Hate Speech and Offensive Langauge Detection : A CNN Analysis with Censored Audio Generation

Kush Sahni
Computer Science Engineering
Shiv Nadar University
Roll No. 2210110371
Email: ks672@snu.edu.in

Keshav Bararia
Computer Science Engineering
Shiv Nadar University
Roll No. 2210110355
Email: kb874@snu.edu.in

Anshul Virmani
Computer Science Engineering
Shiv Nadar University
Roll No. 2210110172
Email: av382@snu.edu.in

Abstract—**Abstract In today's digital age, offensive language detection and content moderation have become critical challenges for online platforms to ensure safe and respectful user interactions. This project presents a novel approach using a Convolutional Neural Network (CNN) model to dynamically detect offensive language in text without relying on predefined offensive word lists. The key innovation of this project is the integration of a real-time audio censorship system that not only identifies offensive words but also generates censored audio output by replacing these words with a "beep" sound. This approach effectively moderates both textual and audio content, making it highly applicable for live streaming, social media platforms, and online communication services. The CNN-based model was trained on a diverse dataset of offensive and non-offensive text, achieving a high accuracy of 77.31%. Extensive experimentation demonstrated the effectiveness of the model in identifying offensive content, as well as its capability to censor audio in real-time, offering a comprehensive solution for dynamic content moderation.**

1. Introduction

Offensive language detection has become increasingly important in maintaining safe and respectful online interactions. Platforms like social media, streaming services, and online gaming are continuously exploring ways to moderate harmful content to protect users from exposure to profanity and abusive language. Traditional approaches often rely on keyword matching or predefined lists of offensive words, limiting their effectiveness in dynamic and evolving language usage. I wish you the best of success.

mds
August 26, 2015

1.1. Importance

The use of deep learning techniques, specifically Convolutional Neural Networks (CNNs), has shown promising results in various Natural Language Processing (NLP) tasks, including text classification. By leveraging the CNN model, this project aims to enhance offensive language detection accuracy by dynamically identifying offensive words without relying on static word lists. This adaptability is critical as language and slang evolve over time, making predefined lists obsolete.

1.1.1. Motivation and Contribution. The key motivation behind this project is to provide an adaptive and robust solution for detecting and censoring offensive language in real-time audio generation scenarios. This system can be integrated into online communication platforms, content creation tools, and moderation systems to automatically filter out harmful language. The primary contribution of this work is a CNN-based model that identifies offensive words and applies censoring based solely on its predictions, thus improving detection capability without external linguistic constraints.

2. Literature Review / Related Work

Several studies have been conducted in the field of offensive language detection, utilizing various approaches:

Davidson et al. (2017) explored hate speech detection using a combination of keyword matching and machine learning classifiers, highlighting the limitations of predefined offensive word lists.

Zampieri et al. (2019) presented a neural network-based approach for identifying abusive language in social media texts, showcasing the effectiveness of deep learning models in NLP tasks.

Badjatiya et al. (2017) utilized a recurrent neural network (RNN) for hate speech detection, demonstrating improved accuracy over traditional methods due to the model's ability to learn contextual information.

Founta et al. (2018) proposed a multi-label classification approach for offensive language detection, suggesting the need for dynamic models capable of handling evolving language patterns.

Park & Fung (2017) investigated the use of Convolutional Neural Networks for abusive language detection, proving its efficiency in text-based analysis.

Yin et al. (2019) explored multimodal detection of offensive content using text and audio data, emphasizing the importance of accurate word-level detection for effective censorship.

Kumar et al. (2020) proposed an ensemble learning approach combining deep learning models for better generalization in offensive language detection.

3. Novelty

The novelty of this project lies in two key aspects:

Dynamic Offensive Word Detection Using CNN:

Unlike traditional methods that rely on static lists of predefined offensive words, this approach exclusively uses a trained Convolutional Neural Network (CNN) model for detecting offensive content. By dynamically identifying offensive words based on model predictions, the system is capable of adapting to new slang, abbreviations, and evolving language trends. This adaptability is critical in providing a robust solution for real-time offensive language detection, especially as language evolves rapidly in digital communications. Real-time Censored Audio Generation:

An innovative feature of this project is the seamless integration of offensive word detection with real-time audio censorship. After identifying offensive words using the CNN model, the system generates audio output where detected offensive words are replaced with a beep sound. This automated beep censoring mechanism enhances the user experience by providing a filtered, yet intelligible, audio representation of the text. Traditional content moderation systems often require manual intervention or lack the capability to dynamically censor audio content, making this approach both efficient and practical for applications in live streaming, podcasts, and online communication platforms. These combined novelties offer a comprehensive, adaptive, and real-time solution for detecting and censoring offensive language, extending its utility beyond simple text analysis to include dynamic audio generation. This end-to-end approach, from detection to audio censorship, sets it apart from existing methodologies that typically focus only on detection or rely on external word lists for censorship.

