SENG 550 L02 Project Report

YouTube Comment Sentiment Analysis for Big Data

Submission: December 19, 2024

Ammaar Melethil | 30141956

Nathan Campbell | 30179708

Aria Sharifnia | 30170058

Kofi Frempong | 30054189

Table of Contents

Preamble

Name	Major Contributions	Contribution %	Initials for signature
Nathan Campbell	I conducted exploratory data analysis (EDA), including visualizing class distributions and text length patterns. I also played a key role in generating insights for the results section and finalizing the paper's conclusion.	23.3%	NC
Aria Sharifnia	I focused on the data preprocessing steps, including stopword removal, lemmatization, and feature extraction using CountVectorizer and TF-IDF. Contributed to writing the methodology section and refining the analysis of the results.	23.3%	AS
Kofi Frempong	I managed the integration of the PySpark framework and addressed class imbalance using weighting techniques. I also contributed significantly to the introduction, abstract, and revisions for coherence and clarity across the paper.	23.3%	KF
Ammaar Melethil	I focused on implementing the machine learning pipeline, including Logistic Regression model training, cross-validation, and hyperparameter tuning. Authored the performance metrics subsection and helped integrate feedback into the final paper.	30%	AM

Github repo: https://github.com/ammaarmelethil/YT-Sentiment-Analysis-for-Big-Data

Abstract

Our project focuses on sentiment analysis of YouTube comments, classifying them as negative, neutral, or positive. We analyzed comments from diverse videos, including Logan Paul's apology and music videos. Our data engineering pipeline employed NLP techniques like stopword removal, lemmatization, and feature extraction using CountVectorizer and TF-IDF. To address class imbalance, we applied class weighting and remapped labels for PySpark MLlib compatibility. A Logistic Regression model achieved a cross-validated test accuracy of **90.57%**, with high precision scores: 95.67% for negative, 84.19% for neutral, and 98.37% for positive sentiments. Exploratory analysis revealed that negative comments tend to be slightly longer than positive ones. Preprocessing refined the feature space by removing irrelevant terms, further enhancing performance. By combining scalable data engineering with machine learning, we effectively performed sentiment analysis. Our findings provide valuable insights for content creators, researchers, and platform moderators, offering a scalable framework to understand audience sentiment and improve engagement on online platforms.

Introduction

YouTube, one of the most widely utilized platforms for user-generated content, generates millions of comments daily. Effectively interpreting the sentiment embedded within these comments is essential for enhancing user experience, detecting toxic behaviour, and deriving actionable insights for content creators. Accordingly, this project centers on the sentiment analysis of YouTube comments, specifically classifying each comment into one of three categories: Negative (-1), Neutral (0), or Positive (1). For training with PySpark's machine learning library, these categories are remapped to 0, 1, and 2, respectively. The overarching goal is to achieve accurate sentiment predictions while identifying nuanced behavioural patterns across a set of video types.

The underlying dataset for this investigation includes pre-labelled YouTube comments sourced from five highly popular videos, each surpassing 10 million views, encompassing a range of thematic

content. To ensure a breadth of sentiment and content diversity, the videos selected represent controversial and polarizing topics, including Logan Paul's apology video, an OKGO (popular band) music video, coverage of the Royal Wedding, a Taylor Swift music video, and politically charged commentary on Donald Trump. The dataset's pre-labelled sentiments serve as a robust foundation for supervised learning techniques, enabling both fine-grained quantitative analyses and qualitative interpretations of language use.

Sentiment analysis of YouTube comments offers significant value for multiple stakeholders. Content creators and community moderators can leverage insights derived from sentiment analysis to understand audience reception, detect instances of inappropriate or harmful discourse, and potentially guide moderation policies. YouTube, as a platform, might employ these insights to refine automated filtering strategies and inform policy adjustments, while commercial stakeholders, such as advertisers and branding teams, could derive valuable indicators regarding public engagement and sentiment trends. Prior work in this domain has frequently adopted natural language processing (NLP) methodologies, ranging from traditional representations like bag-of-words and TF-IDF to contemporary deep learning models based on LSTMs or Transformers, to classify sentiment. However, existing literature often focuses on broad-level distributions or frequently occurring terms, overlooking more nuanced research questions. These may include, for instance, the relationship between comment length and sentiment or the dynamics of sentiment variation across distinct categories of video content.

