CSE 3000 ML Spam and Hate Detection Report

David Chinchilla(dec21003), Tahmeed Solaman (tas20013), Trenyce Taylor (tdt20002), Masrur Alam (maa19033)

Con Github: https://github.com/Masrur15/CSE3000-Hate-and-Speech-Detection

Abstract:

The process of training an AI content moderation system is a detailed endeavor that takes into account several aspects such as data preprocessing, training methodologies, dataset choice, performance metrics, and bias analysis. More specifically, our system is designed to moderate spam and hate speech from regular, clean text. This of course brings up many ethical and bias considerations like what counts as hate speech or spam for example. The model is bound to make mistakes due to the nuances of human language, but there are several ways to steadily improve it with feedback in time. While AI content moderation systems in general are typically error prone, it is only a matter of time and more innovation until they are perfect in their task.

Introduction:

In today's world dominated by social media, it's hard not to want to express yourself in one manner or another. The sheer number of people that have access to these social media platforms means that there are bound to be bad actors that have bad intentions, whether it's spewing hate speech or simply spamming random messages. This of course creates a need for robust and adaptive systems to moderate the content with which the general public can see and interact with on these social media platforms. In theory, a human could moderate all these messages to ensure they are "clean", but this is not feasible as it not only would take a long time to process but perhaps have long term psychological effects as well. Therefore, our solution is to train an AI model to process these messages as it can do so with relative ease when compared to other solutions.

Methods:

Our AI content moderation model is built to classify text as 1 of 3 things, either hate speech, spam, or clean text. It does this by being trained on 2 models primarily. For hate speech, we use a dataset called "Learning From The Worst" which is made up of 41,000 entries with 53% of those entries being hate speech. It also has 6 classes of hate speech as well to further distinguish the data. When it comes to spam detection, we trained our model on a dataset called "SMS Spam Collection" consisting of 452 spam messages and 1002 ham (non-spam) messages.

We start off by preprocessing the data by stripping it of any special characters like punctation, numbers or symbols. This data is then split 80% for training and 20% for testing. We then use a TD-IDF vectorizer function to fit and transform the training and test data. The purpose of this vectorizer is to transform the raw text data into a matrix of numerical features based on the importance of each word in a phrase relative to both datasets. To make it simpler, we intentionally ignored common English words such as "like, has, is" because these would typically have the highest weights as they are used to construct sentences. After transforming the data, we train a logistic regression model on it. Once trained, the model can predict the labels for phrases, as well as compute several metrics including accuracy (how often the model is correct), precision (how many of the predicted positive labels are correct), and F1 score (a balance between precision and recall).

Results:

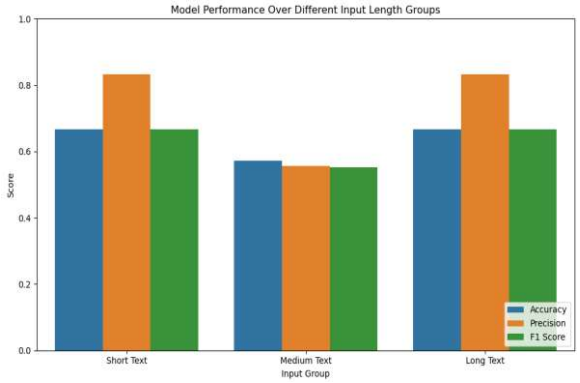

Figure 1: We found that our model predicted the best for short text (less than 6 words) and long text (more than 10 words), whereas it performed the worst for medium text.

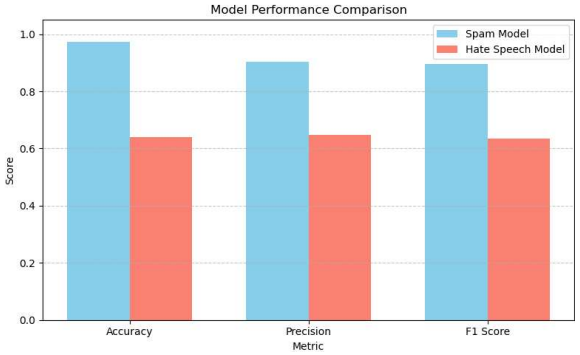

Figure 2: Comparison of the spam and hate dataset's accuracy, precision, and F1 score. We found that our spam dataset has better scores than the hate one

Discussion:

Bias Analysis: Since learning models are primarily trained on the data that they are given, both the spam and hate speech datasets can have biases. Sampling bias is most likely used in the spam dataset since they are from SMS messages. This inherently includes only people who have access to mobile devices that they can text from, and isolates people who don't have access to such technology. It also focuses more on English-speaking countries, which would not give us a smaller sample size and would not represent the global population. The SMS spam dataset forces the model to associate certain words like "win", "click", "text", and other suspicious words that might get flagged as spam in legitimate messages.

The hate speech dataset can likely be based on messages from popular platforms such as Twitter and Reddit, which have their biases as well, since they attract a particular audience who find their platforms entertaining. Hate speech datasets can have cultural and gender biases as well, since most hate speech can be targeted at various racial and gender groups, which often consists of the use of slurs and other derogatory terms that people find offensive, depending on the person's beliefs and background.

Limitations: Both models can memorize keywords and phrases rather than generalizing to new inputs since we are only feeding the model two different datasets, which can cause inaccuracies in the classification of messages. The hate speech model has limitations because it can't detect joking and sarcastic messages, as it usually just finds the keywords that match the criteria for hate speech and doesn't detect any nuances in the English language. In spam detection, wrongly labeling a genuine message as spam can harm communication. In hate speech detection, failing to flag toxic content can lead to real-world harm. Our models are able to classify messages correctly if the message is just a regular sentence while longer or shorter texts can cause it to inaccurately identify hate speech or spam texts.

Ethical Concerns: False positives can lead to dangerous consequences, as flagging a legitimate message could cause a user to not see an important message and get reported as spam. There's also the risk of messages being censored because an automated hate speech detection tool can find a discussion being made between two users that is controversial but is still valid and is not violating any hate speech policies, while the system views the comments being made as hateful. The system that usually classifies these messages as hateful or spam never really tells users why a message was taken down, and the user can never appeal the decision made by the system to take down their message.

Improvements: The SMS spam dataset and the hate speech dataset could be more evenly distributed since the spam dataset has more ham messages and not enough spam ones while the hate speech dataset has more clean messages than hate speech. Instead of using Linear Regression and TD-IDF vectorizer, we can use transformer-based models such as Bert. This would help understand the meanings of words, especially in the context of finding sarcasm and insulting tones.

Conclusion:

The spam model's performance metrics performed very well compared to our hate speech model that did poorly. The spam model was able to correctly identify clear spam messages such as promotional offers and scam messages while also correctly identifying regular messages as non-spam. This performance could be based on many things like the model being able to find patterns within the sentence and the SMS spam dataset being a large enough sample size to provide us with various keywords, phrases, and sentences. The hate speech model was able to correctly identify very extreme hate speech that included vulgar language and derogatory terms but fails to catch borderline ones or even slightly offensive speech as well. It correctly classifies false positives which have negative words and phrases in the messages, but that didn't go to the extent of hurting or degrading anyone. Overall, these results show that choosing the right dataset and model can help produce a successful AI content moderation system.

Reference list:

Hate Speech Dataset: https://github.com/bvidgen/Dynamically-Generated-Hate-Speech-Dataset
SMS Spam Dataset: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset