Use of NLP Techniques to Censor YouTube Content Provoking Eating Disorder Affected Individuals

Ahreum Lee, Taewon Hwang, Seol Han, Chaeyeon Kang

Abstract

Eating disorders (EDs) are a growing public health concern, significantly impacting individuals' physical, psychological, and emotional well-being. Studies show a rise in ED prevalence, particularly among younger demographics, with media playing a critical role in exacerbating these disorders. The AKCSE Life Science (LS) and Computer Science (CS) divisions collaborated to develop a Chrome extension powered by machine learning (ML), specifically to automatically moderate YouTube content and help prevent ED-triggering videos. The CS division experimented with various binary classification models to optimize the extension's effectiveness, manipulating different preprocessing techniques and fine-tuning hyperparameters on YouTube video transcript data to select the most appropriate model. The resulting model was a stop word Naive Bayes model, likely resulting from the small dataset and key characteristics of video transcripts. The model's TF-IDF score allowed us to view the most "triggering" words detected by the model, which aligned with the words we sought during the initial selection process. From these results, there are several directions in which the project could expand, by either improving user experience on the extension, growing the dataset, or challenging the same task for various languages, starting with complete web scraping.

1 Introduction

Eating disorders (EDs) are an escalating public health issue, with prevalence rates rising significantly, especially among younger demographics. Social media and digital platforms play a substantial role in triggering or reinforcing these disorders by making potentially harmful content widely accessible. Among these platforms, YouTube stands out as a major content hub, where moderation largely relies on users flagging harmful material. This approach, however, often falls short, as it cannot always intercept triggering content in a timely or effective manner.

To address these limitations, the AKCSE Life Science and Computer Science divisions initiated a collaborative, multi-disciplinary project focused on developing a solution. This tool would automatically identify and censor ED-triggering content, offering a more proactive solution to reduce harmful exposure and promote safer media environments.

This project aimed to create an effective tool for filtering potentially triggering content and enhancing user protection. YouTube was chosen due to its widespread use across genders and generations, compared to various other social media platforms, like reddit or Instagram. Drawing from the Life Science (LS) division's insights on eating disorders and digital media's psychological impacts, the Computer Science (CS) team developed and optimized machine learning models, focusing on Naive Bayes and Linear Regression, and tested preprocessing methods like lemmatization, stemming, and stop word removal to maximize accuracy. The final Chrome extension integrates video and audio censoring, offering ML-driven moderation to reduce users' exposure to eating disorder triggers on YouTube.

2 Methodology

2.1 Dataset

Targeted specifically to our chosen platform, YouTube, our dataset comprises video titles and transcripts. To gather this data, we implemented a custom web scraping script using youtube-transcript-api which is designed to extract the titles and transcripts from YouTube video URLs. From this script, we were able to collect and annotate a dataset of 100 URLs, each consisting of 2 lines: one title and one transcript, resulting in 100 lines of data. In some cases, the Youtube webscraping API could not scrape a title and/or transcript, because of

an unknown cause or the content was in a language other than English. In this case, we made sure to at least scrape the title, to maximize our efforts in censoring.

2.2 Model

We tested a total of 16 different prediction models by combining 4 distinct preprocessing methods with two ML linear binary classifiers: Naive Bayes and Logistic Regression. Fine-tuning of hyperparameters was performed on both models to optimize their performance. We experimented with four different preprocessing variations to understand how each affected model performance:

- **No preprocessing:** In this variation, the raw text was used as-is, with no modifications. This baseline approach helped us assess the impact of advanced preprocessing techniques.
- **Lemmatization:** Lemmatization reduces words to their base or dictionary form, allowing words like "running" and "ran" to be treated as "run." This method preserves the meaning of the word while standardizing different forms of the same root word.
- **Word Stemming:** Stemming is a more aggressive approach compared to lemmatization. It trims words down to their root by removing affixes (e.g., "running" becomes "run" and "jumps" becomes "jump"). While stemming can result in less linguistically accurate transformations, it reduces the dimensionality of the data.
- **Stop Word Removal:** Stop words, such as "the," "is," and "in," are commonly occurring words that contribute little meaning to the text. Removing stop words reduces the noise in the data, allowing the model to focus on more meaningful terms.

As an aside, we used TF-IDF (Term Frequency-Inverse Document Frequency) scores to challenge our human selection of triggering content. The TF-IDF score quantifies how frequently a term appears in a document (TF) and how important that term is across the entire corpus (IDF), thus terms with higher TF-IDF scores are deemed more significant for classification purposes.

2.2.1 Model Training

For the machine-learning models, we trained and fine-tuned Gaussian Naive Bayes and Logistic Regression on the preprocessed data. These linear models were imported as modules from sklearn.

- **Naive Bayes:** We applied Gaussian Naive Bayes, a simple classifier based on Bayes' Theorem, and optimized the key alpha hyperparameter.
- **Logistic Regression:** Logistic Regression, a linear model used for binary classification, was trained and fine-tuned on hyperparameters such as the regularization strength (C) and iteration number to achieve the best possible balance between model accuracy and overfitting.

