**Recognizing Internet Toxicity**

**IST 736 Final Project**

**Xiangzhen He | Katherine Hurtado-Da Silva | Jazmin Logroño**

# Introduction

Merriam-Webster defines toxic as "extremely harsh, malicious, or harmful." The definition provided is vague because harshness is all relative to a person. For example, a person initially from a tough neighborhood will likely have a higher tolerance for harshness than someone who grew up in Beverly Hills. Due to the vague definition and relative nature of toxicity, toxic speech and writings are difficult to categorize. Curse words introduce another complication. If someone says, "WTF," is that a harsh, malicious statement, or are they using a combination of words to express certain emotions? The internet has introduced new ways for people to express themselves, but also new ways to insult others. Toxic expressions have grown to have a largely negative impact on internet users; identifying toxic expressions can help mediate the exposure to toxic comments and their impacts on internet users.

Toxic comments normally fall into two categories, hate speech and online harassment. Both have been ongoing issues for years and research finds that around 41% of Americans experienced at least one form of online harassment in 2020[[1]](#footnote-2). Toxic comments have worked their way into all parts of the internet. Social media platforms, like Twitter, Instagram, and Facebook, are websites where people sometimes find themselves harassed or "trolled" by other users. The internet's anonymity gives people more confidence and less responsibility for anything written online. Anonymity provides opportunities for negative online behaviors with little to no recourse. This makes it so these people don't see any real negative consequences or punishments for their online behaviors. Social media is not the only problem. Toxic comments have also worked their way into information platforms such as Wikipedia. Although censorship is always a drastic solution, one question to consider before censorship is, do toxic comments belong on information platforms like Wikipedia?

A picture containing text, clipart

Description automatically generated

*Image 1, source: https://en.wikipedia.org/wiki/Toxicity*

Regardless of the ambiguity around free speech and negative internet behaviors, the effects of online toxicity are undeniable.

* Females and individuals from marginalized communities are disproportionately affected by online toxicity.
* 74% of adults that are involved in online gaming reported online toxicity, where 1 out of 10 players reported having depressive or suicidal thoughts in response to the harassment.
* Children are the most vulnerable to this threat. Out of the 95% of United States teenage internet users, 37% are cyberbullying victims, which is a form of online toxicity developed social anxiety.
* L1ght, an AI-based startup conducted an analysis of millions of websites, popular teen chat sites, and gaming platforms and concluded that there was a 900% increase in hate speech directed towards China or individuals with this ethnic background and a 40% rise in online toxicity among teens and children.

Toxic behaviors, like harassment are not tolerated in the physical world and these statistics are pressuring online platforms to accept accountability for the elimination of online toxicity.

With the growing concerns related to negative online behaviors continuing to challenge companies who run online platforms, Section 230 of the Communications Decency Act was enacted in 1996. This legislation provides immunity to online platforms from civil liability based on third party content and allows these online platforms to freely remove content in certain circumstances. Section 230 gives online platforms legal grounds to moderate their content, however at the same time does not force these online platforms to take any actions as they are immune to liability. Many companies may choose not to moderate their platforms which in the end may also lead to more online toxicity.

Businesses have a responsibility of protecting their customers from unreasonable risks during their patronage and online platforms are not the exception. They need to moderate their comment forums to ensure discourse quality, and a safe and civil environment for all users. The negative impact social media platforms have on users’ mental health is at the heart of big tech regulatory conversations and basis for the immense scrutiny these companies currently face. Apart from user disengagement, it is the probable amendment to Section 230 that should be the greatest business concern for online platforms. It is in every online business’ best interest to be proactive through the moderation of their platform’s online toxicity to avoid potential future lawsuits, and our group will assist with these efforts.

# Analysis

## Goal

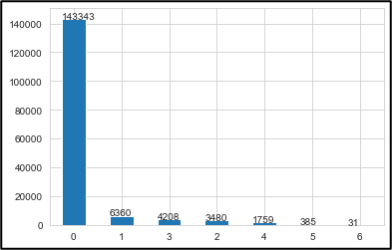
Machine learning algorithms, like Latent Dirichlet Allocation (LDA), K-Means Clustering, Multinomial Naïve Bayes, and Support Vector Machines (SVM) can help identify toxic comments posted online. Additionally, algorithms can help in identifying patterns to detect toxic behavior. The analysis aims to answer the following questions: How well does each algorithm learn patterns to identify differences between toxic and non-toxic comments? What types of toxicity exist in toxic comments? Which words contribute to toxic comments? Are there different levels of toxicity?

## About the Data

The Kaggle toxic comment training data set provided by the Conversation AI team, a research initiative founded by Jigsaw and Google, was used in this report. The original data set is comprised of 159,571 user comments from Wikipedia’s talk page edits. Each comment was labeled by human raters for toxic behavior including toxic, severe toxic, obscene, threat, insult, and identity hate, and more than one label can be assigned to each comment.

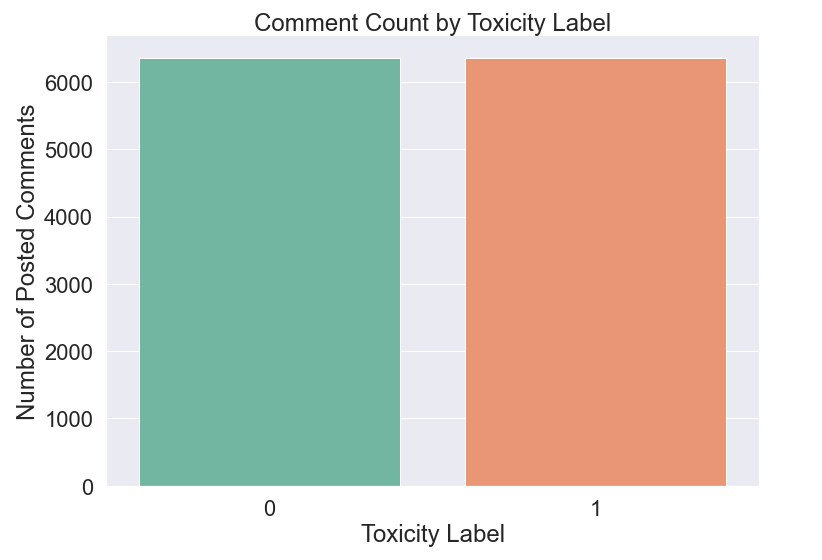
The data was provided in a comma-separated file or csv file where each comment is represented as a row. Any comments that have less or equal to 10 characters were first removed from the data.

When reviewing the total number of labels assigned to each comment in the remaining data set, 143,343 of user comments were not assigned any toxic label. Therefore, most comments were considered non-toxic by human raters.



*Figure 1: Distribution of Toxic Labels*

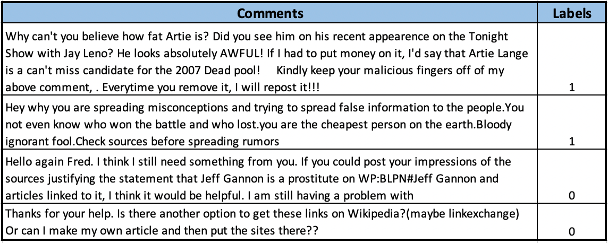
Uneven distributions of labeled data adversely affect accuracy levels in classification models. Due to the importance of avoiding this, the data was carefully selected to ensure there were no major discrepancies in the number of labels. A subset of data that only includes 6,360 randomly selected non-toxic comments and 6,360 comments with only one toxic label assigned were created to perform clustering and classification analyses. The final data set includes a comment column and a label column. The value of a label can be either “0” or “1” indicating non-toxic or toxic comments. Figure 2 illustrates the number of comments by toxicity label, confirming there is no imbalance.



*Figure 2: Distribution of Sampled Comments by Toxic Labels*

## Data Cleaning

Data cleaning was performed by removing URLs and special characters from user comments. Table 1 below presents four sample user comments with their corresponding labels being attached.



*Table 1: Example of User Comments Data Records*

The comments were written in plain English without any emojis. However, each comment consists of lots of punctuation that would need to be removed during the vectorization process.

## Data Exploration

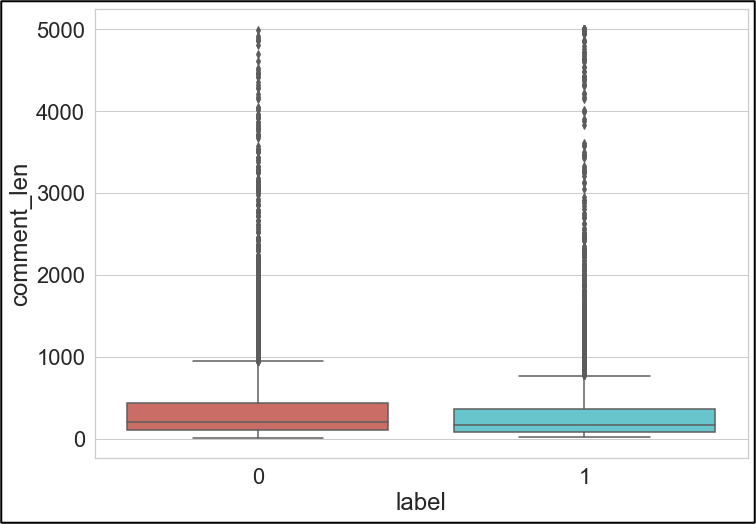
A word cloud was created to show which words were most frequently used in these comments.



*Figure 3: Word Cloud for User Comments*

The top words used were “article”, “page”, “Wikipedia”, “people”, “please”, “talk”, “thank”, “will” and “know”. Words such as “good”, “well” and “right” have positive sentiments. The words that could be used in rude and disrespectful comments are “Bark”, “hate”, “die”, “PIG”, etc. The word cloud also shows that there is not a specific topic for these comments instead of discussing the Wikipedia page and article.

Figure 4 below displays the distribution of the numbers of characters in each type of comment.



*Figure 4: Box Plot of Comment Length in Each Category*

## Text Preprocessing and Data Preparation

The comment text data and labels were saved to separate variables to be used for further analyses.

The text data was transformed to document-term matrices using CountVectorizer and TfidfVectorizer imported from python scikit-learn library. Both vectorizers first tokenized the comment texts by splitting them into individual words or tokens. During this process, punctuation was separated and removed, and each token or word was converted to lower case when using the default parameter. The unique words or tokens obtained after the tokenization process formed the vocabulary for each vectorizer.

CountVectorizer and TfidfVectorizer then converted the comments to vectors using the vocabularies obtained previously. CountVectorizer could return the frequency of each word or token that occurs in the comment text document. TfidfVectorizer uses the term frequency-inverse document frequency weighting to calculate the normalized frequency of each word or token. When there were multiple documents or multiple comments, the results of CountVectorizer and TfidfVectorizer were in the format of document-term matrices where each comment was represented as a row and the unique words or tokens in the vocabulary were represented as columns.

Several different vectorizers were created to transform comment text data into different document-term matrices. Generally, three types of vectorizers were used in this report.

* 1. Unigram term frequency vectorizer created using CountVectorizer. Stop words were removed using the built-in stop word list for English.
  2. N-gram term frequency vectorizer created using CountVectorizer. Stop words were removed using the built-in stop word list for English.
  3. Tf-idf vectorizer created using TfidfVectorizer. Stop words were removed using the built-in stop word list for English.

In order to reduce the vocabulary size, each vectorizer only keeps tokens or words that have a document frequency greater than or equal to 5. Additional vectorizers that include lemmatization process using the WordNetLemmatizer from the nltk package were created as well to compare performance of different tokenization and vectorization processes.

Additionally, producing word clouds and reviewing the features helped provide insight on unhelpful features; over two thousand unhelpful features, including numbers, years, and names were removed for the classification Analysis. Words referencing cultures, ethnicities, religions, countries, locations, and forms of government were removed to avoid introducing bias in the models. The goal was to prevent the models from labeling comments as toxic based on words referencing cultures, ethnicities, and religions. Ideally, a subject matter expert should review all features produced by the vectorization process to ensure all words referencing cultures, ethnicities and religions are removed from the Wikipedia comments corpus. Figure 5 is a word cloud produced after removing the unhelpful features. Words like “lover,” “communication,” “improvement,” and “associate” might help in differentiating non-toxic comments from toxic ones. Words like “donkey,” “dislike,” “coward,” and “killer” will be useful in detecting toxic comments. Curse words will also prove helpful in identifying toxicity.



*Figure 5: Word Cloud for User Comments after removing almost two thousand features*

In the next section, multiple Machine Learning Algorithms will be used on document-term matrices obtained from different vectorizers to analyze the user comments data.

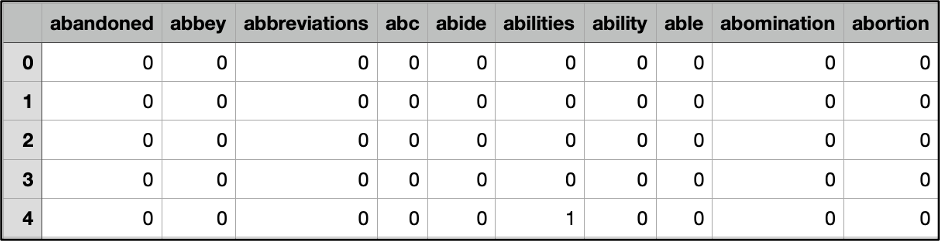
## Models and Analyses

1. Topic Modeling

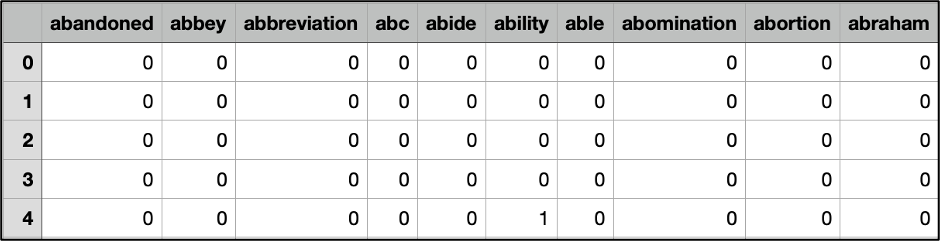
Topic modeling is an unsupervised machine learning technique that is frequently used to discover hidden semantic structures in a collection of text documents. Latent Dirichlet Allocation (LDA) was used in this report to perform topic modeling analysis. LDA is a generative model and is one of the most popular topic modeling methods that builds a model based on the probability distribution of topics and probability of words over the vocabulary.

Two document-term matrices created by CountVectorizer with and without lemmatization were used to perform the topic modeling analysis on the original data set without feature reduction. CountVectorizer was chosen instead of TfidfVectorizer because LDA is based on term count and document count[[2]](#footnote-3). In addition, lemmatization was used instead of stemming since stem words are more difficult to interpret.

Table 2 and 3 below present the first 5 rows and 10 columns of the document-term matrices created by two vectorizers mentioned above. The values in these two matrices can be any integers that are greater or equal to 0.



*Table 2: First 5 rows and 10 columns of the document-term matrix created using unigram term frequencies.*



*Table 3: First 5 rows and 10 columns of the document-term matrix created using unigram term frequencies with lemmatization.*

The document-term matrices are all sparse matrices in which most of the elements are zero. The first matrix contains 7,123 columns and the second matrix is comprised of 6,489 columns. Lemmatization process reduced the size of the vocabulary by combining words having the same base form. For example, “abilities” and “ability” were represented by separate columns in Table 2 but were combined as one column “ability” in Table 3.

After obtaining the document-term matrices from text data, LatentDirichletAllocation imported from sklearn.decomposition module was used to build the LDA model. The most important parameter of LatentDirichletAllocation is the number of components, which controls how many topics the model will be fit into. Therefore, GridSearchCV from scikit-learn library was used to search for the topic number that could produce the best results based on model’s log-likelihood score and perplexity. A higher log-likelihood and lower perplexity indicate better model performance.

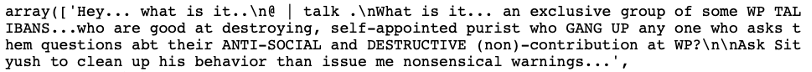
Additionally, the “online” learning\_method was selected and GridSearchCV was also used to search for the learning\_decay value that could generate the highest model results. The “online” learning method is good for large data sets and the learning decay parameter controls the learning rate of the model.

LDA results using different document-term matrices and model parameters will be compared in the results section. In the end, user comments will be reviewed by topics and topics will be reviewed by most highly weighed and most relevant words identified in each topic.

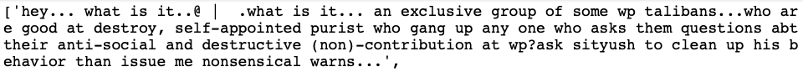
1. Naïve Bayes

Naïve Bayes is a supervised learning algorithm that is compatibility with discrete features and its scalability. This probabilistic learning method is also capable of handling large datasets. Naïve Bayes was explored due to its suitability with the nature of the dataset and for its ability to be scaled to large volumes of posted comments.

While inspecting the sampled comments, it was apparent that there were a significant number of cases where words had “\n” and “\n\n” as prefixes. This would not be discernable to tokenizers and would interfere with the stopword removal and overall quality of tokens. Therefore, a loop was used to replace these incorrect prefixes with blank spaces. The features “Wikipedia”, “wiki”, “article”, “like”, “talk”, “page”, “know”, “did”, “want”, “use”, “does”, “don’t”, “don”, “beca”, “acc”, “say”, “way”, “really”, “bishonen”, “http”, “going”, “think”, “edit”, “need”, and “ing” were also removed through a loop after seeing its lack of relevancy to the context of toxicity as top features for different vectorizations. These words and prefixes were referred to as Basic Features, where a sample of the comments before and after their removal can be seen in the Figures below.



*Figure 6: Sample Comment Prior to Basic Features Removal*

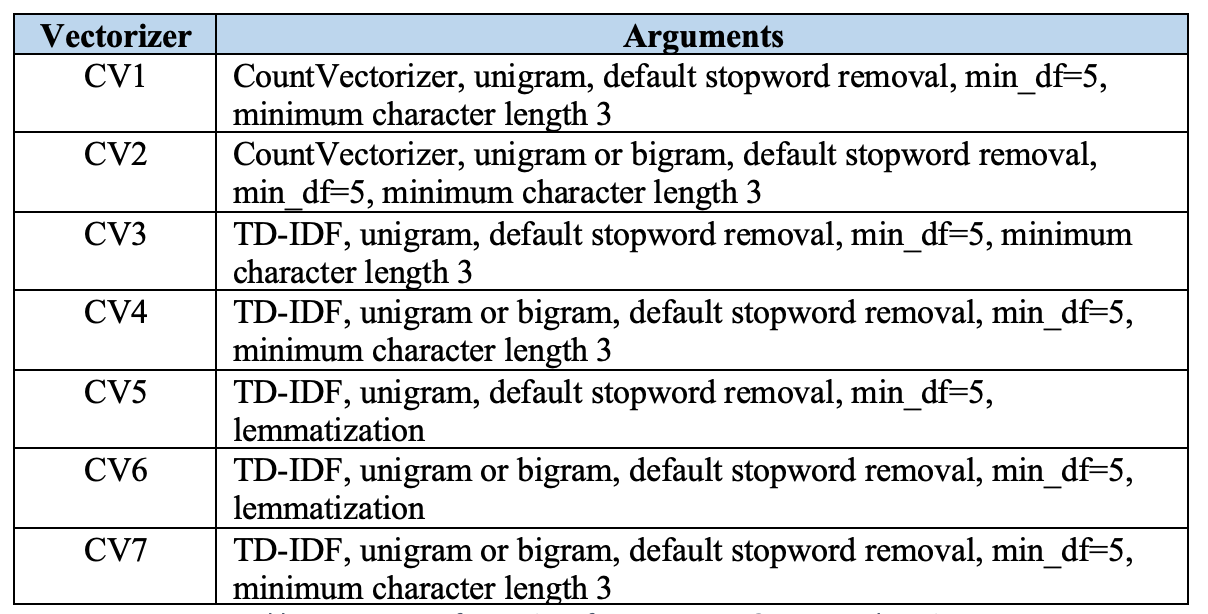


*Figure 7: Comments After Basic Features Removal*

Feature bias was an important focus in the analysis. Allowing social group identifiers to be trained for the identification of toxic content was explored in efforts to ensure discriminatory biases wouldn’t be propagated in the models. Considering bigoted content composes most online toxicity, the identification of features that should be removed to help reduce social biases. without removing the essential context of the dataset, was challenging. Ultimately, a list of features relating to countries, ethnicities, races, sexual orientations, and derogatory terms within these categories were curated and referred to as Social Group Features. The models were run with and without Social Group Features in efforts to compare the quality of important features and training-to-test accuracy trends.

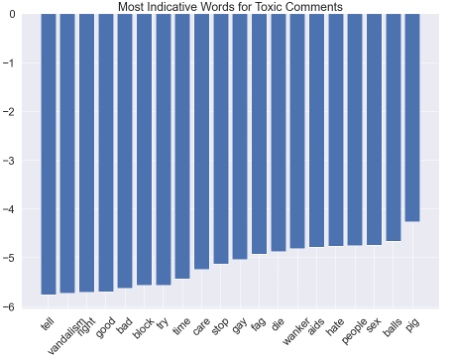
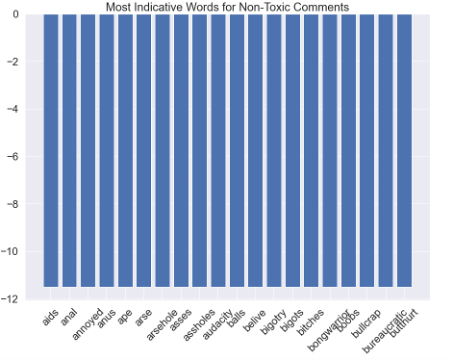
To minimize overfitting, the dataset was split prior to vectorization using the train\_test\_split function, where 70% of the data was reserved for training the data and the remaining 30% was used as unseen data to test the trained model on. The stratify parameter was included to ensure that the proportion of toxic and non-toxic reviews were preserved in the sampled train and test datasets. Cross validation is the practice of training a model on K-1 folds and validating it on the unused fold of the training dataset, where K is the number of groups the analyst decides to divide the training dataset into and the number of times the process is repeated. This method was applied with K=5, where the average accuracy score of all five iterations were estimated to the nearest tenth, to identify the true fidelity of each trained model and to determine its respective predictive performance on the test data. The model’s predictive accuracy on the unseen test data was ultimately referenced to determine the most optimal model for the identification of toxic comments.

As mentioned previously, CountVectorizer and Tf-idf were used to reduce the dimensionality of the comments in the training set. With the goal of preserving the most important features to predict a comment’s toxicity label, seven variations of the vectorizers were used and can be referenced in Table 4.

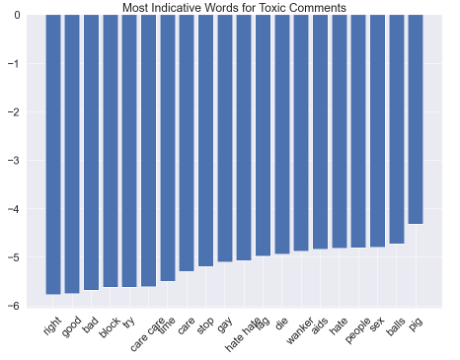
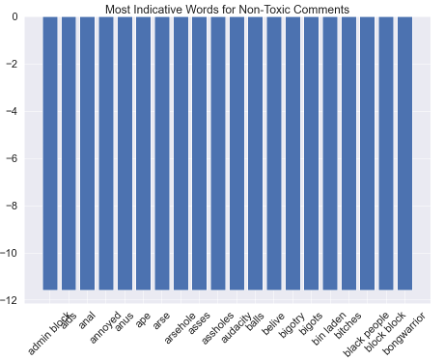


*Table 4: Summary of Vectorizers for Naïve Bayes & KMeans Clustering*

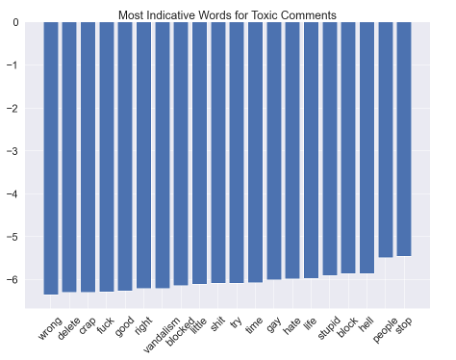
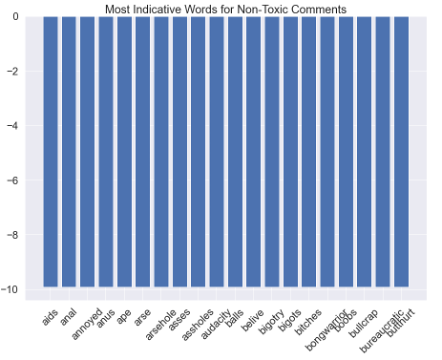
For each vectorizer, the logarithmic probabilities were used to generate the top 20 features used to differentiate non-toxic from toxic comments. The figures below illustrate these features for all seven vectorizers when the dataset included the Social Group Features. The features “anal”, “annoyed”, “anus”, “arse”, “arsehole”, “audacity”, “belive”, “bigotry” and “bongwarrior” were evident in all seven Figures, indicating these were consistent indicators for non-toxic comments. The words “right”, “block”, “try”, “time”, “stop”, “gay”, “hate” and “people” were consistent across all seven vectorizers as features used to predict toxic comments. The next strongest trends for the identification of toxic comments included the word “vandalism”, which appeared as a strong indicator for six out of seven vectorizers, while the words “shit”, “fuck”, “stupid”, “hell” and “life” appeared in five out of the seven vectorizers. For the identification of non-toxic comments, there were no features that were common in six out of seven vectorizers, but the words “aids”, “ape”, “asses”, “asshole”, “balls”, “bigots” and “bitches” did appear as strong indicators for five of the vectorizers. Though it was surprising to see the range of insulting words that consistently appeared to be indicators of non-toxic language, they did appear to be weaker in nature compared to the range of features for toxic comments. It is also worth noting that “black people” appeared as an indicator of non-toxic comments for ngram vectorizers, showing it was not propagated negatively as analysts might assume. However, homophobic verbiage was identified as indicators of toxic comments, indicating the topic in the toxic-labeled comments may be directed at this demographic.



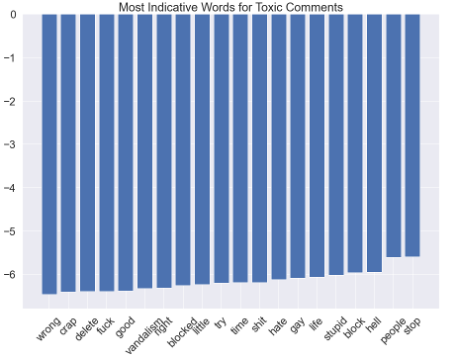
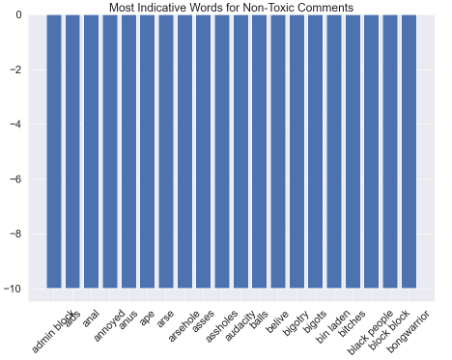
*Figure 8: Most Indicative Words for Naïve Bayes Resulting from CV1*



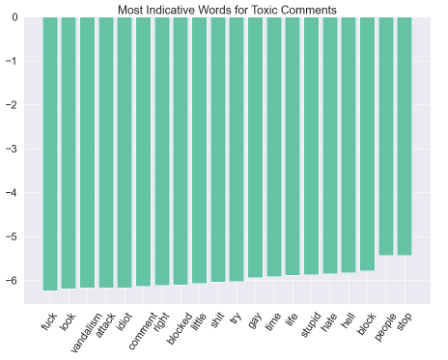
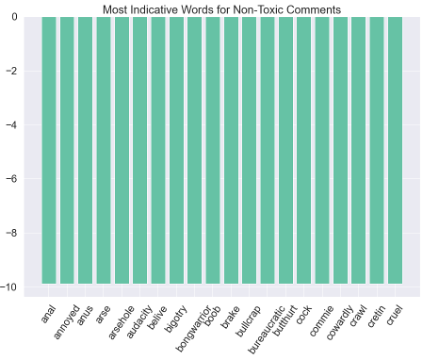
*Figure 9: Most Indicative Words for Naïve Bayes Resulting from CV2*



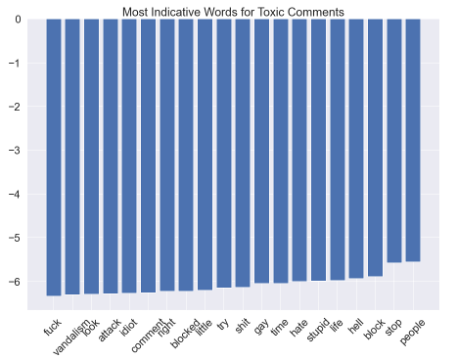
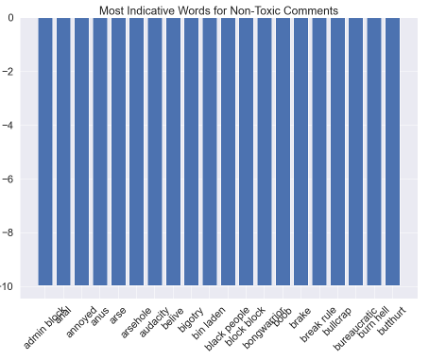
*Figure 10: Most Indicative Words for Naïve Bayes Resulting from CV3*



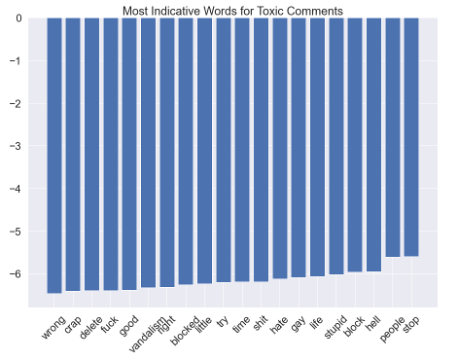
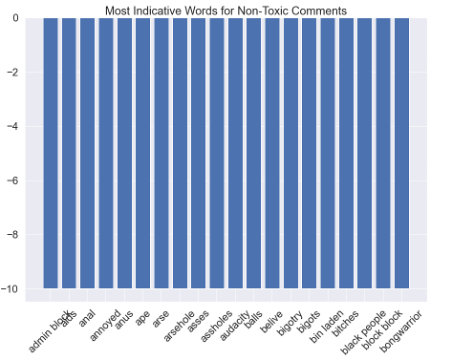
*Figure 11: Most Indicative Words for Naïve Bayes Resulting from CV4*



*Figure 12: Most Indicative Words for Naïve Bayes Resulting from CV5*



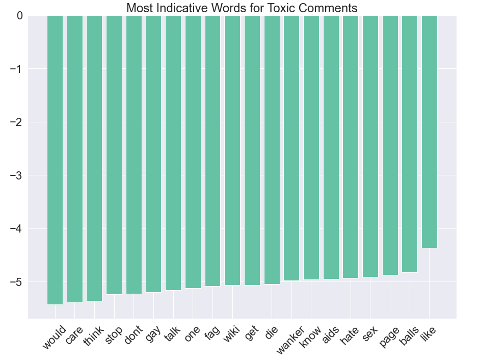
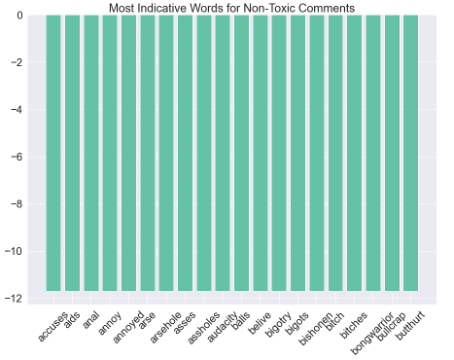
*Figure 13: Most Indicative Words for Naïve Bayes Resulting from CV6*



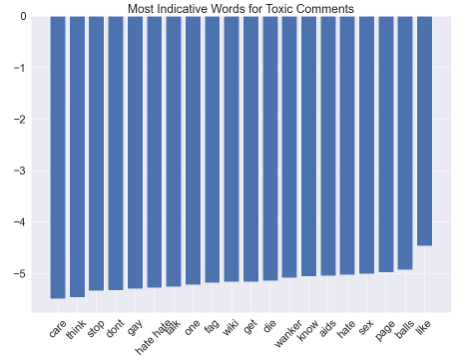
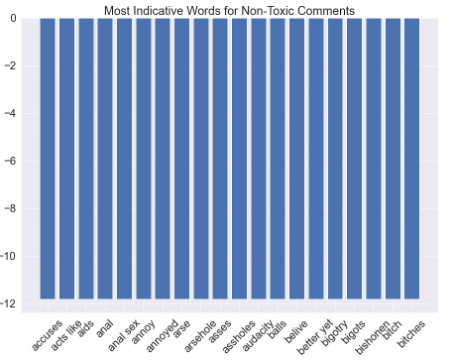
*Figure 14: Most Indicative Words for Naïve Bayes Resulting from CV7*

The figures below illustrate the top features for all seven vectorizers when the Social Group Features were removed from the dataset. Right away, the features used to identify non-toxic comments stand out for the 6th vectorizer. There are no similarities between CV6’s features and any of the features in other figures- it is evident that extensive removal of features has adversely impacted its ability to vectorize the n-grams. Setting CV6 aside, the words “accuses,” “anal,” “annoy,” “annoyed,” “arse,” “arsehole,” “asses,” “audacity,” “belive,” “bigotry,” “bigots,” “bishonen” and “bitch” were strong indicators of non-toxic comments for the remaining six vectorizers. The words “aids”, “asshole”, “balls” and “bitches” also appeared as strong indicators for non-toxic comments in five of the seven vectorizers. Although the word “accuses” was a new top feature for the identification of non-toxic comments, the variations of the same features in these findings and overall similarities to the features previously discussed, indicate that the removal the Social Group Features did not significantly improve the quality of the features.

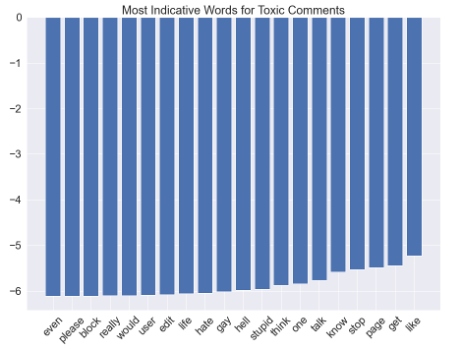
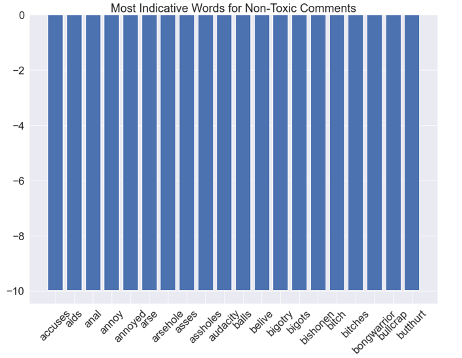
For the identification of toxic comments, the words “stop,” “think,” “gay,” “talk,” “one,” “get,” “know,” “page” and “like” appeared in all seven vectorizers. The next strongest trend for the identification of toxic comments included the words “hate,” “block,” “user,” “edit,” “life,” “stupid” and “hell,” which appeared in five out of the seven vectorizers. Compared to when the Social Group Features were not removed, the indicators “vandalism”, “shit”, “block” and “fuck” were lost. The findings also shows that a bias may have been inadvertently created by removing features that eliminated the bigram “black people” but leaving features that allowed “gay” as a top feature for toxic comments. When comparing the overall features of toxic comments, the relevancy of the features to toxicity appears to be better maintained without the removal of Social Group Features.



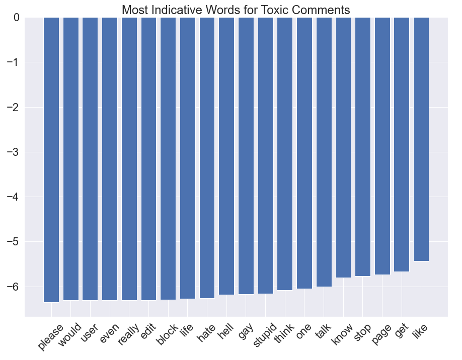
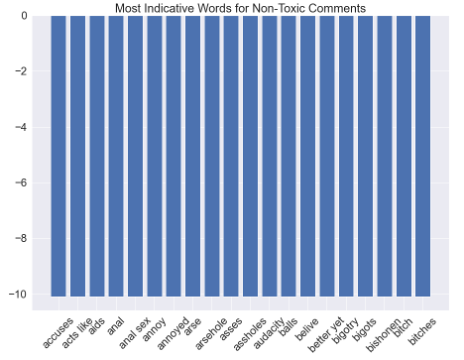
*Figure 15: Most Indicative Words for Naïve Bayes Resulting from CV1 & Social Group Feature Removal*



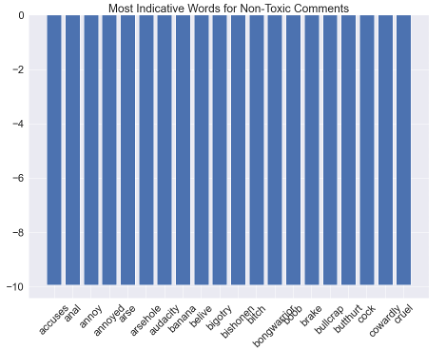
*Figure 16: Most Indicative Words for Naïve Bayes Resulting from CV2 & Social Group Feature Removal*



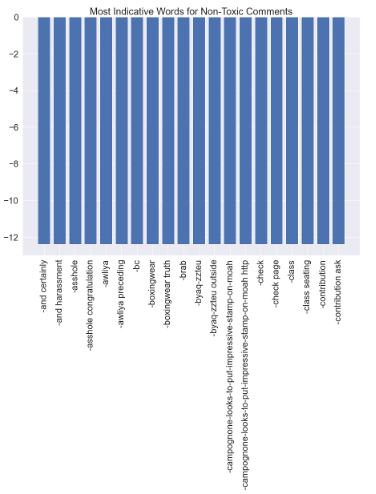
*Figure 17: Most Indicative Words for Naïve Bayes Resulting from CV3 & Social Group Feature Removal*

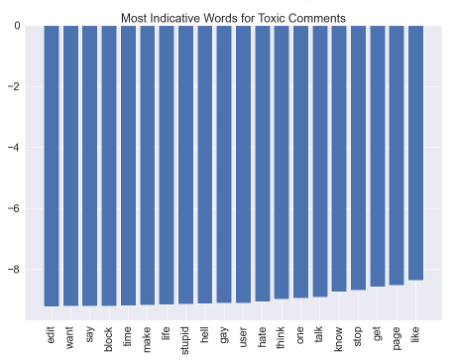


*Figure 18: Most Indicative Words for Naïve Bayes Resulting from CV4 & Social Group Feature Removal*

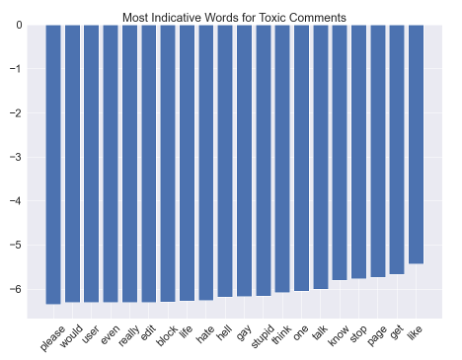
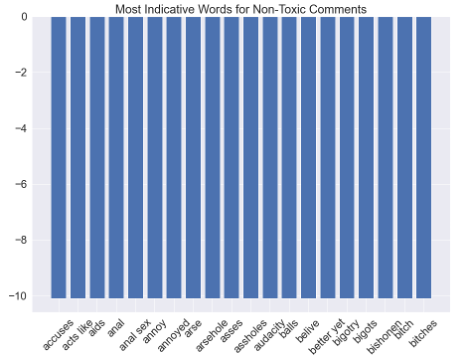


*Figure 19: Most Indicative Words for Naïve Bayes Resulting from CV5 & Social Group Feature Removal*





*Figure 20: Most Indicative Words for Naïve Bayes Resulting from CV6 & Social Group Feature Removal*



