Homework 2 Write-up

Haoshen Wu

Abstract

This report explores methods for detecting offensive language in tweets, starting with the Perspective API's perspective score as a baseline. A custom logistic regression classifier was then developed using TF-IDF features, and advanced models were tested by incorporating additional features and alternative algorithms. Results indicate that while simpler models perform well for this task, additional features and complexity do not always improve accuracy and can introduce challenges related to bias and ethical concerns.

1 Introduction

Detecting offensive language on online platforms has become increasingly important as more interactions occur in digital spaces. Harmful content can undermine the safety and inclusivity of these spaces, making automated detection tools critical for content moderation. However, building effective and unbiased classifiers for abusive language remains a challenging task due to the nuanced and context-dependent nature of offensive language, as well as the potential for unintended bias.

This homework assignment investigates different approaches for classifying offensive language in provided tweets data. First, we use existed perspective scores as a rule-based baseline and with a manual threshold to identify offensive language. Following this, we develop a custom logistic regression classifier, using TF-IDF and n-gram (unigram and bigram) features to capture language patterns indicative of offensive language. Additionally, we experiment with advanced models that incorporate extra features, such as the toxicity score from newest version of Perspective API[1], and test alternative algorithms to improve classification performance. Through these methods, we assess the effectiveness of each approach, explore potential biases, and reflect on the ethical implications of using machine learning for content detection.

2 Analysis of Perspective API Scores

2.1 Analysis on dev.tsv

To establish a baseline for offensive language detection, we used perspective score provided in our dataset which assign a probability indicating the likelihood that a tweet contains harmful language. For the initial analysis, we classified tweets in the dev.tsv dataset as offensive (OFF) if their score exceeded a manually set threshold. The rule-based approach allowed us to evaluate the Perspective API's performance on offensive language detection, using accuracy, precision, recall and F1 score across OFF and NOT labels. Table below summarizes the performance metrics with the 0.8 threshold.

Metric	OFF	NOT	Macro Avg
Accuracy			76.4
Precision	88.6%	74.7%	
Recall	33.4%	97.9%	
F1 Score	48.5%	84.7%	66.6%

The threshold yielded an overall accuracy of approximately 76.4% on the dev.tsv dataset. However, while the model achieved high precision for the OFF class, the low recall indicates it missed a significant number of offensive tweets, suggesting that many neutral or less offensive tweets were accurately classified as non-offensive, but many offensive tweets were left undetected.

1 https://www.perspectiveapi.com/

To try improve recall for offensive content, we experimented with a lower threshold, effectively classifying more tweets as OFF. Here is the table when using a threshold of 0.5:

Metric	**OFF**	**NOT**	**Macro Avg**
Accuracy			80.4%
Precision	69.8%	86%	
Recall	72.5%	84.3%	
F1 Score	71.1%	85.2%	78.1%

As in the table, multiple metrics improved, especially the recall on OFF class balancing out to 72.5% and the overall F1 score rising to 78.2%. This improvement highlights an important distinction: while toxicity and offensiveness are related, they are not synonymous. Lowering the threshold captured more offensive tweets, but it also caused an increase in tweets classified as OFF that were less toxic, revealing the difficulty in defining offensive content solely based on toxicity scores.

2.2 Analysis on mini_demographic_dev.tsv

We further evaluated the model using mini_demographic_dev.tsv dataset, which contains only non-offensive tweets labeled by demographic group. This analysis helped assess the False Positive Rate (FPR) across demographics, indicating the proportion of tweets mistakenly labeled as offensive. At the 0.8 threshold, the FPR results were as follow:

Demographic	FPR
White	7.3%
Hispanic	10.1%
African American	18.9%
Other	1.2%

This data shows a higher FPR FOR African American and Hispanic tweets compared to other groups, suggesting a potential bias in the Perspective API's classification. When we lowered the threshold to 0.5, the FPR increased across all demographic:

Demographic	FPR
White	15.2%
Hispanic	17.3%
African American	27.4%
Other	5.9%

These results highlight an important concern. Lowering the threshold made the model more prone to misclassifying tweets from all demographics as offensive, with a particularly pronounced effect for African American and Hispanic groups. This reinforces the idea that toxicity scores alone may introduce demographic biases, particularly when using a single threshold across diverse language styles.

2.3 Reflection

These results indicate that the Perspective API is more attuned to detecting general toxicity than targeted offensive language. This observation helps explain potential challenges with custom models: if the dataset leans toward labeling more neutral or less toxic tweets as offensive, it may be challenging for a model to generalize well, especially when distinguishing subtle cases of offensiveness.

3 Basic Custom Offensive Language Classifier

To improve upon the baseline established with the Perspective API, we developed a custom offensive language classifier using logistics regression with TF-IDF features. This model aimed to capture specific word patterns and phrases associated with offensive content while exploring potential better performance.

3.1 Feature Design and Model Choice

We used TF-IDF vectorization with unigrams and bigrams to capture individual words and word pairs, allowing the model to recognize nuanced language patterns indicative of offensive content. This feature design was chosen because unigrams and bigrams provide a flexible, interpretable representation of language, enabling the model to detect both individual offensive words and contextually offensive phrases, as well as it can reduce the impact of not important word when training the model.

