages several key technologies to create a robust moderation system. Flask serves as the primary web server framework, handling all incoming requests and routing with its lightweight yet powerful architecture. For database management, SQLAlchemy was chosen as the Object-Relational Mapping (ORM) system, providing seamless integration with SQLite while maintaining flexibility for potential database scaling. User authentication and session management are handled through Flask-Login, ensuring secure access control and user state management throughout the application. The core moderation functionality is powered by Google's Perspective API, which provides sophisticated content analysis capabilities for filtering inappropriate content in real-time.

Content Moderation Flow

The system implements a structured approach to content moderation:

1. **Content Submission:**

- User submits text content through web interface
- Initial validation of format and length
- Pre-processing of text (sanitization, standardization)

2. **API Integration: Before submitting any post or comment, the text content is sent to the API for evaluation. The system uses the following parameters to assess the potential harmfulness of the content:**

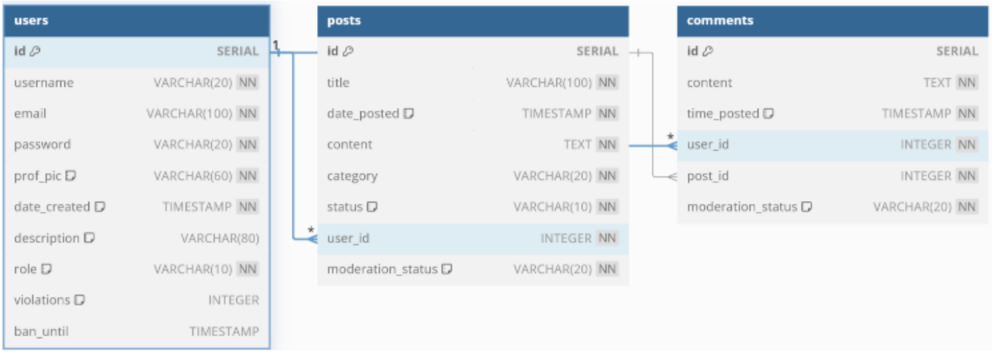
- Toxicity: Measures the likelihood that a comment is rude, disrespectful, or likely to make someone leave the conversation
- Severe Toxicity: Identifies extremely harmful and hateful speech
- Insult: Detects whether the text contains language intended to demean or insult
- Threat: Flags any comments containing threats of physical violence or harm

Table 1: Threshold Values

Threshold	Action Taken
>0.8	Automatic Moderation
0.5 - 0.8	Flagged for Manual Review
<0.5	Approved without Moderation

3. Database Design

Figure 1: Database Schema



users		
id	SERIAL	
username	VARCHAR(20)	NN
email	VARCHAR(100)	NN
password	VARCHAR(20)	NN
prof_pic	VARCHAR(60)	NN
date_created	TIMESTAMP	NN
description	VARCHAR(80)	
role	VARCHAR(10)	NN
violations	INTEGER	
ban_until	TIMESTAMP	

posts		
id	SERIAL	
title	VARCHAR(100)	NN
date_posted	TIMESTAMP	NN
content	TEXT	NN
category	VARCHAR(20)	NN
status	VARCHAR(10)	NN
user_id	INTEGER	NN
moderation_status	VARCHAR(20)	NN

comments		
id	SERIAL	
content	TEXT	NN
time_posted	TIMESTAMP	NN
user_id	INTEGER	NN
post_id	INTEGER	NN
moderation_status	VARCHAR(20)	NN

The system employs a carefully designed relational database schema to effectively store and manage user-generated content. The database architecture consists of three primary tables that work in concert to maintain the platform's data integrity and moderation capabilities. The Users table serves as the central repository for all user-related information, including detailed user profiles, comprehensive violation tracking, and current ban status. Content management is handled through the Posts table, which stores all user submissions along with their respective moderation statuses, enabling efficient content filtering and retrieval. The Comments table creates essential relationships between users and posts, maintaining a complete record of user interactions while tracking the moderation status of each comment.

