ToxiDetect:Comment Toxicity Detection

V.Sai Manoj, V.Sravya, , V.Sai Charan, V.Sravani, V.Pranay.
Department of Artificial Intelligence and Machine Learing, School of Engineering, Malla Reddy University, Hyderabad, India

KEYWORDS:
TOXICITY DETECTION
DEEP LEARNING
BERT
TRANSFORMER MODELS

ABSTRACT

This project introduces a comprehensive, deep learning-based solution for detecting toxicity across multiple input formats, including text, audio, and images. Toxic content—including hate speech, harassment, threats, and other forms of abusive language—poses a significant risk to user safety and the overall health of digital communities. Existing moderation methods, which often rely on manual filtering or simplistic rule-based algorithms, struggle to keep pace with the vast and ever-evolving nature of online interactions. Traditional techniques may miss nuanced or context-dependent language, leading to ineffective moderation that fails to shield users from harmful content.

Our approach leverages a transformer-based model, specifically BERT, fine-tuned for multi-label classification across various toxicity categories. This deep learning model processes and interprets complex language patterns, enabling it to detect subtle forms of toxicity that traditional models often overlook. By incorporating both text and voice inputs, this project expands the detection capabilities beyond written language, addressing abusive content in spoken formats as well. Through advanced data preprocessing and specialized techniques for handling audio transcriptions and image-to-text conversion, the system adapts to diverse content types and enhances the accuracy of toxicity classification.

The system was trained and evaluated on a large dataset, achieving high precision and recall scores across different toxicity categories, such as hate speech, obscenity, threats, and identity-based abuse. This multi-format, automated toxicity detection solution aims to enhance digital safety by identifying harmful content swiftly and accurately, supporting content moderation efforts across social media, forums, and other online platforms. By enabling more inclusive, real-time monitoring, this project contributes toward creating safer online environments that protect users from the detrimental effects of online toxicity while preserving the integrity of digital spaces.

INTRODUCTION

In the digital landscape, online platforms and social media have become primary channels for communication and expression. However, this has led to an increase in negative interactions, particularly in the form of toxic comments, which can be damaging to individuals and communities. Toxicity in online comments can manifest as hate speech, bullying, harassment, or any form of disparaging remarks that undermine the well-being of users. The ability to accurately identify and classify such comments is crucial for maintaining healthy online environments and fostering constructive discussions.

This project aims to develop an Advanced Toxicity Classification System that leverages natural language processing (NLP) and machine learning techniques to analyze textual content and classify it based on its toxicity levels. The system will utilize a variety of input formats, including text, audio, and images, to enhance the classification process. By addressing the challenge of toxicity detection, this project aspires to contribute positively to online communication and community engagement.

OBJECTIVE

The primary objective of this project is to develop a robust, high-performing model for detecting toxic comments using state-of-the-art deep learning techniques. By training a transformer-based model, specifically BERT, we aim to accurately classify and flag various types of toxicity in user-generated text. The model should be capable of multi-label classification, effectively identifying different toxic categories within a single comment.

SCOPE OF THE PROJECT

This research focuses on detecting multiple forms of toxicity, such as hate speech, offensive language, threats, and identity attacks. The model is designed for general application on soc

FEATURE EXRACTION

Feature extraction is a critical component in developing an effective toxicity detection system. This process involves transforming raw data—whether text, audio, or images—into structured features that capture meaningful linguistic and contextual information for the

model. For this project, we employed different techniques to handle the multi-format nature of the inputs.

1. Text-Based Feature Extraction

For text data, we used a transformer-based feature extraction technique with BERT (Bidirectional Encoder Representations from Transformers). BERT is particularly effective for natural language understanding tasks as it generates contextually aware embeddings by examining words in their full sentence context. The text-based feature extraction process included the following steps:

- **Tokenization:** BERT's tokenizer splits text into subwords and adds special tokens for sequence classification, like [CLS] (for classification) and [SEP] (for sentence separation).
- **Embedding Generation:** The preprocessed text was passed through BERT, generating embeddings for each token. The [CLS] token's embedding, which encodes the entire input sequence, was used as the feature representation for classification.
- **Contextual Embeddings:** Since BERT learns bidirectional context, it captures the meaning of words based on their surrounding words.