This project aims to address these understudied dimensions. By examining sentiment across diverse and, at times, divisive content, we move beyond binary classifications to consider relational factors such as the correlation between comment length and sentiment, as well as the identification of weighty words indicative of particular sentiment classes. To mitigate issues related to class imbalance, specifically the overrepresentation of neutral comments, class weighting strategies are applied to ensure more equitable model training. Key research questions guiding this inquiry include: **Can the sentiment of YouTube comments be reliably predicted using preprocessed textual data? Which common words**

are strongly associated with each sentiment category? and **Does the length of a comment correlate with its sentiment score?**

To rigorously explore these questions, the study employs a data engineering pipeline constructed with PySpark and established NLP methods. This pipeline encompasses data ingestion, text preprocessing, exploratory data analysis, feature engineering, and model training, while explicitly addressing the challenges of class imbalance and noisy textual inputs. The results yield several findings: E.g. negative comments tend greater length than their positive counterparts, and the vocabulary strongly associated with each sentiment class furnishes meaningful context for understanding user behaviour. The Logistic Regression model developed in this project achieved a test accuracy of **90.57%**, underscoring its effectiveness and resilience.

Methodology

Exploration of data features and refinement of feature space

The initial phase of the analysis involved loading the datasets into a PySpark DataFrame and performing preliminary data cleaning operations. Specifically, comments lacking valid text or sentiment labels were removed. These datasets, sourced from five distinct YouTube videos encompassing diverse and frequently antagonistic content such as Logan Paul's apology video, OKGO's music video, and political commentary involving Donald Trump, were subsequently merged into a unified data frame. Exploratory Data Analysis (EDA) was then conducted to identify patterns and challenges within the data.

Initial visualizations of the class distribution (**Appendix 1**) indicated a marked imbalance, with the Neutral class (mapped to class 1) noticeably overrepresented compared to both the Negative (class 0) and Positive (class 2) categories. Such an imbalance introduces inherent challenges in model training, potentially biasing predictions toward the majority class. To mitigate this risk, a class weighting strategy was later integrated into the training process, ensuring a more equitable representation of minority classes.

Subsequent analyses focused on the distribution of text length (**Appendix 2**). While a majority of comments were brief (under approximately 200 characters), no pronounced variations in length emerged among the three sentiment categories. This finding suggested that text length alone was unlikely to skew sentiment predictions or introduce systematic bias, thereby allowing the modelling strategy to focus on more linguistically meaningful features.

Refinement of the feature space began with a comprehensive text normalization protocol. Stopword removal, for example, eliminated high-frequency yet semantically inert words (e.g., "the," "is") that tend not to offer substantive insights into user sentiment. Domain-specific frequency analysis (**Appendix 3**) further identified words such as "paper," "music," "song," and "printer" as both prevalent and contextually irrelevant to the sentiment classification task. These terms were appended to a custom stopword list to reduce noise. Lemmatization was then employed to unify word forms, thereby enhancing the semantic consistency of the corpus (e.g., "running" and "runs" both mapped to "run"). These preprocessing steps were validated by examining token count distributions (**Appendix 4**), which confirmed a reduction in extraneous tokens and improved dataset quality.

Experimental Setup

After normalization, tokenization, and lemmatization, the transformed text was converted into numeric features using CountVectorizer, producing a token frequency matrix suitable for model input. To further enhance the discriminative power of these representations, TF-IDF (Term Frequency-Inverse Document Frequency) weighting was applied. TF-IDF prioritizes tokens that are both meaningful and less commonly used, ensuring that repetitive or uninformative terms exert less influence on classification decisions.

In addition to textual feature engineering, label adjustments were performed to accommodate PySpark MLlib requirements. Sentiment labels originally assigned as (-1, 0, 1) were remapped to (0, 1, 2), ensuring non-negative integer labels compatible with MLlib's standard classification workflows. Finally, class weighting was incorporated to address the significant class imbalance observed during EDA, balancing the contribution of minority classes (Negative and Positive) relative to the majority Neutral

class. This measure aimed to facilitate more equitable training and improve the model's ability to generalize effectively across diverse comment types and sentiment polarities.

