Data Glacier

Your Deep Learning Partner

Exploratory Data Analysis

Hate Speech Detection using Transformers

Yash Parmar: yparmar2024@gmail.com

June 16th, 2025

Agenda

Executive Summary

Problem Statement

Approach

EDA

EDA Summary

Recommendations

Executive Summary

Social media companies often have to rely on manual moderation to identify and delete hateful tweets, making the process time-consuming and labor-intensive. While automated systems help flag potentially harmful content, they often struggle with context and nuance, leading to false positives and negatives. This means human moderators must review flagged tweets, which slows down the process and limits scalability. Moreover, when tweets that are not actually hateful get flagged or removed, it can frustrate users and lower customer satisfaction. Therefore, accuracy in detecting hate speech is just as important as speed, as both impact user trust and the platform's credibility.

Problem Statement

To address the inefficiencies of manual moderation, social media platforms can automate the process through Twitter hate speech detection using machine learning and natural language processing (NLP) techniques. By training models on large datasets of labeled tweets, these systems can learn to identify patterns and language associated with hate speech. Once deployed, such models can quickly scan vast volumes of tweets in real time, flagging or removing harmful content with minimal human intervention. This not only speeds up moderation but also helps maintain a safer online environment. However, to avoid false positives that could hurt user experience, the model must be highly accurate and continually updated to adapt to evolving language and context.

Approach

To begin solving the problem, I analyzed and cleaned the data to understand its structure before performing EDA. Here are the steps I followed:

1. Checked for missing/null values to ensure no empty tweet entries
2. Created a cleaning function to standardize and remove any unnecessary symbols and patterns to mess with model evaluation
3. Converted all tweet text to lowercase to ensure consistency between tokens
4. Removed special characters and extra whitespace to reduce token noise which will not contribute to semantic meaning
5. Stored clean tweets in a new column to ensure preservation of prior data
6. Identified a class imbalance in the training dataset

Exploratory Data Analysis

To begin the exploratory data analysis (EDA), I prepared several text-based evaluations to better understand the dataset. They are listed below:

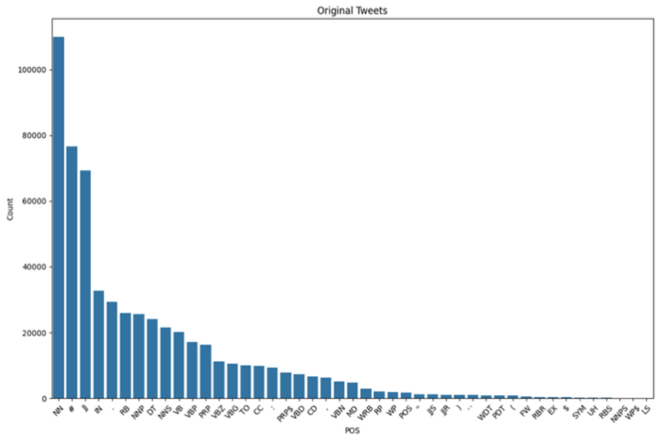
- **POS Tagging:** Used to identify patterns in parts of speech that may be prominent in hate vs. non-hate speech. This was performed on the original tweets, cleaned tweets, as well as separately on cleaned hate and non-hate tweets.
- **Average Word Count:** Measured before and after cleaning to assess the impact of noise and redundant text on tweet length.
- **N-gram Analysis:** Conducted unigram, bigram, and trigram analysis on original tweets, cleaned tweets, and separately on cleaned hate and non-hate tweets to identify the most frequently occurring word combinations.
- **Word Clouds:** Created for original, cleaned, hate, and non-hate tweets to visualize the most common words in each category and highlight thematic differences.