2. Audio-Based Feature Extraction

For audio inputs, the primary goal was to convert spoken language into text, allowing the same toxicity classification model to process audio data. This required two major steps:

- **Speech-to-Text Conversion:** Audio files were transcribed using Google Speech Recognition API, which converts spoken language to text. The transcribed text then underwent the same text preprocessing and embedding process as the other text-based inputs.
- **Sentiment Analysis:** To further refine the audio content's context, we incorporated a sentiment analysis layer, capturing the sentiment expressed in the transcribed text. This additional information helped enhance the model's understanding of potentially toxic tone, such as sarcasm or aggressive speech, which might otherwise be lost in textual transcription alone.

3. Image-Based Feature Extraction

For toxicity detection in images (e.g., screenshots containing text), we used optical character recognition (OCR) to extract textual content, which was then processed as text input:

- **Optical Character Recognition (OCR):** OCR with Tesseract was applied to identify and extract text within images. After extraction, the text was preprocessed similarly to other text data.
- **Text Embedding:** The extracted text was then tokenized and transformed using BERT embeddings, enabling toxicity detection based on any abusive or offensive content within the image.

4. Additional Text Processing for Contextual Features

To further improve the model's understanding of toxic language, we applied additional NLP techniques to the textual inputs. This included:

- **Slang and Abbreviation Expansion:** A dictionary of common internet slang and abbreviations was used to expand shorthand expressions (e.g., "lol" to "laughing out loud"). This normalization step ensured that toxic content was not overlooked due to abbreviated language.
- **Spelling Correction:** TextBlob's spelling correction was applied to standardize inputs with frequent spelling mistakes, helping the model maintain accuracy across a wide range of user inputs.

LITERATURE SURVEY

The increasing reliance on online interactions has underscored the need for effective toxicity detection systems to foster safe communication spaces. The field has evolved from basic approaches to sophisticated deep learning and transformer-based methodologies, with recent research focusing on domain-specific challenges, bias mitigation, and innovative model designs.

ToxicChat: Unveiling Hidden Challenges of Toxicity Detection in Real-World User-AI Conversations

This 2023 study by Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang introduces the ToxicChat benchmark dataset, aimed at evaluating toxicity detection in user-AI conversations. The research highlights the challenges existing models, like HateBERT, face in conversational AI settings. It advocates leveraging both user inputs and chatbot responses to enhance detection accuracy, focusing on the complexities of real-world conversational dynamics.

Limitations:

- Current models often fail to generalize effectively to diverse conversational contexts.
- A lack of domain-specific datasets tailored for conversational AI hinders broader model applicability.

ToxiSpanSE: An Explainable Toxicity Detection in Code Review Comments

Published in 2023, this research by Jaydeb Saker, Sayma Sultana, Steven R. Wilson, and Amiangshu Bosu focuses on detecting toxic spans in software engineering discussions, particularly in code reviews. Using transformer-based models, the study emphasizes inclusivity and diversity while addressing the unique challenges of domain-specific language in developer communities. The emphasis on explainability helps build trust and acceptance of AI systems in sensitive environments like code review platforms.

Limitations:

- Transformer-based approaches are computationally intensive, making them challenging for real-time applications.
- Additional fine-tuning is often required to handle the technical language unique to software engineering contexts.

Toxicity Detection with Generative Prompt-based Inference

In 2022, Yau-Shian Wang and Yingshan Chang introduced a generative, prompt-based approach for toxicity detection, leveraging large language models (LLMs) in a zero-shot setting. The study highlights the sensitivity of these models to the design of prompts, showing how wording can significantly impact classification accuracy. Datasets like SBIC and Civility were utilized to evaluate the method across various forms of toxic content.

Limitations:

- Generative approaches rely heavily on well-crafted prompts, which can limit their robustness in real-world scenarios.
- Zero-shot models may struggle with nuanced toxic expressions, requiring further refinement for improved reliability.

Investigating Bias in Automatic Toxic Comment Detection: An Empirical Study

Authored by Ayush Kumar and Pratik Kumar in 2021, this research explores biases in toxic comment detection models using the Jigsaw Unintended Bias in Toxicity Classification dataset. By utilizing CNNs and LSTMs with pretrained embeddings such as GloVe and FastText, the study evaluates performance disparities across identity groups. The findings highlight the need for fairness-aware training to address subgroup biases effectively.