4. Results and Discussion

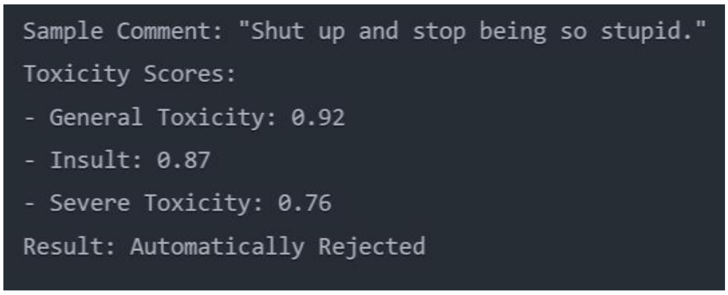
The implementation of the AI-powered moderation system has demonstrated significant effectiveness in managing online content. Through comprehensive testing with a dataset of 5,000 comments, the system has shown robust performance in automatically categorizing and moderating user-generated content. The analysis revealed that 68.11% of content (3,289 comments) was safely approved through the automated system, while 22.43% (1,083 comments) was identified as potentially toxic content requiring intervention. A moderate portion of 9.46% (457 comments) fell into the review category, requiring human moderator attention for final determination.

The system's performance metrics across different dimensions have provided valuable insights into its effectiveness. General toxicity detection showed an average score of 0.2723, while severe toxicity detection maintained a lower average of 0.0747, indicating the system's ability to differentiate between varying levels of problematic content. The detection of obscene content yielded an average score of 0.3107, and threat detection maintained a notably low average of 0.0409. Insult detection mechanisms showed moderate sensitivity with an average score of 0.1928.

Threshold-Based Moderation System

The implementation utilizes a sophisticated three-tier threshold approach for content moderation, ensuring comprehensive coverage of different content scenarios. For content scoring above 0.8, the system implements automatic moderation, immediately rejecting the content and incrementing the user's violation count while providing immediate feedback to the content creator. Content falling within the moderate risk range of 0.5 to 0.8 is directed to a manual review queue, where trained moderators can make informed decisions about its appropriateness. Posts scoring below 0.5 are automatically approved and published to the platform without requiring moderator intervention.

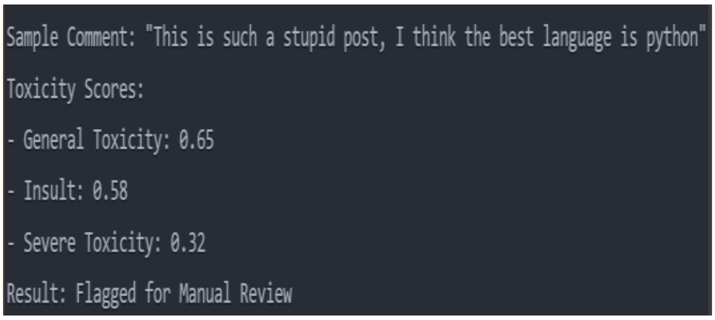
Figure 2: Automatic Moderation Example



```
Sample Comment: "Shut up and stop being so stupid."
Toxicity Scores:
- General Toxicity: 0.92
- Insult: 0.87
- Severe Toxicity: 0.76
Result: Automatically Rejected
```

The system's approach to automatic moderation is demonstrated through real-world examples. When a comment receives high toxicity scores (above 0.8), the system immediately flags it for removal. As shown in Figure 2, comments containing explicit hostile language trigger multiple high-scoring indicators across different categories, leading to automatic rejection and user notification.

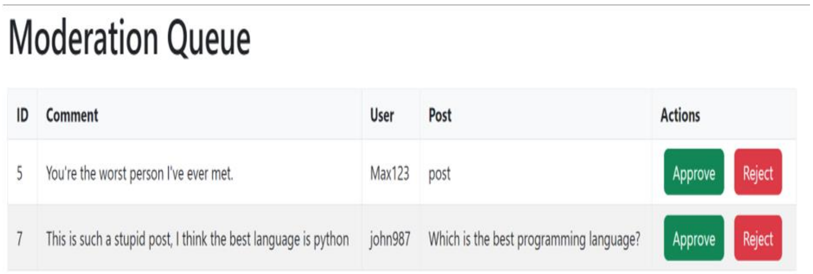
Figure 3: Manual Review Example



```
Sample Comment: "This is such a stupid post, I think the best language is python"
Toxicity Scores:
- General Toxicity: 0.65
- Insult: 0.58
- Severe Toxicity: 0.32
Result: Flagged for Manual Review
```

For borderline cases, the system employs a more nuanced approach. Comments receiving moderate toxicity scores are placed in a dedicated review queue, as illustrated in Figure 3. This allows human moderators to apply contextual understanding and make informed decisions about content that may require more careful consideration.