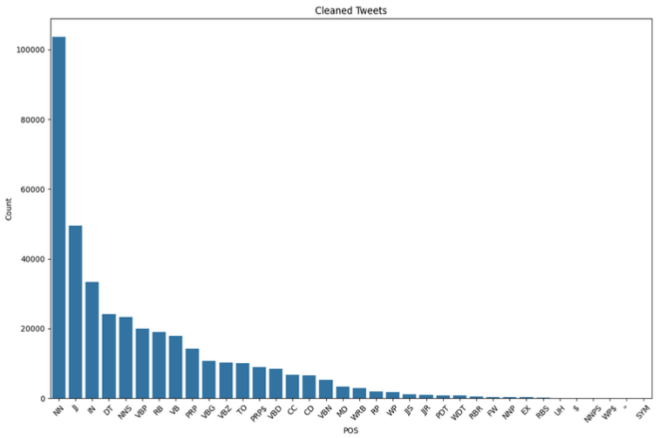
Exploratory Data Analysis – POS Tagging

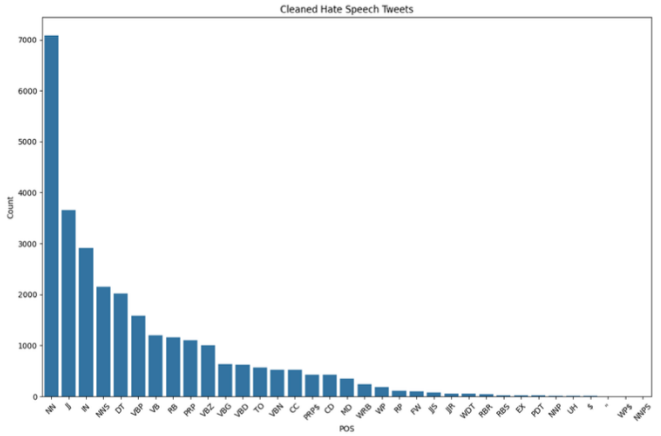
Observations:

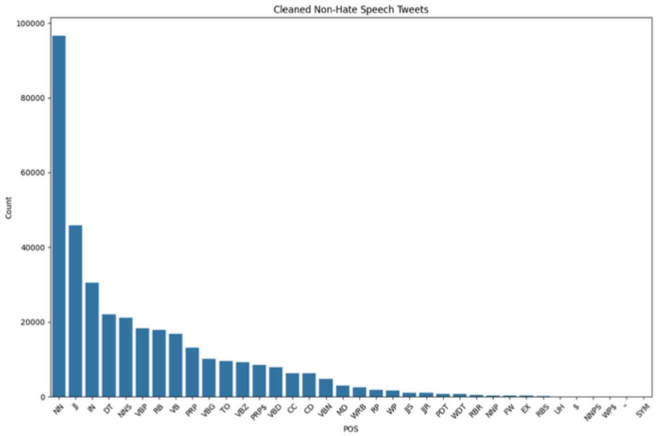
- The most frequent POS tag across all datasets was **NN** (singular noun), which is expected since tweets typically refer to people, topics, or things.
- In the **uncleaned tweets**, **hashtags (#)** were the second most frequent tag, which makes sense given they are common on social media.
- In the **cleaned tweets**, hashtags were removed, and **JJ** (adjectives) became the second most common. This shift reflects the descriptive nature of language in tweets, particularly to express emotion or sentiment—either hateful or non-hateful.
- Other tags like **IN**, **NNS**, and **DT** (prepositions, plural nouns, and determiners) supported sentence structure.

Conclusion: Cleaning effectively removed noise (e.g., hashtags), enhancing the presence of meaningful parts of speech like nouns and adjectives, which are critical for identifying sentiment and hate speech.


Original Tweets


Cleaned Tweets


Cleaned Hate Speech Tweets

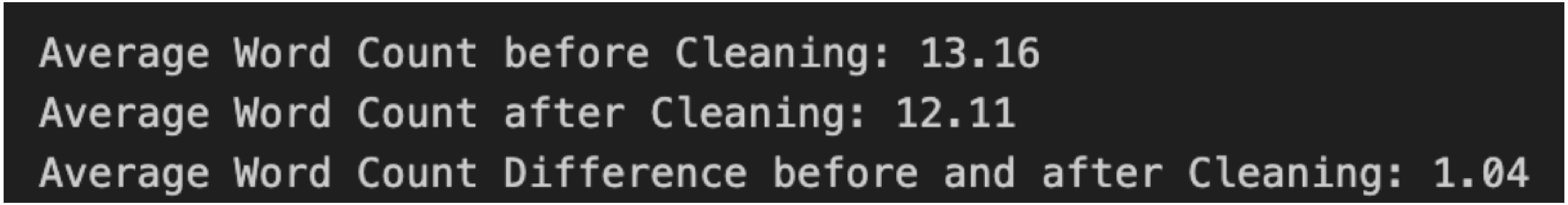

Cleaned Non-Hate Speech Tweets

Exploratory Data Analysis – Average Word Count

Observations:

- On average, **one word was removed per tweet** after cleaning.
- These removed words were typically **user mentions (@user)**, **links**, or **special characters**—elements not useful for sentiment analysis.

Conclusion: The cleaning process reduced noise without removing important sentiment-bearing content, supporting cleaner and more reliable tweet analysis.



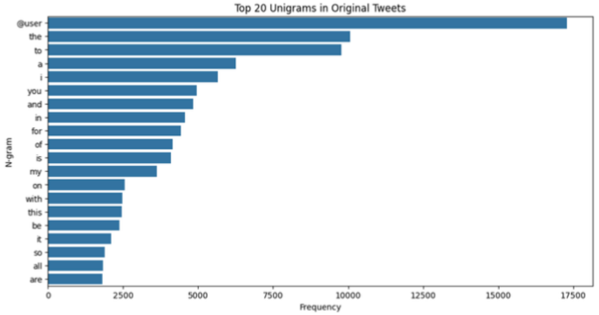
```
Average Word Count before Cleaning: 13.16
Average Word Count after Cleaning: 12.11
Average Word Count Difference before and after Cleaning: 1.04
```

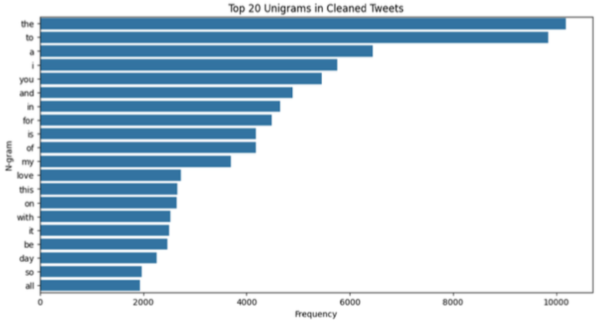
Exploratory Data Analysis – N-gram Analysis

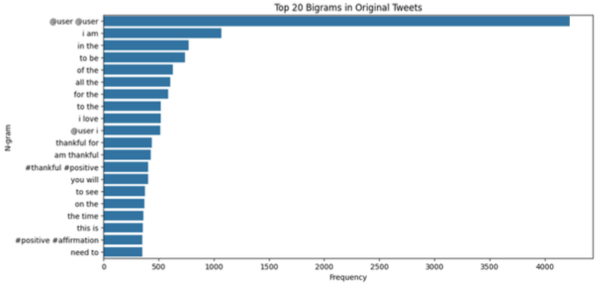
Observations:

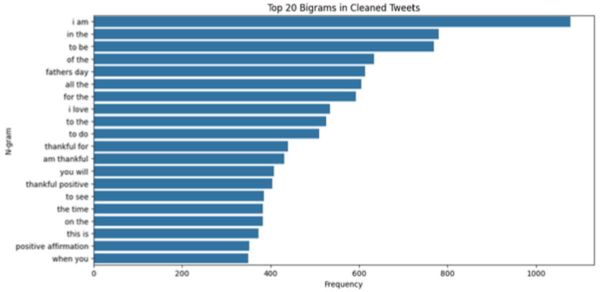
- Original tweets are dominated by **“@user”**, which appears heavily in unigrams, bigrams, and trigrams.
- **Hyperlinks** and other non-textual tokens often appear in n-gram results, masking real linguistic patterns.
- After cleaning, common n-grams shift to meaningful phrases such as **“i am”**, **“i am thankful”**, and **“my life”**, indicating expressions of emotion or stance.