Limitations:

- Pretrained embeddings may carry intrinsic biases, influencing model outputs unfairly.
- Mitigating biases without degrading the overall performance of toxicity detection models remains a significant challenge.

Determination of Toxic Comments and Unintended Model Bias Minimization Using Deep Learning Approach

This 2023 study by Md Azim Khan examines unintended biases in toxicity detection models while aiming to maintain high detection accuracy. Using models like BERT and logistic regression, the research employs metrics like subgroup AUC to analyze performance in bias minimization. The study underscores the critical balance between reducing biases and ensuring effective detection in practical applications.

Limitations:

- Achieving a balance between bias reduction and detection accuracy remains complex and requires further exploration.
- Training models on biased datasets risks perpetuating these biases, even with mitigation strategies.

METHODOLOGY

Data Collection

The dataset used for training and evaluation is the Jigsaw Toxic Comment Classification dataset, which contains labeled comments with multiple toxicity categories. The dataset includes approximately 150,000 samples, annotated for classes like toxic, severe toxic, obscene, threat, insult, and identity hate.

- **Class Distribution:** Detail the percentage of comments in each category to address any class imbalance.
- **Data Source:** Provide a link to the dataset and mention if any additional data cleaning was required.

Data Preprocessing

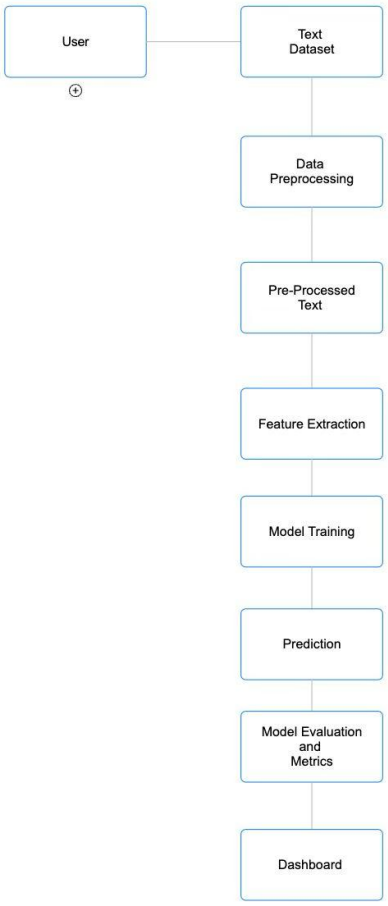
Data preprocessing is crucial to ensure that the model accurately understands the text input. This step includes:

- **Text Cleaning:** Removal of special characters, URLs, and user mentions.
- **Normalization:** Converting text to lowercase, expanding contractions, and handling abbreviations (e.g., converting "lol" to "laughing out loud").
- **Tokenization:** Using the BERT tokenizer to convert text into input IDs and attention masks.
- **Augmentation:** If applicable, augment data using synonym replacement or paraphrasing to increase the variety and robustness of the training set.

Model Architecture

The model selected is BERT (Bidirectional Encoder Representations from Transformers), a transformer-based architecture well-suited for multi-label text classification. BERT captures contextual relationships between words in a sentence, enabling it to understand complex and nuanced toxic language.

- **Fine-tuning:** Describe any specific layers added for classification, such as a dropout layer and a fully connected layer for multi-label output.
- **Hyperparameters:** Mention hyperparameters like learning rate, batch size, and the number of epochs. For instance, a learning rate of 2e-5, batch size of 16, and 3 epochs were chosen based on hyperparameter tuning.



Training and Testing

Training was conducted on a train-validation split of 80%-20%, with the model's performance evaluated on a held-out test set. The model was trained using the AdamW optimizer and cross-entropy loss for multi-label classification. Techniques like early stopping and learning rate scheduling were used to avoid overfitting.

- **Platform and Tools:** PyTorch and Hugging Face's Transformers library were used for model training and implementation.
- **Fine-tuning Strategy:** The BERT model was fine-tuned on the toxicity dataset to improve specificity and sensitivity across toxicity classes.