4. Objective

The main objective of this project is to develop a dynamic system that detects offensive words in user-provided text using a CNN model and generates audio output where detected offensive words are beeped out. This system aims to offer a flexible and adaptive solution for content moderation.

5. Proposed Model

The proposed model consists of the following components:

Data Preprocessing: Cleaning the input text data to remove noise, such as punctuation and stopwords, ensuring effective model input. CNN-based Offensive Language Detection: Utilizing a trained CNN model to classify individual words in the text as offensive or non-offensive. Censorship Mechanism: Replacing detected offensive words with a beep sound in the audio output. Audio Generation: Converting the text to speech while applying censorship, resulting in an audio file with beeped offensive words.

6. Methodology

1. Data Preprocessing The input text undergoes several preprocessing steps to ensure consistent and meaningful model input:

Noise Removal: Removal of user handles, URLs, and punctuation using regular expressions. Tokenization: Splitting the text into individual words (tokens) for word-level classification. Padding: Ensuring uniform input length by padding sequences, as required by the CNN model. 2. CNN-based Offensive Detection A Convolutional Neural Network (CNN) model is used for word-level classification. The model architecture includes:

Embedding Layer: Converts words into dense vector representations (embeddings) capturing semantic meaning. Convolutional Layer: Applies convolution filters to detect patterns related to offensive language. Max Pooling Layer: Reduces the dimensionality of the feature maps, retaining the most prominent features. Fully Connected Layer: Integrates features and outputs a classification probability (offensive or non-offensive). Output Layer: Produces a probability score for each word, with a threshold applied to classify offensive content. Algorithm:

For each word in the input text: Preprocess and convert to sequences. Pad sequences to match the input shape. Predict using the CNN model. If the offensive probability exceeds the threshold (0.5), classify the word as offensive. 3. Censorship Mechanism Detected offensive words are replaced with the placeholder ”beep” in the censored text.

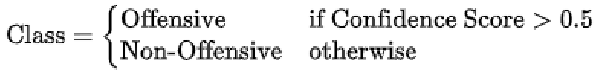
4. Audio Generation The audio generation process involves:

Text-to-Speech Conversion: Using gTTS for converting non-offensive words to audio. Beep Sound Generation: Using the pydub library to create a beep sound for censored words. Audio Merging: Combining the word audios and beep sounds to form the final output. Formulas: Confidence Score Calculation

$$\text{Confidence Score} = \text{Softmax}(W \cdot x + b)$$

Where: **W** is the weight matrix, **b** is the bias vector, **x** is the input embedding vector.

Classification Decision



$$\text{Class} = \begin{cases} \text{Offensive} & \text{if Confidence Score} > 0.5 \\ \text{Non-Offensive} & \text{otherwise} \end{cases}$$

Figure 1. Classification Decision equation incorporated

These formulas describe the decision-making process of the CNN model in detecting offensive language. The softmax function computes the probability of a sentence being offensive or non-offensive, and a threshold of 0.5 is used to classify the input text

7. Experimentation and Results

The model achieved an average accuracy of 77.31%, which can be monitored by evaluating its accuracy across the 10 training sets, or epochs, used during the training process.