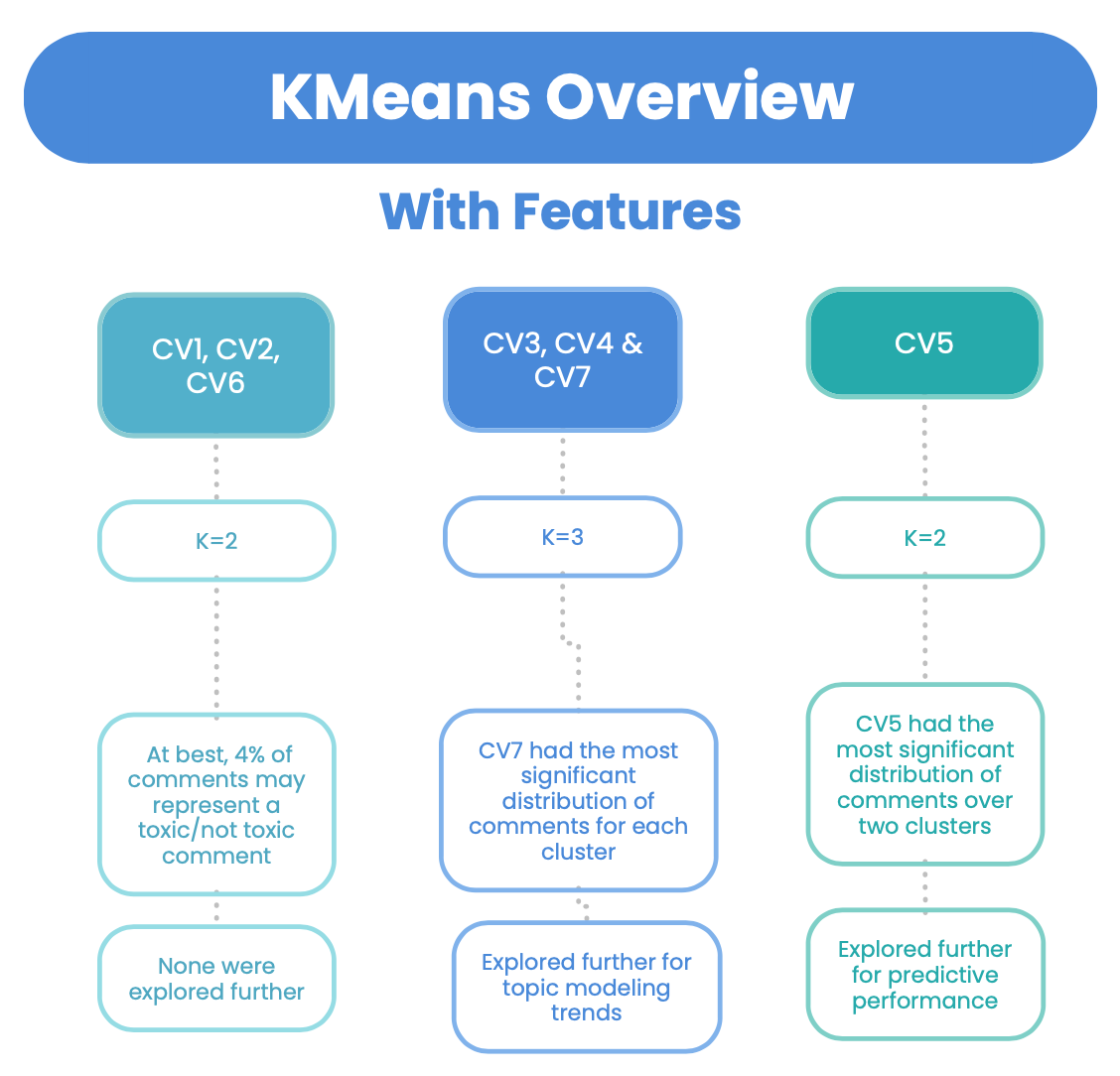
*Figure 21: Most Indicative Words for Naïve Bayes Resulting from CV7 & Social Group Feature Removal*

1. KMeans Clustering

KMeans clustering is an unsupervised learning algorithm, meaning it uses the similarities of the feature frequencies to group data together. KMeans was selected due to its adaptable nature to new data, scalability with large datasets, and because it does not use the gold standard labels to determine which cluster data belongs to. This model’s non-reliance on labeled data is the biggest advantage, where the potential lack of resources preventing the labeling of newly curated comments would not interfere with its application. Furthermore, KMeans could be leveraged to uncover subsets of toxicity within comments or to predict which cluster new comments belong to, making its duality the most promising aspect of this model.

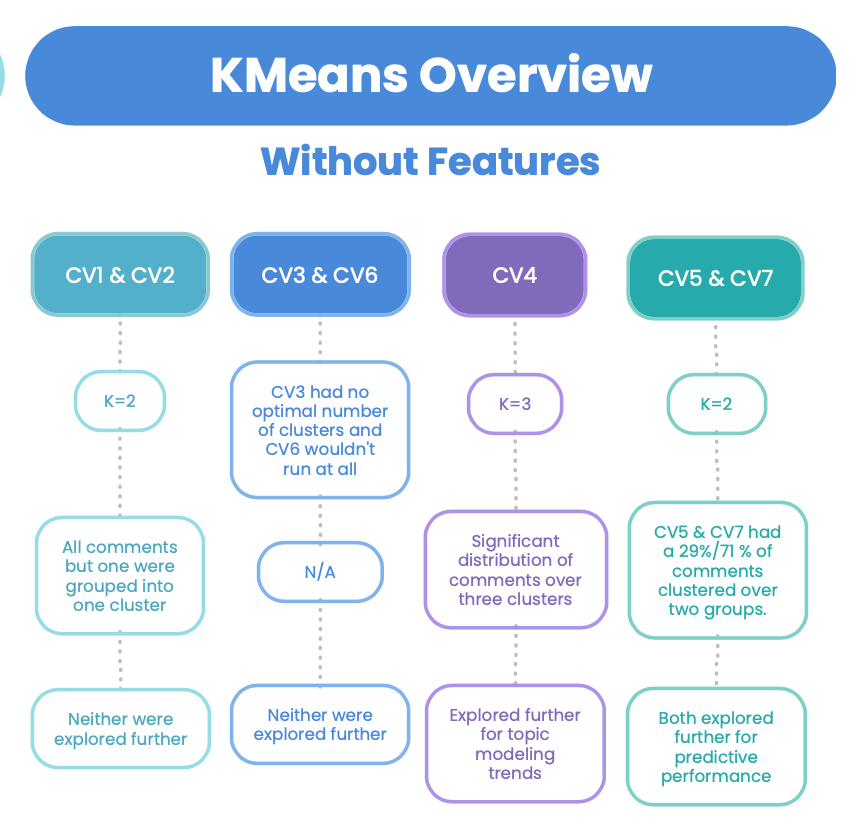
The same methodology used in Naïve Bayes was applied to KMeans Clustering. The training and test sets, vectorizers, removal of Basic Features, and additional removal of Social Group Features for a secondary analysis, all align with Naïve Bayes. A feature unique to KMeans, was the need to determine the number clusters the data would be fitted to. Yellowbrick’s KElbowVisualizer function was used to fit the seven types of vectorized data to a range of values for the cluster sizes. The point of inflection, that resembles an elbow, was used to determine the optimal value for a given dataset. Since the data is labeled, the expectation was to see these graphs report two as the optimal number of clusters, representative of not toxic and toxic comments. Figure 22 and Figure 23 summarize the KElbowVisualizer results for each vectorizer for the dataset with and without the Social Group Features.

Despite CV1, CV2, and CV6 reporting two optimal clusters for KMeans, they were not explored further due to the comments essentially being grouped as one cluster. As indicated in Figure 1, the data is balanced and contradicts what KMeans generated under these vectorizers. CV3, CV4, and CV7 all reported three optimal clusters for their respective features, which did not coincide with two types of labels in the dataset. However, it was assumed that these three groups may have been attributed to KMeans identifying subgroups of toxicity, like toxic, severe toxic, and not toxic, and were subsequently explored from a topic modeling perspective. Lastly, CV5 had the most promising distribution of comments across the two optimal clusters that KElbowVisualizer identified and was therefore explored further to see if it could be used as a predictive tool for toxic comments.



*Figure 22: KMeans Optimal Number of Clusters for Comments with Social Group Features*

When removing the Social Group Features from the dataset, CV1 and CV2 reported two optimal clusters for KMeans, but were not explored further due to the poor distribution of comments over both clusters as well. CV3 and CV6 were also not explored further due KElbowVisualizer’s inability to identify an optimal number of clusters for the features resulting from CV3 and for not being able to support this function at all when attempting the application of CV6. CV4 reported three optimal clusters for its respective features and was explored further from a topic modeling perspective. Features resulting from CV5 and CV7 demonstrated a promising distribution of comments across the two optimal clusters that KElbowVisualizer identified, providing the basis for its potential use as a predictive tool for toxic comments as well.



*Figure 23: KMeans Optimal Number of Clusters for Comments with Social Group Features Removed*

1. Support Vector Machines

Support Vector Machines, or SVM, are often used in text classification analysis. SVM works by finding the best line, plane, or hyperplane that separates toxic comments from non-toxic comments. Four SVM models using different kernels, Linear, Polynomial (Poly), Radial Basis Function (RBF), and Sigmoid were trained on the Wikipedia comments corpus. The four models were trained using 70% of data while 30% was used to test the models. A training and testing set was created using sklearn’s train\_test\_split. Stratify sampling was used to ensure balanced training and testing sets. Reviewing the twenty most indicative words for toxic and non-toxic comments for each model reveals Support Vector Machines can categorize toxic and non-toxic comments.

The most indicative words learned by the SVM Linear model are shown in Figures 24 and 25. Figure 24 shows the most indicative words for comments labeled as not toxic. The SVM Linear model learned words like “appealing,” “abilities,” “representing,” “bond,” and “potentially” for non-toxic comments. “Dig” was also learned by the model. When reviewing the Wikipedia comments corpus, the word “dig” was used in eleven comments; ten of the comments are labeled as non-toxic. The word is usually used to describe researching. For example, “dig up some information.” Figure 25 shows the most indicative words for comments labeled as toxic by the SVM Linear model. The SVM Linear model identified several curse words along with words such as “dominated,” “censoring,” “bullies,” “misery” and “abused” as key contributors to toxic comments. The word “boring” appears twenty-one times in the Wikipedia comments corpus. Eighteen of the times it is used in aggressive and relatively insulting comments; therefore, it is an indicative word for toxic comments.