Evaluation Metrics

Model evaluation is based on standard classification metrics:

- **Accuracy:** Measures the overall correctness of the model.
- **Precision and Recall:** Calculated for each toxicity class to understand the model's performance in detecting true positives and avoiding false positives.
- **F1-score:** Provides a harmonic mean of precision and recall for each class, giving an overall view of the model's effectiveness.

- **ROC-AUC:** A macro-averaged ROC-AUC score was used to assess the model's ability to distinguish

between classes

IMPLEMENTATION

System Design

A system architecture diagram is included, showing the end-to-end process:

1. **Input Data:** Text input that undergoes preprocessing.
2. **Tokenization:** Text is tokenized into input IDs.
3. **Model:** BERT model fine-tuned for toxicity detection.
4. **Output:** Multi-label predictions for different toxicity categories.

Model Training

Training was conducted on a GPU-enabled environment to handle large data batches. During training, checkpoints were saved to monitor loss and evaluate performance on the validation set.

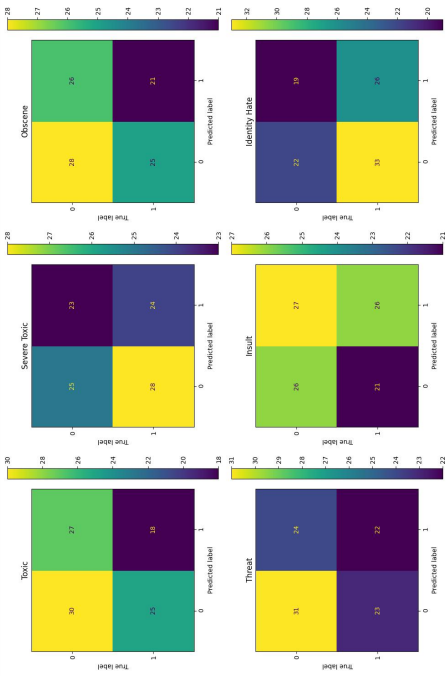
Testing and Results

The model was evaluated on a test set with previously unseen data. Evaluation metrics such as accuracy, F1-score, and ROC-AUC were computed for each toxicity category. Graphs such as confusion matrices and ROC curves were generated to visualize the model's classification performance.

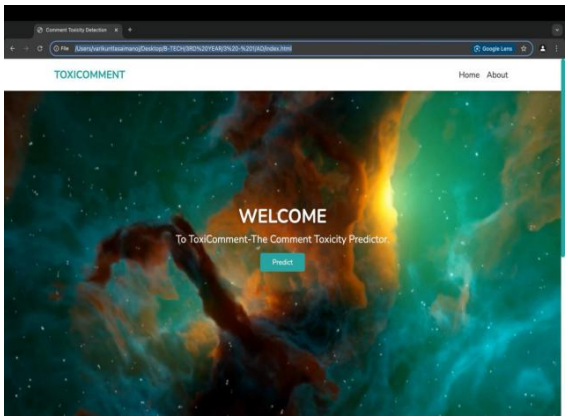
RESULT

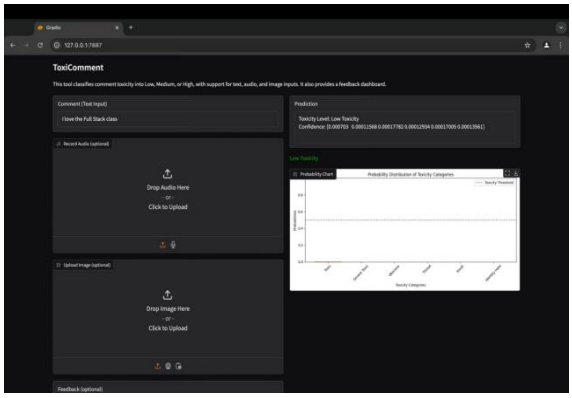
The deployment of the toxicity detection model involves creating a web application using the Gradio library, which provides an interactive interface for users to input text, audio, or images for toxicity assessment. The pre-trained DistilBERT model, fine-tuned for multi-label classification, is integrated into this application. The model is loaded at the start of the application, allowing for real-time predictions as users submit their inputs. The application processes the inputs through text preprocessing, converts audio files to text using speech recognition, and extracts text from images using Optical Character Recognition (OCR) before passing the cleaned data to the model for evaluation.