Experimentation factors

The experiment utilized a Logistic Regression model for multi-class sentiment classification. Logistic Regression was chosen for its efficiency and interpretability, making it well-suited for large-scale datasets. To optimize the model, k-fold cross-validation was performed rather than a train-test split. Despite being more computationally expensive, k-fold cross-validation provides a more robust performance estimate by using the entire dataset for training and testing, reduces variance in results, and is especially effective for addressing class imbalances and hyperparameter tuning. A parameter grid was used to tune two key hyperparameters.:

1. **Regularization Strength (RegParam)**: Tested values included [0.01, 0.1, 1.0].
2. **ElasticNet Mixing Parameter (elasticNetParam)**: Tested values included [0.0, 0.5, 1.0].

Cross-validation enabled the selection of the best hyperparameter combination, ensuring a balance between overfitting and underfitting.

Experiment process

- **Data Cleaning**: The datasets were cleaned by removing missing values and irrelevant entries, resulting in a unified data frame ready for analysis.
- **Exploratory Analysis**: Iterative EDA was conducted to refine the preprocessing pipeline, addressing challenges like class imbalance and noisy text.
- **Feature Engineering**: Feature vectors were extracted using CountVectorizer and refined with TF-IDF to optimize the representation of the text data.
- **Class Weighting and Label Remapping**: Class weights were applied to counteract the class imbalance, and labels were remapped for compatibility with PySpark's MLlib.

- **Model Training**: The Logistic Regression model was trained using the preprocessed dataset, with cross-validation to identify the best hyperparameters.
- **Evaluation**: The trained model was evaluated on the test dataset, and predictions were analyzed for overall accuracy, as well as precision, recall, and F1-score for each class.
- **Visualization**: The results were visualized using a **confusion matrix heatmap** (Appendix 5), which provided insights into the model's performance across different classes.

Performance metrics

The performance of the Logistic Regression model was evaluated using several metrics to ensure its robustness and reliability:

1. **Overall Accuracy**: The cross-validated test accuracy was **90.57%**, indicating strong generalization performance.
2. **Class-Specific Metrics**:
    - **Class 0 (Negative)**: Precision: 95.67%, Recall: 88.16%, F1-Score: 91.76%
    - **Class 1 (Neutral)**: Precision: 84.19%, Recall: 97.68%, F1-Score: 90.43%
    - **Class 2 (Positive)**: Precision: 98.37%, Recall: 82.30%, F1-Score: 89.62%

Results

While the primary objective of this research was to conduct sentiment analysis on YouTube comments, the findings also illuminate noteworthy patterns in user engagement and comment characteristics. One particular observation pertains to comment length. As illustrated by the text length distribution (**Appendix 2**), the majority of YouTube comments measure fewer than 200 characters, and only a negligible fraction surpasses 500 characters. This likely reflects the short-form nature of YouTube's content, often ranging between 7 and 15 minutes which may motivate users to respond with short commentary rather than extended observations.

Additionally, the analysis uncovered a subtle yet intriguing pattern regarding sentiment-specific differences in comment length. Negative comments exhibited a marginally higher average length than their positive counterparts. Although the precise cause of this divergence remains uncertain, one plausible explanation is that users may invest greater effort in articulating their negative viewpoints, potentially to substantiate their criticisms or to provoke further discussion. This phenomenon suggests that sentiment not only influences the thematic content of comments but may also correlate with their structural properties, underscoring the multifaceted nature of online user engagement.