Chart, bar chart

Description automatically generated

*Figure 24: Most Indicative words for Non-Toxic Comments produced by the SVM Linear Model*

Chart, bar chart

Description automatically generated

*Figure 25: Most Indicative words for Toxic Comments produced by the SVM Linear Model*

The most indicative features learned by the SVM Radial Basis Function (RBF) kernel model are shown on Figures 26 and 27. Words like “acceptance,” “accepting,” “accordance,” “accommodate” and “accomplish” are indicative for non-toxic comments (Figure 26). The model failed to identity words like “abused” and “abuses” as indicators for toxicity although they mostly appear in toxic comments. However, the model labeled “punish” as a toxic feature (Figure 27); reviewing the Wikipedia comments corpus reveals the word “punish” is used nine times in different comments. All comments are labeled as toxic.

Chart, bar chart

Description automatically generated

*Figure 26: Most Indicative words for Non-Toxic Comments produced by the SVM Radial Basis Function Model*

**Chart, bar chart

Description automatically generated**

*Figure 27: Most Indicative words for Toxic Comments produced by the SVM Radial Basis Function Model*

The SVM Sigmoid Function model produced a similar list of non-toxic words as the SVM RBF model as shown in Figure 27. The words “abide” and “absolutely” are indicative words in the SVM Sigmoid Function model, but not the SVM RBF model. The SVM Sigmoid Function model produced a completely different list of toxic words than SVM Linear and RBF models. The SVM Sigmoid model uses the word “disturbed” as an indicative word for toxic comments. Reviewing the Wikipedia comments corpus reveals the word is used eight times: four times in comments labeled as toxic, and four times in comments labeled as not toxic. Therefore, the feature “disturbed” is not very indicative of toxic or not toxic comments.

Chart, bar chart

Description automatically generated

*Figure 28: Most Indicative words for Non-Toxic Comments produced by the SVM Sigmoid Function Model*

Chart, bar chart

Description automatically generated

*Figure 29: Most Indicative words for Toxic Comments produced by the SVM Sigmoid Function Model*

The last Support Vector Machine model used a Polynomial kernel during training, and it produced the same list of non-toxic words produced by the model using an RBF kernel (Figure 30). The list of toxic features shown on Figure 31 demonstrate the model’s inability to identify toxic words. The word “yay” appears six times in the Wikipedia comments corpus, and it is only used twice in toxic comments.

Chart, bar chart

Description automatically generated

*Figure 30: Most Indicative words for Non-Toxic Comments produced by the SVM Polynomial Model*

*Chart, bar chart

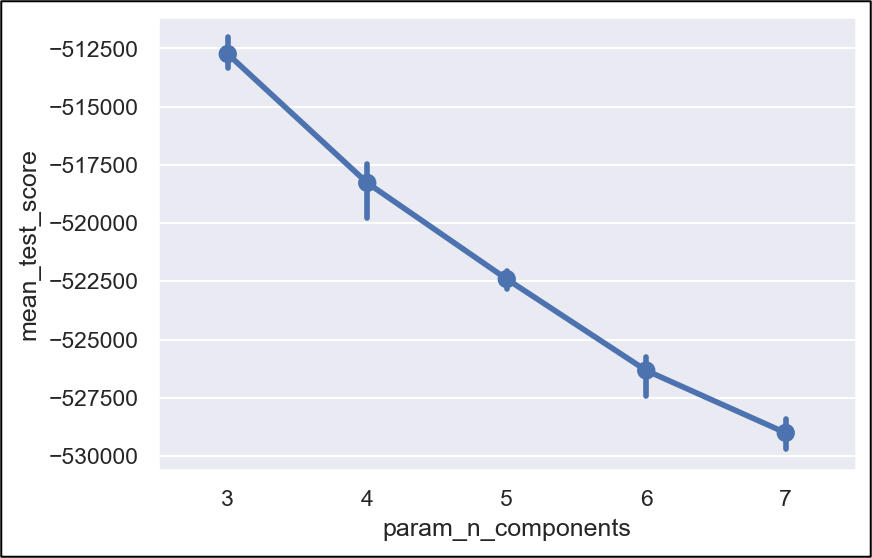
Description automatically generated*

*Figure 31: Most Indicative words for Toxic Comments produced by the SVM Polynomial Model*

# Results

1. Topic Modeling using LDA

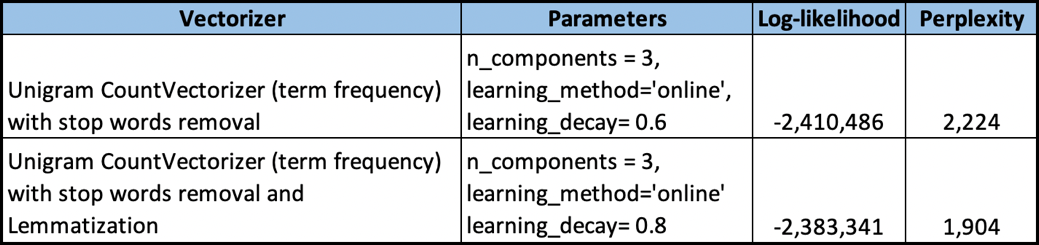
The topic number was determined by comparing the average log-likelihood values of LDA models when performing 5-fold cross validations on the entire data set. Figure 32 below presents the model performance when using different topic numbers.



*Figure 32: Model performance when having different number of topics*

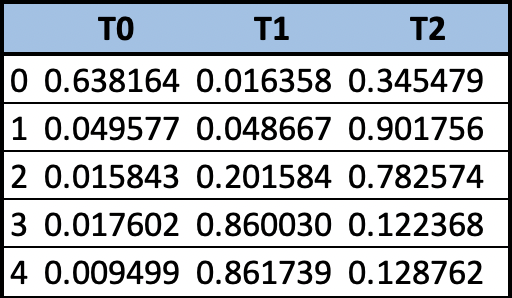
The best model performance with the highest average result was obtained when the topic number equals to 3. Therefore, LDA models were built to fit data into 3 topics.

Table5 below summarizes the parameters and performance of the LDA models built using two document-term matrices created from different vectorizers. The second model had better performance with higher log-likelihood and lower perplexity values.



*Table 5: Summary of LDA Model Results*

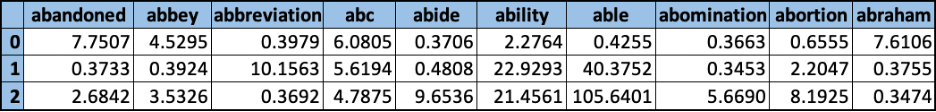
The Topic-Document Matrix created from the better performing model consisted of 12,720 rows and 3 columns. Each row contains the probabilities of a user comment belonging to each of the three topics created by the model.



*Table 6: First 5 Rows of the Topic-Document Matrix Generated by the Best LDA Model*

Table 6 shows the first five rows in the Topic-Document Matrix. Based on the probability values, the first comment is most highly connected to topic 0, the second and third comments are most highly connected to topic 2, and the fourth and fifth comments are mostly highly connected to topic 1.

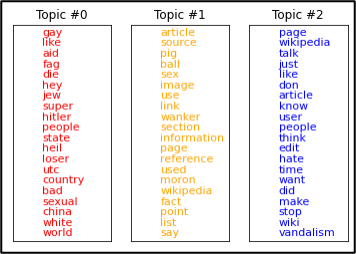
A Word-Topic Matrix was then created using the components\_ attribute of LatentDirichletAllocation to show the importance of each word in the vocabulary in each of the three topics.



*Table 7: First 10 Columns in Word-Topic Matrix Created from LDA Model Results*

As shown in Table 7, different words weight differently in each topic, the higher the number means the higher the level of importance. “abandoned” and “abraham” have the highest weight in topic 0, “abbreviation” is important in topic 1, “abide”, “able”, “abomination” and “abortion” have very high weights in topic 2.

Using the values in the Word-Topic Matrix, the top 15 words that have the highest weights in each topic were identified and summarized in Figure 33 below.



*Figure 33: Top 15 Words in Each Topic*

In topic 0, there are many words that are directly toxic, such as “die”, “loser” and “bad”. Additionally, there are also words like “gay”, “jew”, “hitler”, “china” and “white” which suggest that topic 0 is mostly likely about identity hate. In topic 1, the words that reflect toxicity type are “pig”, “wanker” and “moron”, which are all frequently used words in insults. The last topic is most likely non-toxic. Even though it includes words like “hate” and “vandalism”, they may have different meanings depending on context.

In addition, Intertopic Distance Map was created using LDAvis imported from pyLDAvis.sklearn to show how topics relate to each other. The top 30 most relevant words for each topic can also be generated and reviewed at the same time. The words of a topic are ranked in decreasing order according to their topic-specific probability when λ value is set to one[[3]](#footnote-4).

Figure 34 shows the top words used in the topic that have the highest number of words that are directly toxic. The topic number assigned to each topic by LDAvis is different than the topic number assigned by LatentDirichletAllocation.



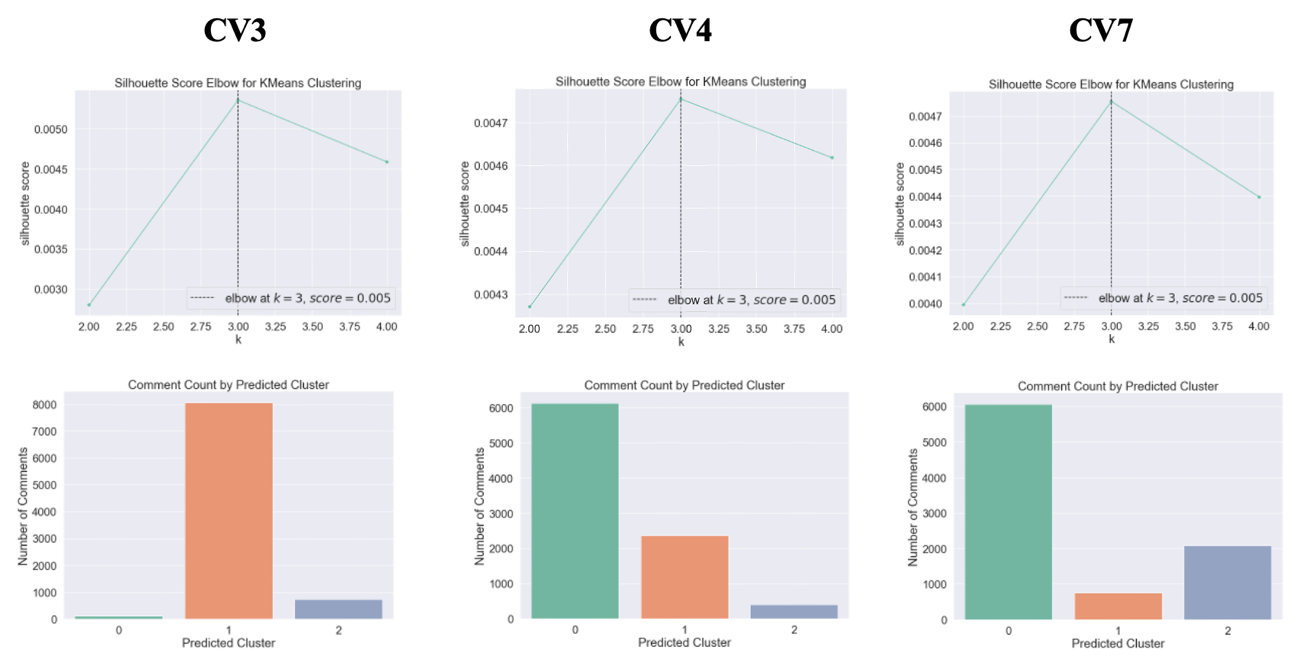
*Figure 34: Intertopic Distance Map and Top 30 Words for One of the Topics*

This list of words confirmed the findings obtained from scikit-learn LatentDirichletAllocation that one of the topics is most likely about identity hate. Additionally, this topic is the least frequent in the entire data and all three topics are not very similar to each other based on the distance observed on the left side of Figure 34.