2.3 Chrome Extension

Our title and transcript scraping code was integrated into a Python application, which works in conjunction with JavaScript components to create a functional Chrome extension. To enhance the user experience and ensure effective moderation, we developed additional JavaScript-based censoring features, notably visual censoring, in the form of an overlay object, and audio censoring. When a video is detected as containing triggering content, these features are activated to protect the user from exposure.

3 Results Analysis

3.1 Model Accuracy

Among the preprocessing methods tested, stop word removal proved to be the most effective. It achieved a maximum accuracy of 0.9 with the Naive Bayes classifier (Figure 2). This was notably higher than the performance of other preprocessing methods, which did not achieve accuracy scores above 0.5. This demonstrates the critical importance of filtering out irrelevant or common words in text classification tasks, especially when dealing with sensitive content like ED triggers. Fine-tuning the models generally had minimal impact on performance. For the Naive Bayes models, fine-tuning did not affect accuracy across any of the preprocessing variations. However, for Logistic Regression, fine-tuning led to a small but noticeable improvement, increasing accuracy by 0.1 in some cases (Figure 3). This indicates that while hyperparameter optimization can be beneficial, it was not a

major factor in improving the model's effectiveness for this particular task. Finally, we exported the Naive Bayes model with the highest accuracy of 0.9. Naive Bayes is simple and efficient, which makes it particularly effective for small datasets. Its success here is likely due to these advantages, allowing it to perform better than other models in this context.

3.2 TF-IDF Scores

The top 10 words associated with triggering content included terms such as "fat," "jealous," "tiktoks," and "model" (Figure 1). These words have clear relevance to content that may potentially trigger eating disorders (EDs) and align with our expectations for identifying harmful content. However, other frequently occurring words identified through TF-IDF scoring were less relevant to the subject matter. Common stop words such as "what," "the," and "of" appeared prominently in the data. The prevalence of these irrelevant terms is likely why stop word removal had a significant impact on improving model accuracy. By eliminating these low-information terms, the model could focus more effectively on meaningful features in the dataset.

4 Conclusion

The experimentation led to a straightforward Naive Bayes model, achieving effective results. The TF-IDF analysis of trigger words aligned closely with initial criteria for identifying triggering content, reinforcing the model's relevance. This extension offers direct, practical moderation for content linked to eating disorder triggers.

4.1 Challenges and Applications

Expanding the dataset to 1,000 lines and incorporating non-English videos could significantly enhance model accuracy and expand its use to users outside of English-speaking countries. This would involve creating a multilingual web-scraping script and exploring NLTK's preprocessing functionalities for various languages.

5 Statement of contributions

The success of this project can be attributed to the collaborative efforts of all team members—Ahreum, Taewon, Seol and Chaeyeon. Led by team leader Ahreum, each member took part in the production of the project, notably Ahreum for ML experimentation and integration, Taewon for web development, and Seol and Chaeyeon for the research being the foundation of this project and dataset annotation.

6 Figures

Top 10 Words More Common in Trigger Content:

	word	score_diff
3966	tips	0.048567
1317	fat	0.043595
1892	jealous	0.040233
3953	tiktoks	0.040233
2903	person	0.035304
4251	what	0.034513
4166	victoria	0.033605
1429	forever	0.032633
2290	model	0.032377
4081	uk	0.030477

Top 10 Words More Common in Non-Trigger Content:

	word	score_diff
3904	the	-0.058286
2196	maze	-0.036200
1011	diaries	-0.032628
340	bazaar	-0.032628
1638	harper	-0.032628
1017	diet	-0.032242
481	breakfasts	-0.028579
2709	of	-0.027182
2342	music	-0.025847
367	beginners	-0.024596

Figure 1: TF-IDF Scores for Top 10 Relevant Words

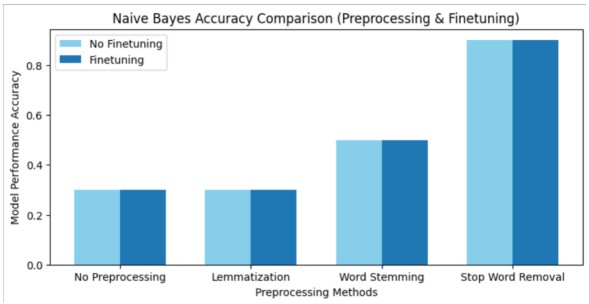

Figure 2: Naive Bayes Model Performance

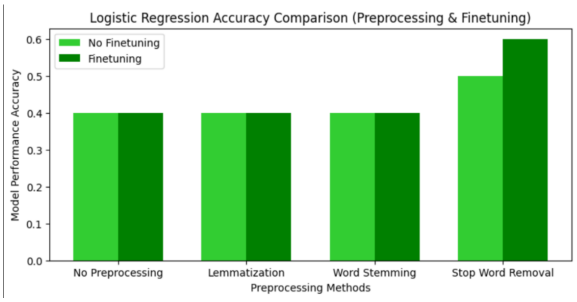

Figure 3: Logistic Regression Model Performance

References

1. **nltk Linear Classifier Modules** https://scikit-learn.org/stable/modules/linear_model.html
2. **Youtube Transcript Scraper API** https://pypi.org/project/youtube-transcript-api/