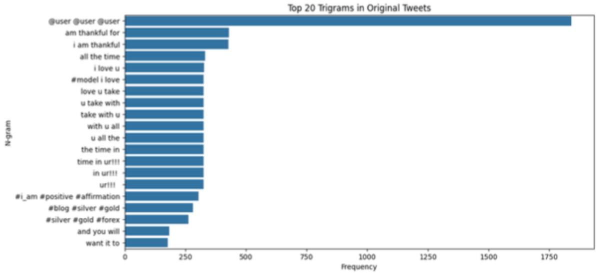
Conclusion: The cleaning process successfully removes noise that dominates original n-grams and uncovers valuable textual signals that are more indicative of tweet content and tone.

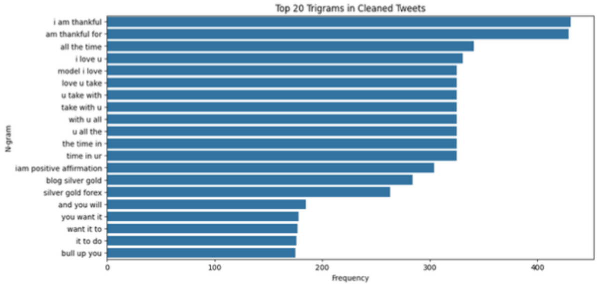

Top 20 Unigrams in Original Tweets


Top 20 Unigrams in Cleaned Tweets


Top 20 Bigrams in Original Tweets


Top 20 Bigrams in Cleaned Tweets


Top 20 Trigrams in Original Tweets

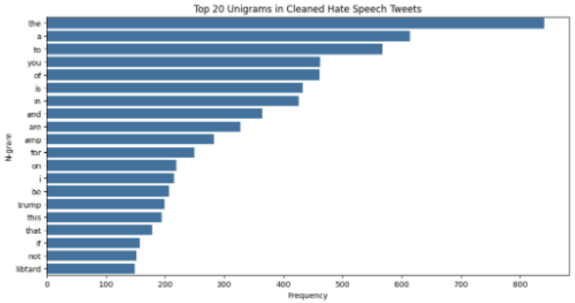

Top 20 Trigrams in Cleaned Tweets

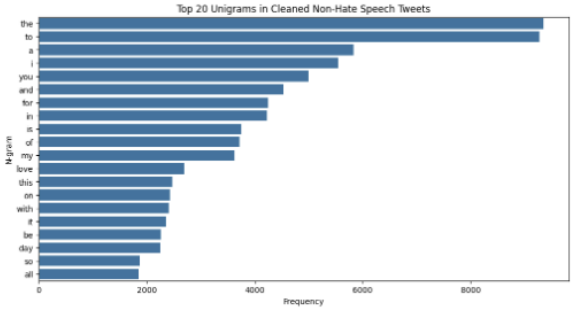
Exploratory Data Analysis – N-gram Analysis

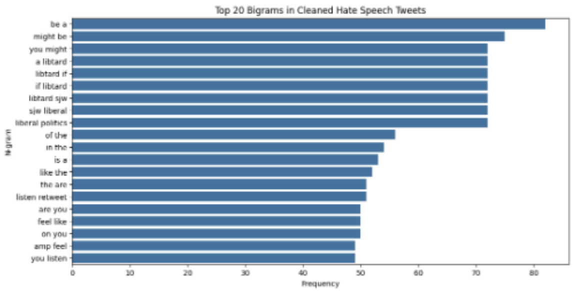
Observations:

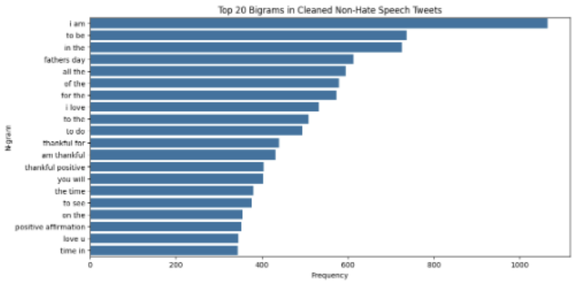
- Cleaned **hate speech tweets** frequently include targeted phrases like **“be a”** or **“you might be”**, often structured as insults or derogatory statements.
- Cleaned **non-hate speech tweets** commonly include **neutral or reflective language**, such as **“i am thankful”** or **“my life”**.
- Hate speech n-grams tend to be more **aggressive or provocative**, while non-hate speech is **more varied and often positive or neutral**.