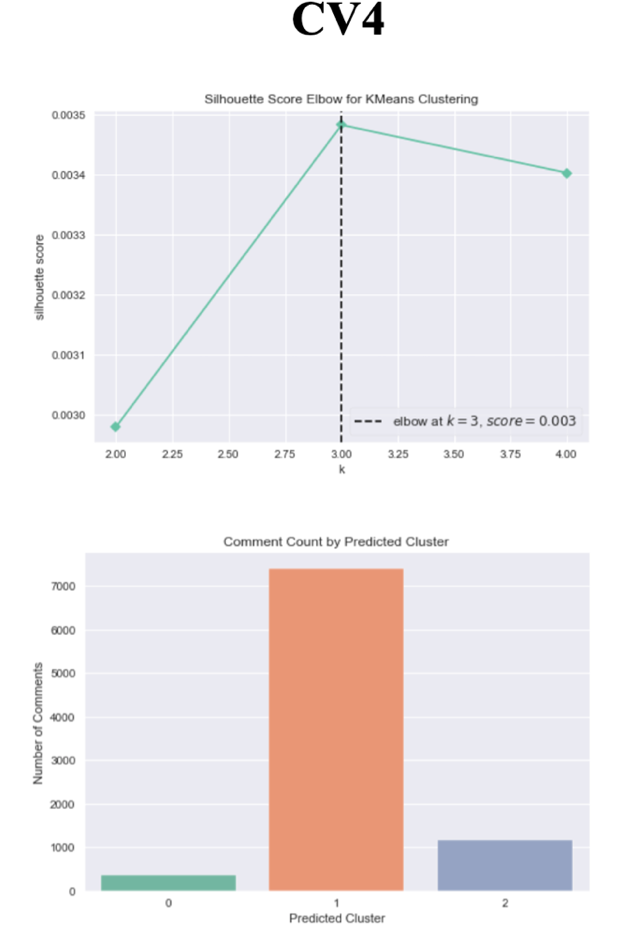
In all, results of the LDA models indicate that the data set includes non-toxic comments with no specific topic meanings, and toxic comments consisting of insults and identity hate. The three topics can also be defined as non-toxic, toxic, and severe toxic based on the words used in each topic that reflect the level and strength of toxicity.

1. K-means Clustering

As previously mentioned, KElbowVisualizer concluded the vectorized comments with the Social Group Features, resulting from CV3, CV4 and CV7, required three optimal clusters. When removing these features, CV4 was the only vectorizer that fell within this trend. Despite the optimal number of clusters not aligning with the two categories of labels, non-toxic and toxic, they were explored further to determine if subsets of toxicity were uncovered through KMeans. Figure 35 illustrates the point of inflection, supporting a cluster size of three, and its respective distribution of comments over the three clusters. Figure 36 illustrates the same information for the dataset that had the Social Group Features removed. Both figures show that despite three optimal clusters being identified, there is a dominating cluster that encompasses most of the comments. At best, CV7 in Figure 35 shows the highest distribution of comments for each cluster, indicating that this may have the most content to justify trends within each cluster.

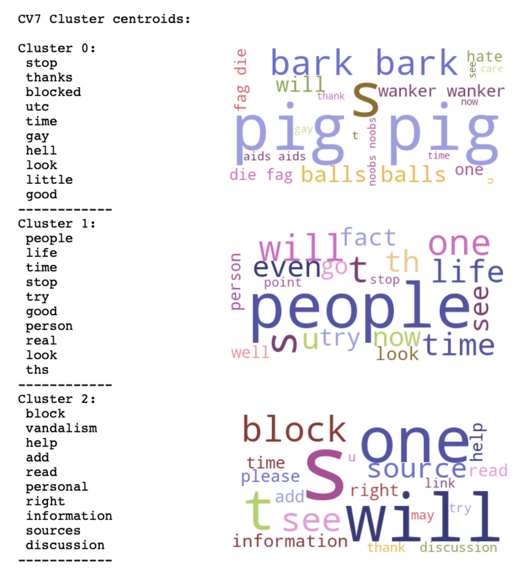
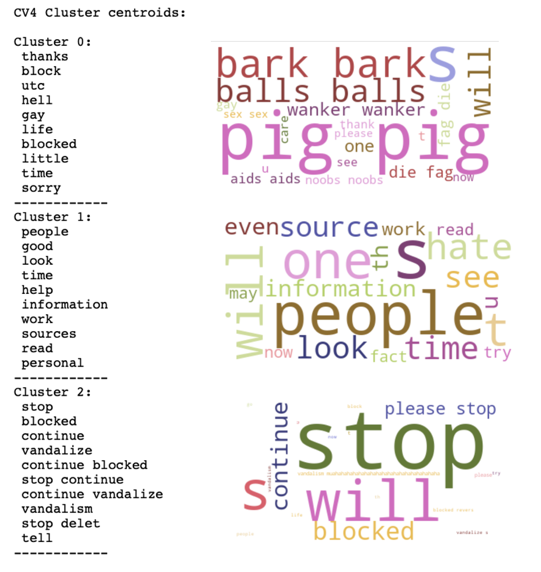
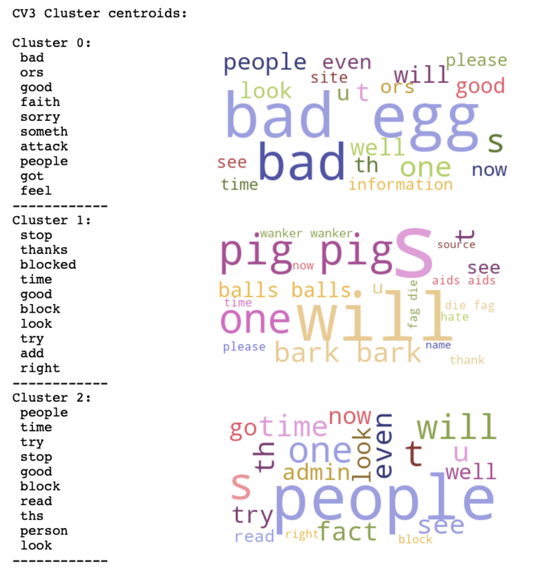


*Figure 35: Distribution of Comment Counts by Cluster- with Social Group Features*



*Figure 36: Distribution of Comment Counts by Cluster- without Social Group Features*

Figure 37 illustrates the cluster centroids for the datasets that maintained the Social Group Features. Even though the centroids are meant to show the average attributes used to group comments together, its vagueness as it pertains to toxicity prompted the generation of word clouds for the comments in each cluster. It is worth noting that the word clouds are based on the actual text in the comments and not the vectorized features, making it easier to decipher what the possible topic for each cluster is. With words like “aids”, “fag”, “die”, “wanker” and “pig”, the word clouds helped illustrate that the largest cluster in all three vectorizers was comprised of toxic comments with homophobic sentiments. Given the higher counts of comments across the three clusters in CV4 and CV7, cluster 2 in both models appear to focus on defensive language against toxicity, while the remaining cluster appears to represent non-toxic language. Words like “stop”, “blocked”, “help” and the bigram “please help” in cluster 2 support the notion that this language is being posted on behalf of a victim or a user attempting to dissuade an aggressor. Figure 38 shows that the topic trends discussed were not significantly impacted by the Social Group Features removal. The only discernable difference was the content within the cluster that represents non-toxic comments. With words like “page”, “Wikipedia” and “article”, it is evident KMeans predominantly clustered comments regarding the online platform as non-toxic.