For the classification model, logistic regression was selected due to its interpretability and effectiveness in handling high-dimensional, sparse text data. Additionally, logistic regression's linear nature allows for a straightforward understanding of feature importance, helping us identify which words and phrases most strongly influence predictions.

3.2 Model Fine-Tuning

To optimize the model, we fine-tuned key hyperparameters, including the regularization strength **(C),** which controls the penalty applied to feature weights. A higher **C** value reduces regularization, making the model more flexible, while a lower **C** increases regularization, discouraging overly complex feature patterns. By experimenting with C values, we found that a value of 1.5 provided a balance that minimized both false positives and false negatives. Additionally, we used class_weight='balanced' to account for the class imbalance in the dataset, helping the model learn to detect the minority OFF class more accurately.

3.3 Model performance on dev.tsv

Evaluated on the dev.tsv dataset, our logistic regression model demonstrated improved balance in performance compared to the Perspective API's baseline. Table below shows the performance metrics:

Metric	**OFF**	**NOT**	**Macro Avg**
Accuracy			75.2%
Precision	64.4%	79.8%	
Recall	57%	84.3 %	
F1 Score	60.5%	82 %	71.2%

3.4 Model Performance on mini_demographic_dev.tsv

We also evaluated the custom model on the mini_demographic_dev.tsv dataset, which contains only non-offensive tweets labeled by demographic group. This test allowed us to analyze the False Positive Rate (FPR) across demographics. The FPR results are as follows:

Demographic	FPR
White	18.9 %
Hispanic	19.7%
African American	28.3%
Other	0%

Similar to the Perspective API baseline, the custom model exhibited higher FPR for African American and Hispanic tweets, indicating a potential bias in detecting language style more prevalent in these group.

3.5 Reflection on Top Features

To further understand the model's decision-making process, we examined the top 100 features (unigrams and bigrams) with the highest coefficients for predicting the OFF label as shown in the picture below:

Most of these features were straightforward curse words or slurs, emphasizing a strong correlation between offensive language detection and high toxicity. This aligns with observations from the baseline model, reinforcing that offensive language detection often relies heavily on identifying toxic words. However, this approach may overlook less toxic but offensive content, highlighting the challenge in creating a classifier that captures a broader range of offensive language.

4 Advanced Model Analysis and Results

To explore the potential whether incorporating additional context could enhance offensive language detection, we developed an advanced model that included the newest and current version of Perspective API's toxicity score as an extra feature alongside the TF-IDF vectors. The toxicity score provided an independent assessment of the likelihood of harmful content, which we hope would complement the information captured by the TF-IDF features.

4.1 Model Design

The advanced model used the same logistic regression setup as our baseline custom model because it is tested the best model for our task, with the addition of toxicity_score as a feature. The TF-IDF vectors were combined with the toxicity score to create a richer feature set, aiming to capture both explicit word patterns and a pre-trained measure of toxicity by calling function provided by Perspective API. We hoped that the model would leverage this additional information to improve performance in detecting offensive content.

4.2 Result on dev.tsv

Despite incorporating the toxicity_score feature, the advanced model did not outperform the baseline custom model as expected. The following table summarizes the performance metrics of the advanced model:

Metric	OFF	NOT	Macro Avg
Accuracy			70.6%
Precision	53.5%	91.2%	
Recall	87.9%	61.9 %	
F1 Score	66.6%	73.8%	70.2%

Although the advanced model achieved higher recall for the OFF class, it came at the cost of precision, lowering the overall accuracy and F1 scores compared to the baseline custom model. This suggests that adding the toxicity_score may have introduced some redundancy, as the TF-IDF features already capture much of the offensive language information. Since toxicity and offensiveness are related but not identical, the model may have struggled to balance these two signals, leading to overestimation of offensive language in some cases.

4.3 Reflection and Implications

The results from the advanced model suggest that while toxicity scores provide valuable context, they do not necessarily improve offensive language detection when used alongside TF-IDF features. This outcome aligns with our findings from examining the top features of the baseline model, where many high-coefficient terms corresponded to straightforward curse words. The challenge lies in distinguishing offensive content that is less overtly toxic or relies on context, which cannot be fully captured by toxicity scores alone. This reinforces the importance of carefully selecting features in offensive language detection models to avoid introducing bias or redundancy.

5 Conclusion and Thoughts

Our findings in this assignment reinforce the need for careful consideration in defining offensive language as well as features designing. Our model's reliance on toxic language showed a tendency to misclassify certain more neutral or less toxic speech, highlighting the risk of being unable to detect certain offensive speeches.

Besides, the cost of misclassification is substantial: high FPR can result in censorship or unfair penalization, especially for marginalized groups.

Finally, this project underscores that while machine learning offers powerful tools for content moderation, deploying these models responsibly requires ongoing evaluation of bias, transparency, and adaptability to different social and linguistic contexts. Future work could benefit from more context-aware models and techniques to detect nuanced forms of offensive language while minimizing unfair impacts on specific communities.