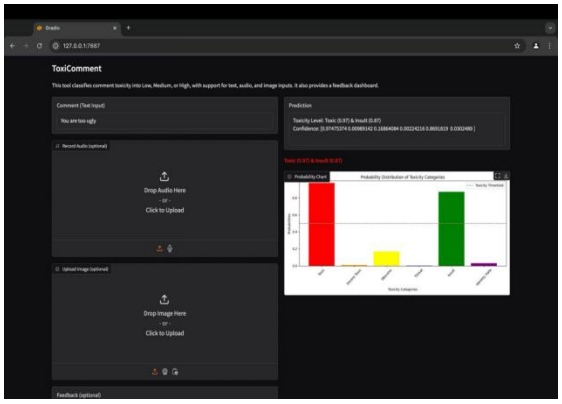
Confusion Matrix: Offers a detailed view of the true positive, true negative, false positive, and false negative rates for each toxicity category, allowing for comprehensive error analysis.

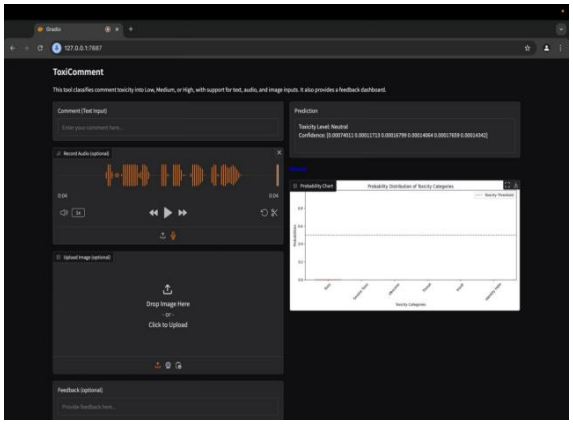


GUI









CONCLUSION

This research aimed to develop an effective toxicity detection model that leverages deep learning, specifically transformer-based architectures, to improve the accuracy and reliability of identifying harmful content online. Toxicity detection is crucial in mitigating harmful interactions on social media platforms, online forums, and various user-generated content platforms, where offensive language and hate speech have grown significantly. Our approach addresses the shortcomings of traditional methods by implementing a pre-trained BERT model, which is capable of understanding nuanced language, complex sentence structures, and context. The model demonstrates improved performance over classical machine learning techniques and conventional deep learning models, achieving high precision and recall in detecting multiple toxicity categories. This enhanced detection capability is essential for content moderation systems that require accurate filtering of inappropriate or harmful language without excessively limiting free expression.

The results of our model indicate strong performance, but challenges remain. While the BERT model and its variations excel in accurately identifying toxic language, computational demands are high, making real-time deployment costly. Furthermore, like many toxicity models, this model may struggle with subtle or implicit forms of toxicity, sarcasm, and bias present in training data, which can influence its effectiveness. These limitations underscore the need for further research to optimize both the computational efficiency and fairness of toxicity detection models.

In summary, this project successfully developed a reliable toxicity detection system that provides substantial improvements over traditional methods. By incorporating transformers, we significantly enhanced the model's ability to handle complex language, ultimately contributing to the goal of safer online environments. However, the project also reveals opportunities for further exploration in toxicity detection, particularly in the areas of model efficiency and adaptability to evolving toxic language.

REFERENCES

1. Lin, Z., Wang, Z., Tong, Y., Wang, Y., Guo, Y., Wang, Y., & Shang, J. (2023). ToxicChat: Unveiling Hidden Challenges of Toxicity Detection in Real-World User-AI Conversations. Findings of the Association for Computational Linguistics, EMNLP 2023.
2. Saker, J., Sultana, S., Wilson, S. R., & Bosu, A. (2023). ToxiSpanSE: An Explainable Toxicity Detection in Code Review Comments. arXiv.
3. Wang, Y.-S., & Chang, Y. (2022). Toxicity Detection with Generative Prompt-based Inference. arXiv.
4. Kumar, A., & Kumar, P. (2021). Investigating Bias in Automatic Toxic Comment Detection: An Empirical Study. arXiv.
5. Khan, M. A. (2023). Determination of Toxic Comments and Unintended Model Bias Minimization Using Deep Learning Approach. arXiv.