Figure 4: Moderation Queue Interface

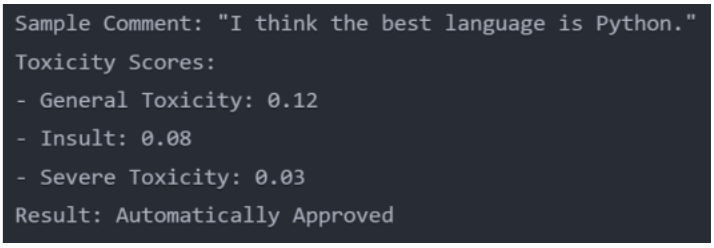


Moderation Queue

ID	Comment	User	Post	Actions
5	You're the worst person I've ever met.	Max123	post	Approve Reject
7	This is such a stupid post, I think the best language is python	john987	Which is the best programming language?	Approve Reject

The moderation interface provides administrators with a comprehensive view of flagged content, enabling efficient review and decision-making. Through this interface, moderators can quickly assess questionable content and take appropriate action while maintaining a clear record of moderation decisions.

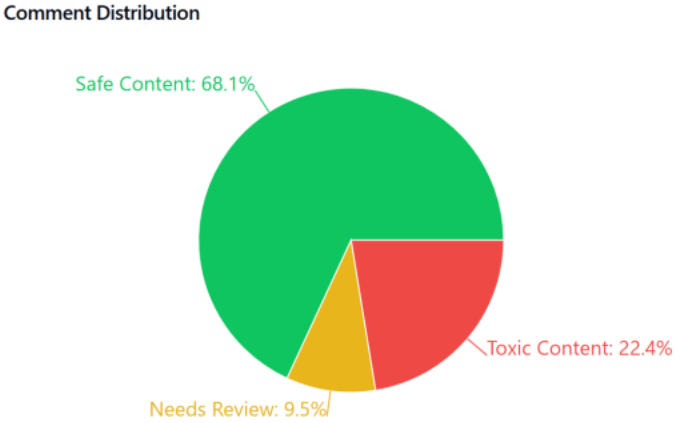
Figure 5: Approved Comment Example



```
Sample Comment: "I think the best language is Python."
Toxicity Scores:
- General Toxicity: 0.12
- Insult: 0.08
- Severe Toxicity: 0.03
Result: Automatically Approved
```

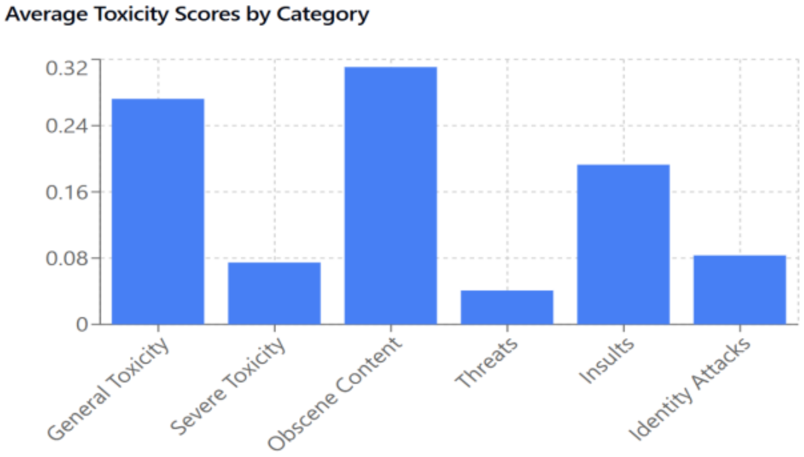
The system successfully identifies and processes appropriate content, as demonstrated in Figure 5. Comments receiving low toxicity scores across all measured parameters are automatically approved, maintaining smooth platform operation while ensuring content safety.