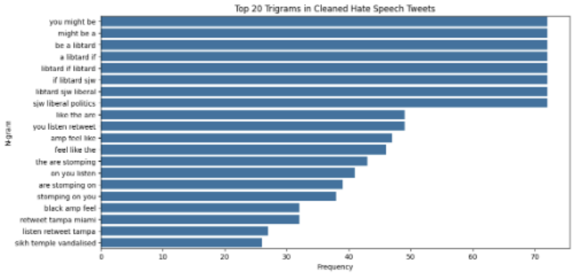
Conclusion: N-grams help differentiate aggressive tone and structure in hate speech from reflective or benign language in non-hate speech, supporting model explainability and targeted feature selection.

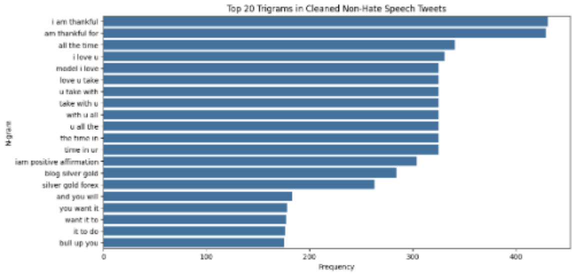

Top 20 Unigrams in Cleaned Hate Speech Tweets


Top 20 Unigrams in Cleaned Non-Hate Speech Tweets


Top 20 Bigrams in Cleaned Hate Speech Tweets


Top 20 Bigrams in Cleaned Non-Hate Speech Tweets


Top 20 Trigrams in Cleaned Hate Speech Tweets

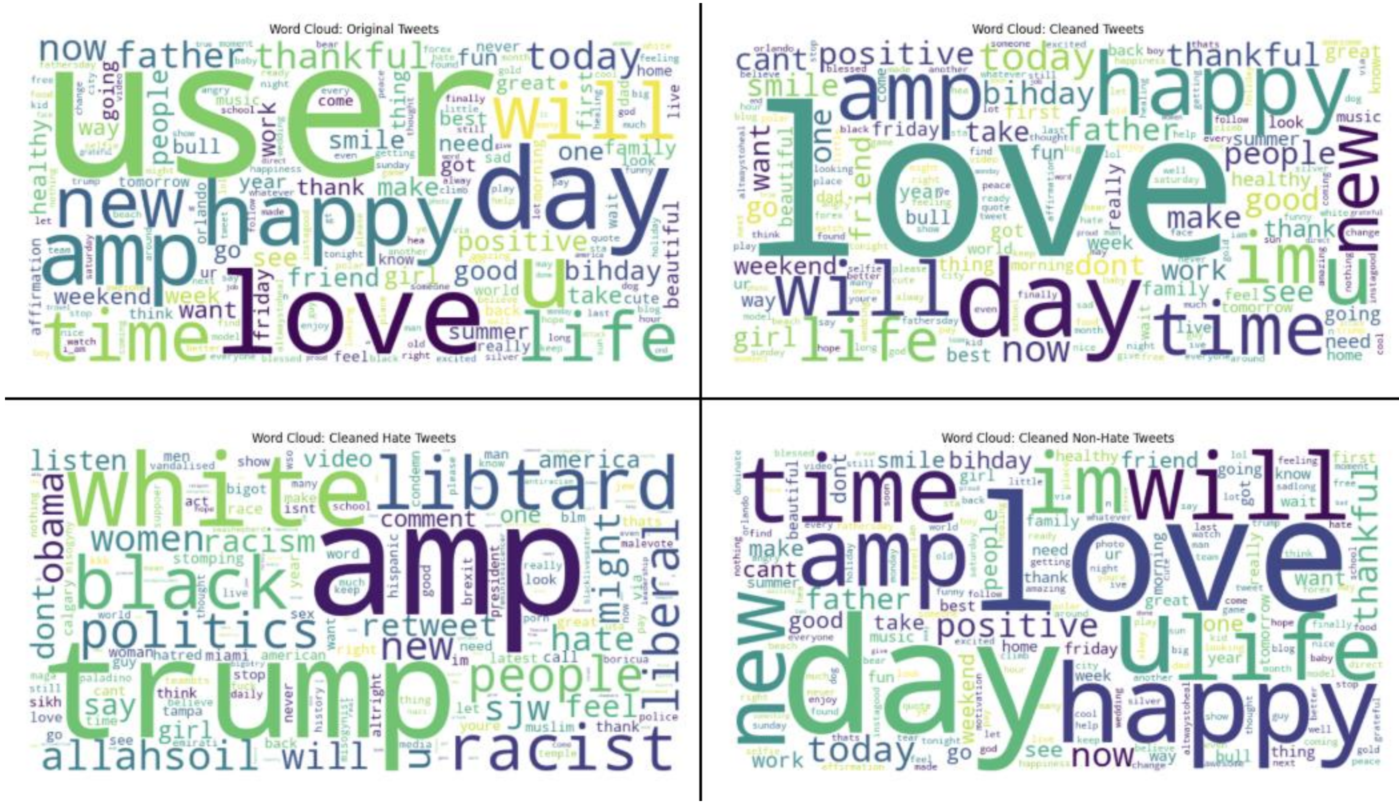

Top 20 Trigrams in Cleaned Non-Hate Speech Tweets

Exploratory Data Analysis – N-gram Analysis

Observations:

- **Original tweets** showed high frequency of noise like "@user."
- **Cleaned tweets** emphasized words like "love," supporting that noise was successfully removed.
- **Cleaned hate speech tweets** contained offensive, political, or racially charged words.
- **Cleaned non-hate tweets** looked similar to general cleaned tweets but had slightly less emphasis on the word "love."
- Notably, **"love" appeared in both hate and non-hate tweets**, sometimes sarcastically.

Conclusion: Word clouds confirmed the class imbalance and sarcasm challenges in the dataset. Some sentiment-laden words (e.g., "love") appeared in hate speech, complicating interpretation and classification.



Exploratory Data Analysis Summary

- Tweet Length and Word Count:
  - Most tweets are very short, with under 20 words on average.
  - Short length limits context – local features may dominate.
- Dominance of Special Tokens Pre-cleaning:
  - The raw dataset is full of non-linguist tokens like “@user” and hyperlinks.
  - After cleaning, text becomes clearer but remains simple and informal.
- N-gram Analysis:
  - Cleaned hate speech contains targeted phrases and insulting constructs.
  - Non-hate speech often contains emotionally reflective or personal language.
- Word Patterns & Simplicity:
  - Tweets contain repetitive language and limited semantic nuance.
  - Most tweets convey sentiment or aggression in a few key words or phrases.

Recommendations (Technical Summary)

TF-IDF + Linear SVM:

Strengths:

- Fast, lightweight, and easy to implement
- Highly interpretable – feature importance is clear
- Works well with short, keyword-heavy texts like tweets

Limitations:

- Ignores word order and semantic context
- Struggles with sarcasm or implicit hate
- Performance drops if offensive terms are masked or rare

BERT Embeddings:

Strengths:

- Captures word meaning in context, even with subtle phrasing
- Handles slang, typos, and informal language well
- Strong performance on complex or less structured text

Limitations:

- Computationally expensive and slower to train
- Harder to interpret model decisions
- May overfit or underperform on very short, direct text

Recommendations (Business Summary)

The tweets in our dataset often contain sarcasm, coded language, and slang–all of which make hate speech harder to detect using traditional methods. TF-IDF + Linear SVM relies on exact keywords and fails to capture the context or intent behind indirect or sarcastic expressions.

To address this, we recommend using **BERT embeddings**, which better understand contextual meaning, informal language, and slang usage. BERT’s ability to capture the subtlety and nuance of hate speech makes it a more reliable choice for this detection task.

This approach enhances both accuracy and fairness in identifying hate content that may be disguised through indirect or informal expressions.

Thank You

Data Glacier

Your Deep Learning Partner