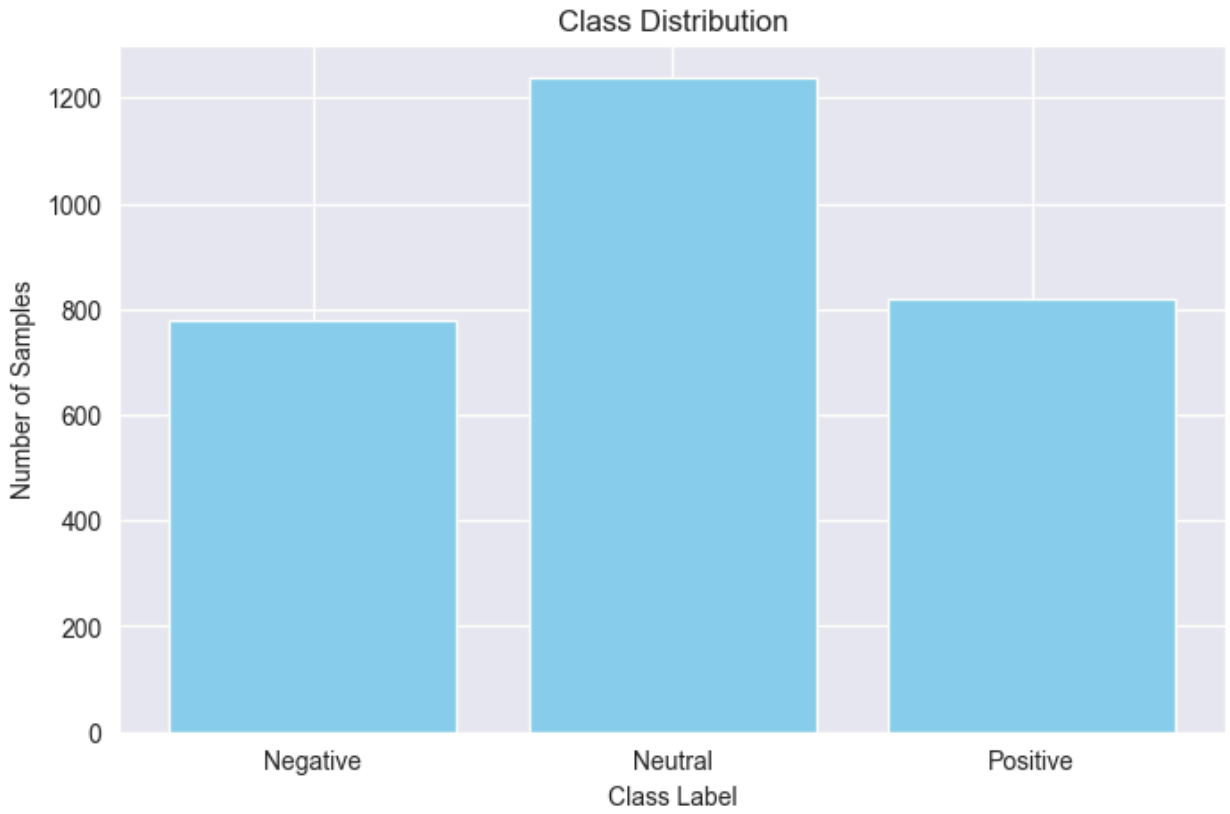
Beyond sentiment considerations, we also examined term frequency distributions (most common words) before preprocessing (**Appendix 3**). Unsurprisingly, "video" emerged as the most frequently occurring word, reflecting the platform's core function. Further, terms like "music" and "song" were prominent, consistent with the inclusion of two music videos in the dataset. More unusual, however, was the prevalence of words such as "paper" and "printer," attributable to the visual content of one featured music video, which integrated hundreds of printers into its aesthetic design. While these domain-specific words accurately represent the content environment, they contributed minimal analytical value to sentiment classification. Hence, they were removed during preprocessing to refine the feature space and enhance modelling performance.

Overall, the project's results confirm the effectiveness of the data engineering pipeline in accurately classifying YouTube comments into distinct sentiment categories. With a cross-validated test accuracy of 90.57%, the Logistic Regression model demonstrated strong overall performance. Evaluation metrics for precision, recall, and F1-score further indicate balanced performance across the three sentiment classes, despite the initial class imbalance. Negative comments exhibited comparatively lower recall but high precision, suggesting that when the model labels a comment as negative, it is highly likely to be correct. Positive comments attained the highest precision, although some were periodically misclassified as Neutral, as evidenced by the confusion matrix (**Appendix 5**). This pattern may stem from linguistic similarities between Positive and Neutral sentiments.