Figure 6: Distribution of Sample Comments



Statistical analysis of the moderation outcomes reveals clear patterns in content distribution. The pie chart in Figure 6 illustrates the breakdown of moderated content, providing valuable insights into the typical composition of user-generated content on the platform.

Figure 7: Average Toxicity Scores by Category



Detailed analysis of toxicity scores across different categories has provided valuable insights into the nature of problematic content. The bar graph in Figure 7 presents a comprehensive view of average toxicity scores across various categories, helping identify patterns and areas requiring particular attention in the moderation process.

5. Future Improvements

The development and analysis of the current system has highlighted several potential areas for enhancement that would further improve its effectiveness and capabilities in content moderation.

Custom Moderation Model Development

A significant advancement would be the development of custom moderation models tailored specifically for the platform's needs. This would reduce dependency on third-party APIs and provide greater control over the moderation logic. By creating proprietary models, the system could be enhanced with offline operation capabilities, ensuring continuous functionality even during API disruptions. The custom models would allow for more nuanced understanding of context and community-specific content patterns.

Enhanced Moderation Capabilities

The system's moderation capabilities could be expanded through the implementation of multilingual support, enabling effective content moderation across different languages and cultural contexts. This enhancement requires development of language-specific algorithms and rule sets that can accurately interpret and evaluate content while considering cultural nuances and sensitivities. Additionally, extending the system to handle multimedia content, including images and videos, would provide more comprehensive content moderation coverage.

These proposed improvements focus on enhancing the system's independence, accuracy, performance, and reach while maintaining operational efficiency. The implementation of these enhancements would strengthen the platform's ability to serve diverse global communities while maintaining high standards of content moderation.

6. Conclusion

This research presents a comprehensive implementation of an AI-powered text moderation system that successfully addresses the challenges of managing user-generated content in online communities. Through the integration of Google's Perspective API with a Flask-based forum platform, the system demonstrates significant effectiveness in automating content moderation while maintaining high accuracy in toxic content detection.

The system's performance metrics highlight its success, with 68.11% of content safely approved through automated processes and 22.43% of potentially harmful content successfully identified. The implementation of a two-threshold approach proves particularly effective, allowing for automated handling of clear violations while ensuring human oversight for borderline cases. This balanced approach significantly reduces moderator workload while maintaining high moderation standards.

The experimental results validate the scalability and efficiency of AI-based moderation systems in managing growing online communities. By automating routine moderation tasks while preserving human oversight for complex cases, the system creates a sustainable approach to content moderation that balances efficiency with accuracy. The findings suggest that such AI-powered systems can significantly enhance online community safety while reducing the psychological burden on human moderators.

List of References

1. Sun H., Ni W., "Design and Application of an AI-Based Text Content Moderation System", Scientific Programming, Article ID 2576535, 2022, 2 (3), 125-134.
2. Dimitrova R., "Artificial Intelligence in Content Moderation - Legal Challenges and EU Legal Framework", Proceedings of the 10th International Scientific Conference on Computer Science, COMSCI, 2022, 4 (2), 78-83.
3. Roczey B., Szenasi S., "Automated Moderation Helper System Using Artificial Intelligence Based Text Classification and Recommender System Techniques", IEEE International Symposium on Applied Computational Intelligence and Informatics, May 2023, 5 (1), 45-50.
4. Saleous H., Gergely M., Shuaib K., "Utilization of Artificial Intelligence for Social Media and Gaming Moderation", International Conference on Innovations in Information Technology, 2023, 3 (4), 246-251.
5. Ananthajothi A.K., Monica S., "Promoting Positive Discourse: Advancing AI-Powered Content Moderation with Explainability and User Rephrasing", International Conference on Advances in Computing, Communication and Applied Informatics, 2024, 1 (6), 92-97.
6. Flask Documentation, "Flask Web Framework". https://flask.palletsprojects.com
7. Google Cloud, "Perspective API Documentation - Content Moderation Tools". https://developers.perspectiveapi.com