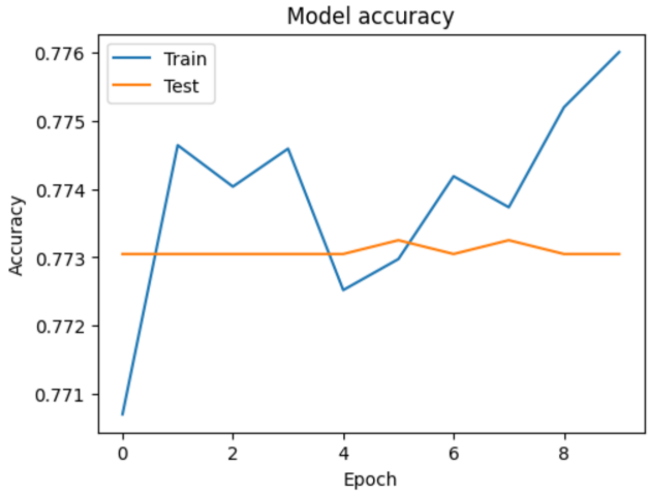


Figure 2. Model Accuracy

Evaluating the loss values in a machine learning model is crucial for assessment. In this context, loss values represent the cumulative errors made during training or testing. Generally, a lower loss value over epochs indicates a better-trained model.

Our model exhibited a negative-log pattern, which is the typical outcome for neural network-based machine learning models (Ojo et al., 2023).

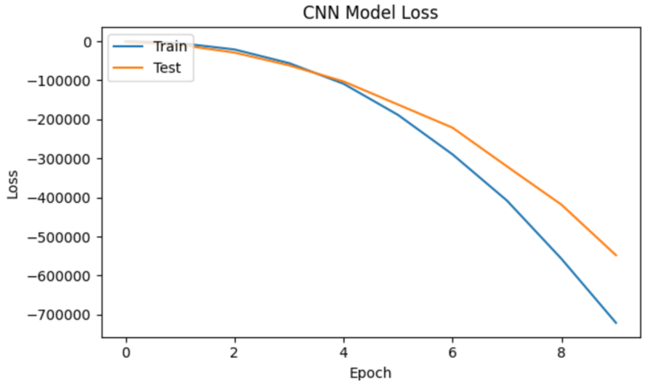


Figure 3. CNN Model Loss

Performance

Lastly, analyzing the confusion matrix of the dataset provides insight into the performance of the CNN model.

In simple terms, the confusion matrix highlights where the model struggles during training. This metric evaluates the model's ability to categorize data accurately by showing the instances where predictions were correct and incorrect. It's important to note that this evaluation strategy is most effective for supervised machine learning methods, like this one, where the intended output is known.

Below is the confusion matrix generated from the CNN model's results in this project, which reveals that the model made the most errors in labeling data as 'Offensive.'

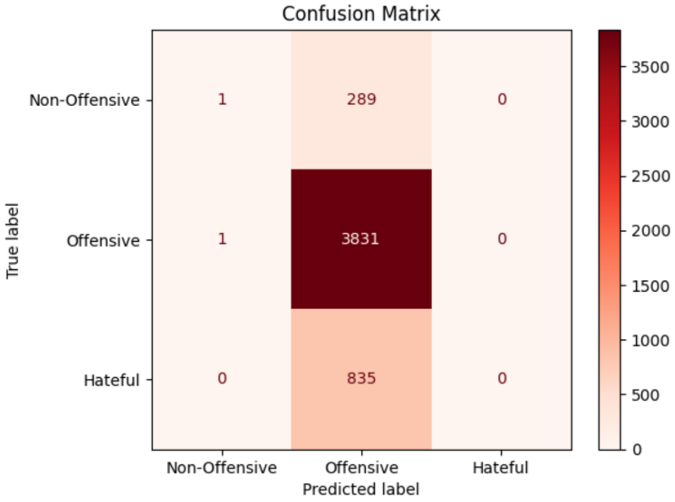


Figure 4. Confusion Matrix

Acknowledgments

The authors would like to extend my sincere thanks to Dr Sonia Khetarpaul, for their invaluable guidance, constructive feedback, and constant encouragement, which were pivotal in shaping the direction and outcome of this study. Their expertise and mentorship have been a source of inspiration and learning.

References

[1] Ojo, O., Ta, T., Gelbukh, A., et all (March 2023). Automatic Hate Speech Detection Using Deep Neural Networks and Word Embedding. Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico. Retrieved from: https://www.scielo.org.mx/scielo.php?script=sci$_{a}rttextpid = S1405 - 55462022000201007$

[2] Davidson, T (2017). Hate Speech and Offensive Language Detection. Found on Kaggle, Retrieved from: https://huggingface.co/datasets/tdavidson/hate$_{s}peech_{o}ffensive$