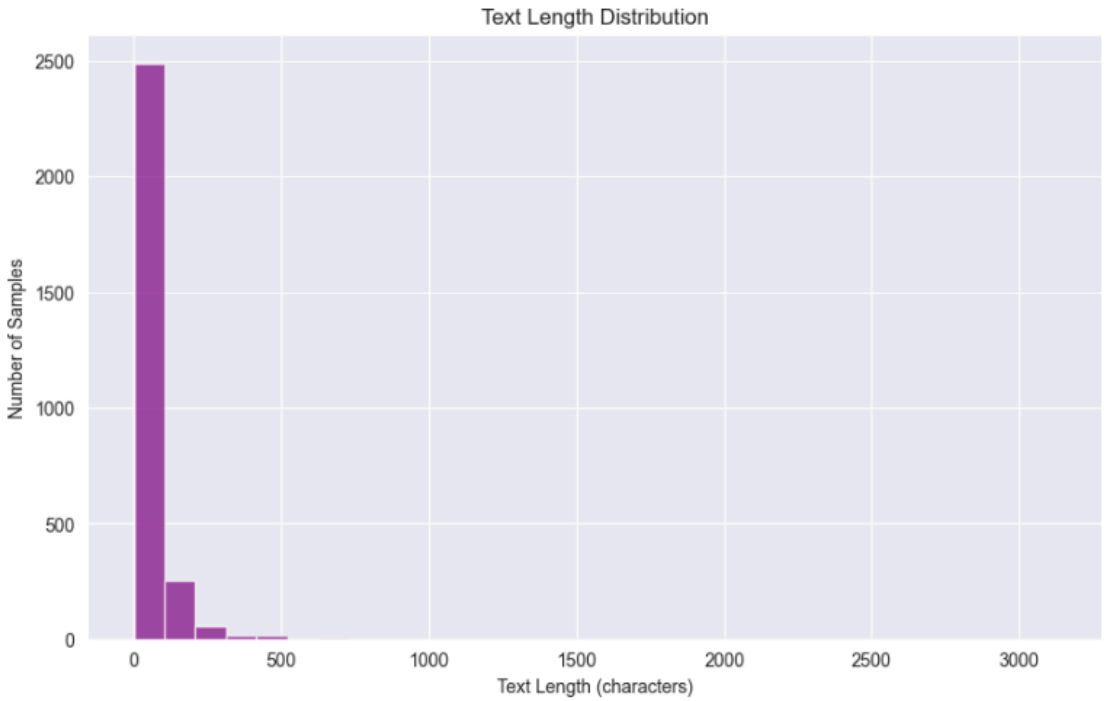
In conclusion, this project demonstrates the potential of combining scalable data engineering techniques with machine learning to analyze and classify sentiments in user-generated text. The findings from this study provide actionable insights for researchers, content creators, and platform moderators, paving the way for improved user engagement and sentiment understanding on platforms like YouTube. With further refinements such as more detailed stopword removal, and exploring more advanced models like LSTMs or transformers, this pipeline can serve as a scalable and extensible framework for sentiment analysis in broader applications.

Appendix

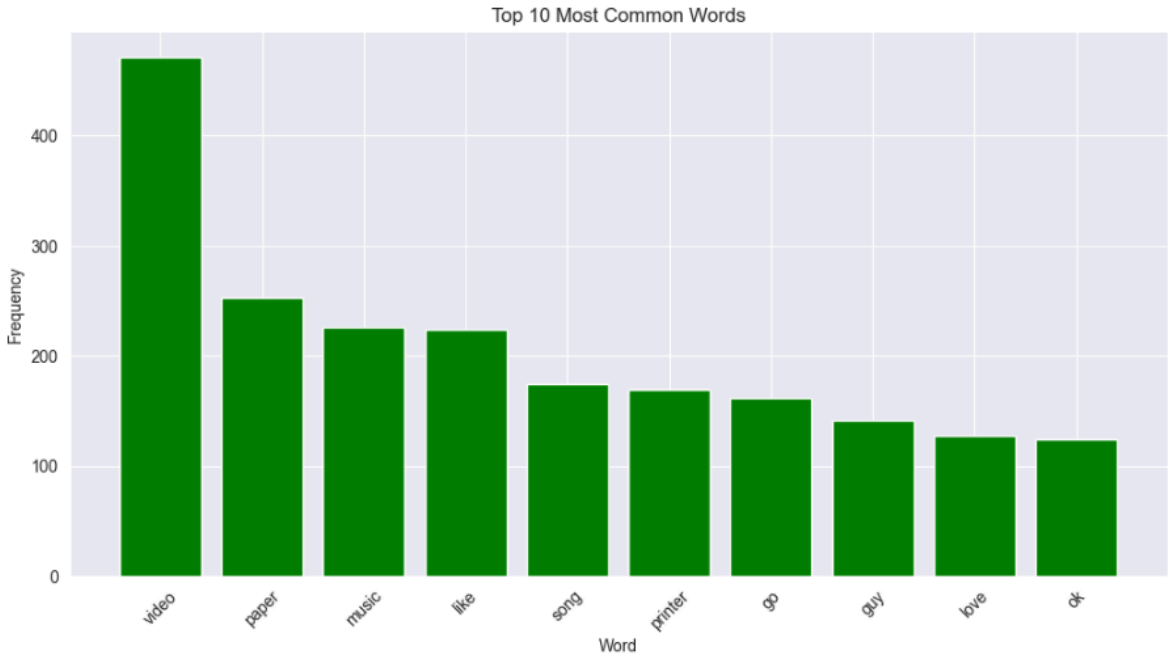
Appendix 1



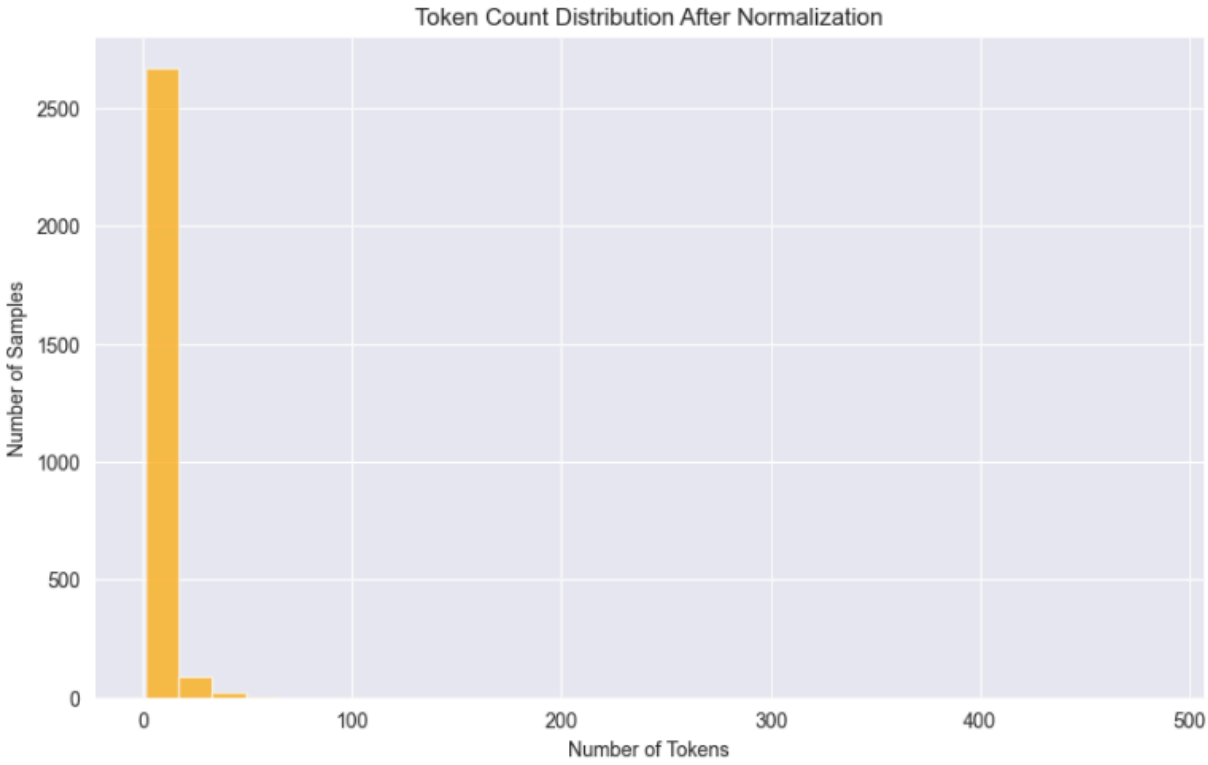
Appendix 2



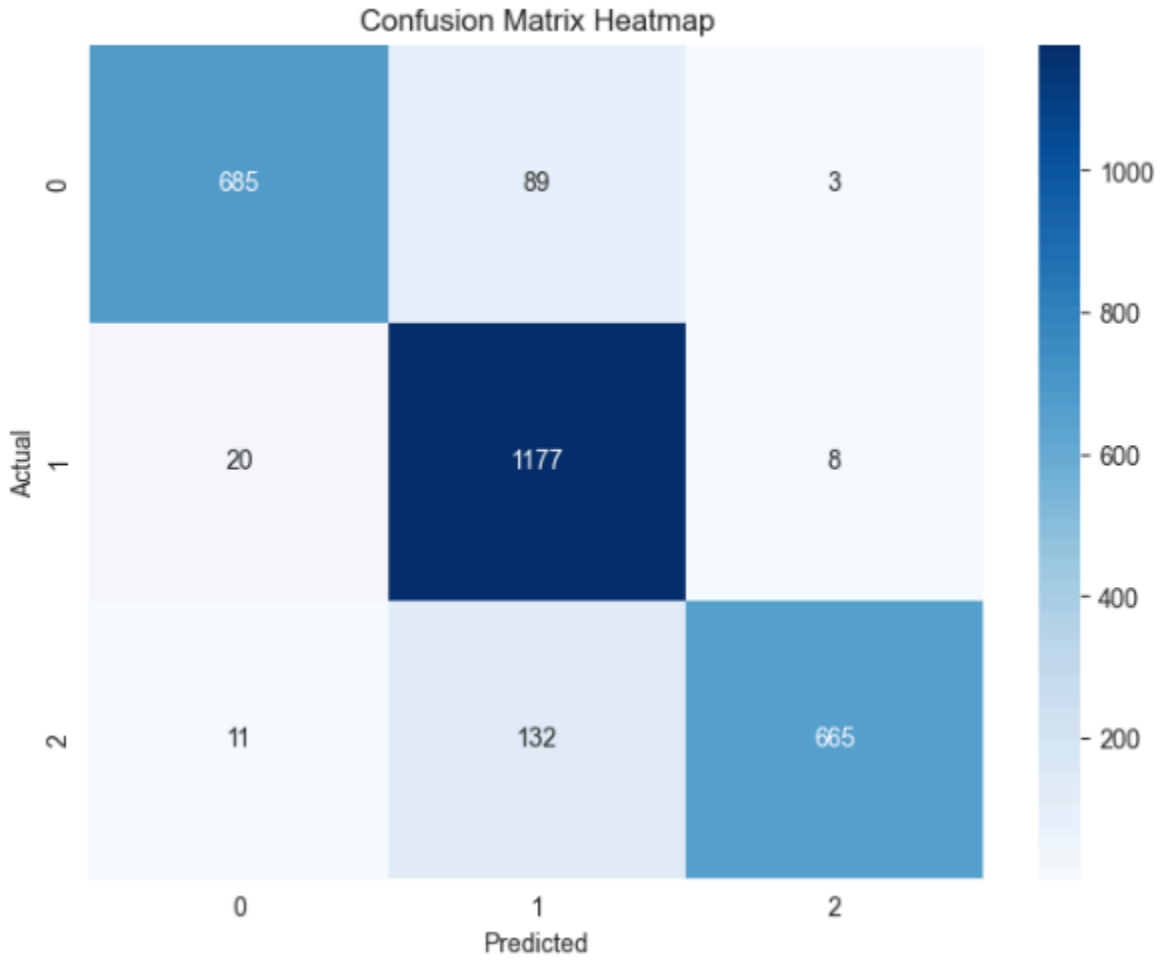
Appendix 3



Appendix 4


Token Count Distribution After Normalization
Number of Samples
2500
2000
1500
1000
500
0
0
100
200
300
400
500
Number of Tokens

Appendix 5


Confusion Matrix Heatmap
685
89
3
20
1177
8
11
132
665
0
1
2
Actual
0
1
2
Predicted
1000
800
600
400
200