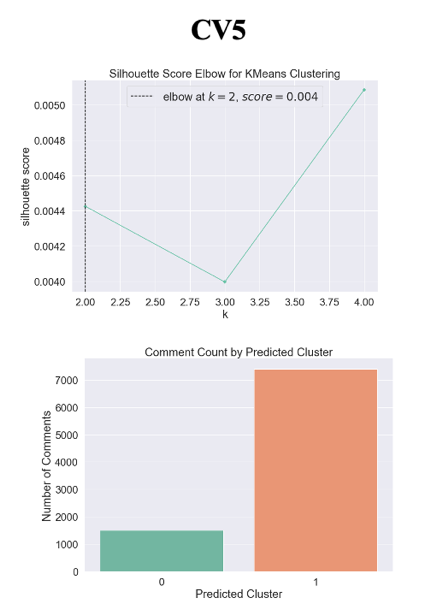


*Figure 37: Cluster Centroids & Respective Comment Word Clouds- with Social Group Features*

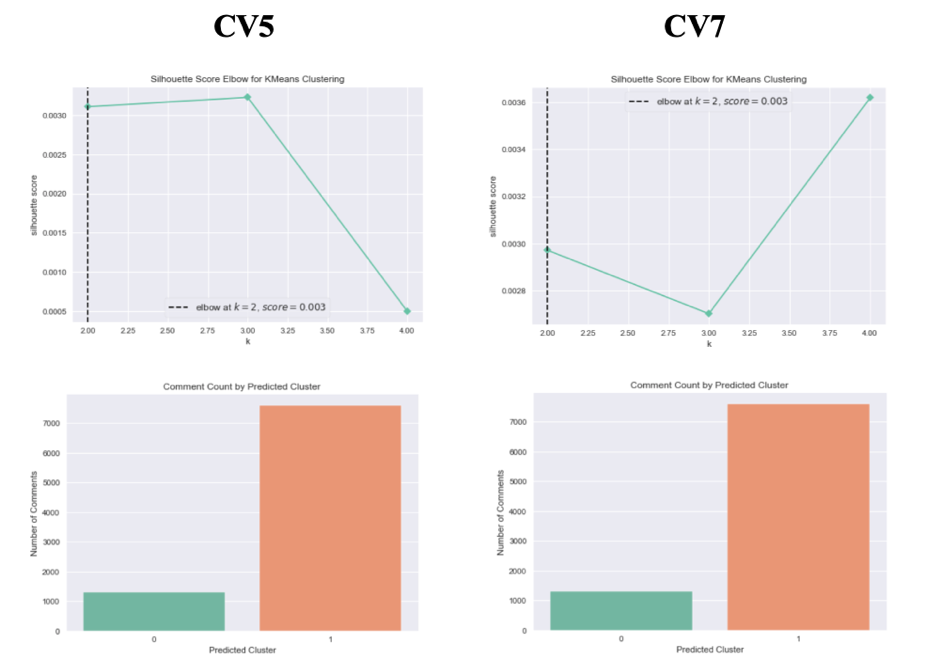


*Figure 38: Cluster Centroids & Respective Comment Word Cloud- without Social Group Features*

For the dataset with the Social Group Features, CV5 was selected to explore its predictive performance. When the Social Group Features were removed, CV5 and CV7 were also identified as potential predictive models. Figure 39 and Figure 40 both illustrate KElbowVisualer results, where the optimal number of clusters was determined to be two, along with the distribution of the comments over the two clusters. Although the number of clusters aligns with the presumed non-toxic and toxic labels, the predictive performance was not expected to be effective due to the significant number of comments grouped together in clusters 1 for all models below. As shown in Figure 1, the data was balanced and would require about an even split of comments over both clusters to have a chance of predictive success.

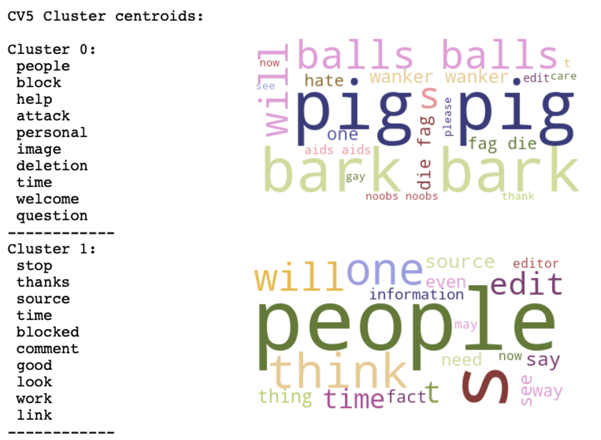


*Figure 39: Distribution of Comment Counts by Cluster- with Social Group Features*

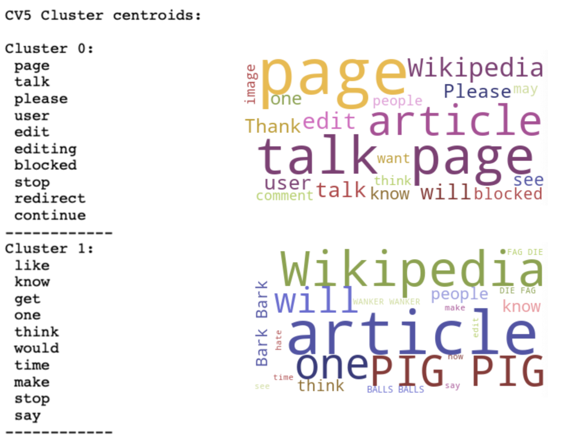


*Figure 40: Distribution of Comment Counts by Cluster- without Social Group Features*

Figure 41 and Figure 42 were included to determine whether the clusters represent toxic and non-toxic comments. For the dataset with the Social Group Features, cluster 0 appears to represent toxic comments, whereas cluster 1 represents this label for both models where the datasets had the Social Group Features removed. Since cluster 0 represents toxic content in Figure 41, all cluster assignments in this dataset were switched to ensure it correctly aligned the gold labels, where 0 and 1 represented non-toxic and toxic content, respectively.

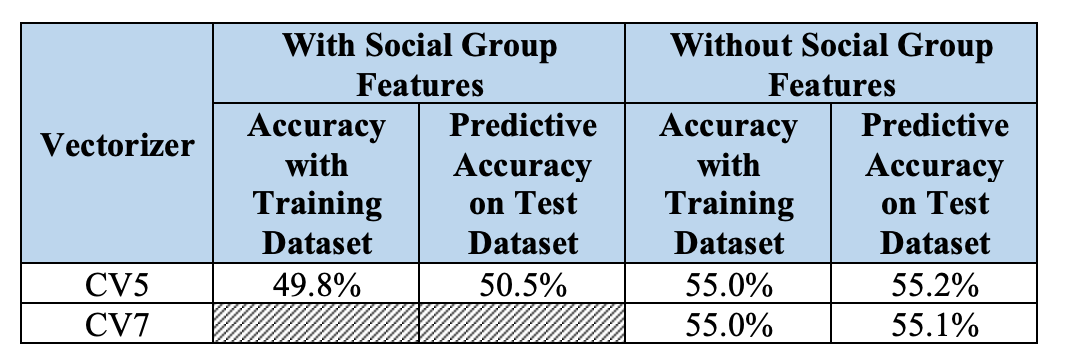


*Figure 41: Cluster Centroids & Respective Comment Word Cloud- with Social Group Features*



*Figure 42: Cluster Centroids & Respective Comment Word Clouds- without Social Group Features*

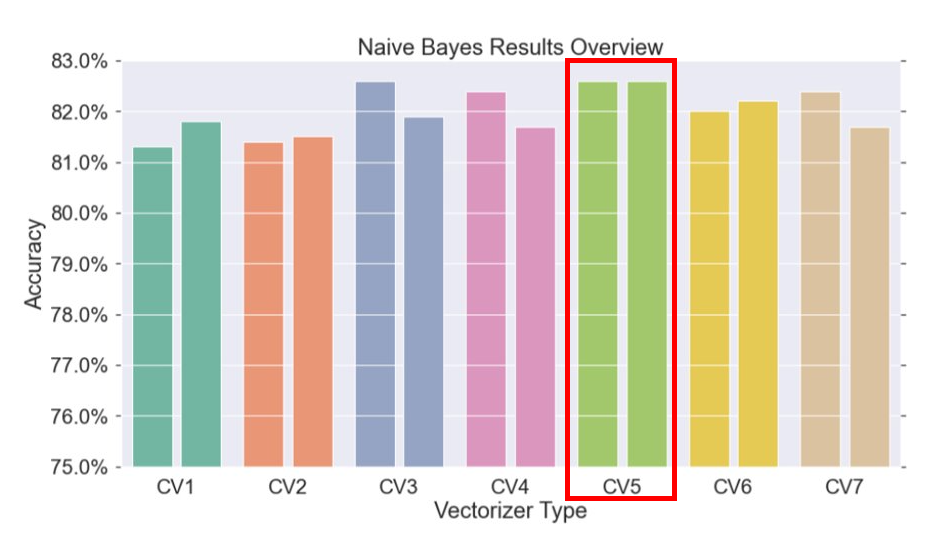
The .predict() function was used against the training and test sets to see which comments were assigned to either cluster. After converting the predicted cluster number to its corresponding label, an accuracy measure was calculated by counting the number of times it matched with the corresponding toxicity label. Table 8 lists the results for the datasets with and without the Social Group Features. The findings indicate that the removal of the Social Group Features increased the model's overall ability to categorize toxic comments by about 5%. However, with 55% accuracy, KMeans is not an effective methodology in the identification of toxic comments nor in the recognition of subsets of toxic content.



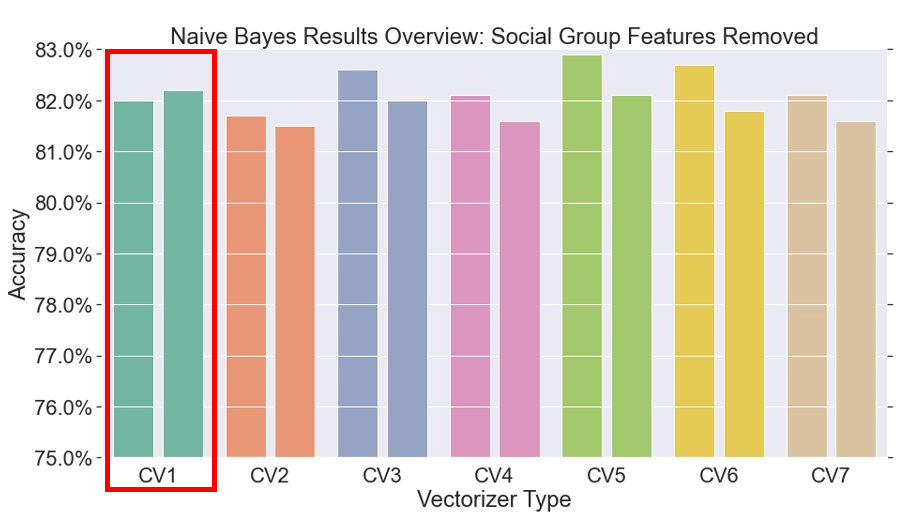
*Table 8: KMeans Clustering Predictive Accuracy Results*

1. Naïve Bayes

Figure 43 and Figure 44 illustrate the mean accuracy over 5 folds of cross validation of all seven analyses on testing and training datasets, with and without the Social Group Features. Figure 43 shows that four out of seven models performed slightly better or the same on the unseen test data indicating, where the highest predictive performance was attributed to CV5. Figure 44 indicates that the removal of Social Group Features resulted in a slight predictive decline, where only CV1 demonstrated a higher predictive accuracy compared to its training performance. Aside from being the only vectorizer where the predictive accuracy increased from its training performance, it was also the highest accuracy out of all seven vectorizers. Though the deltas are small, lower training errors and higher testing errors are normally associated with overfitting, which may be an issue if more features continue to be excluded.



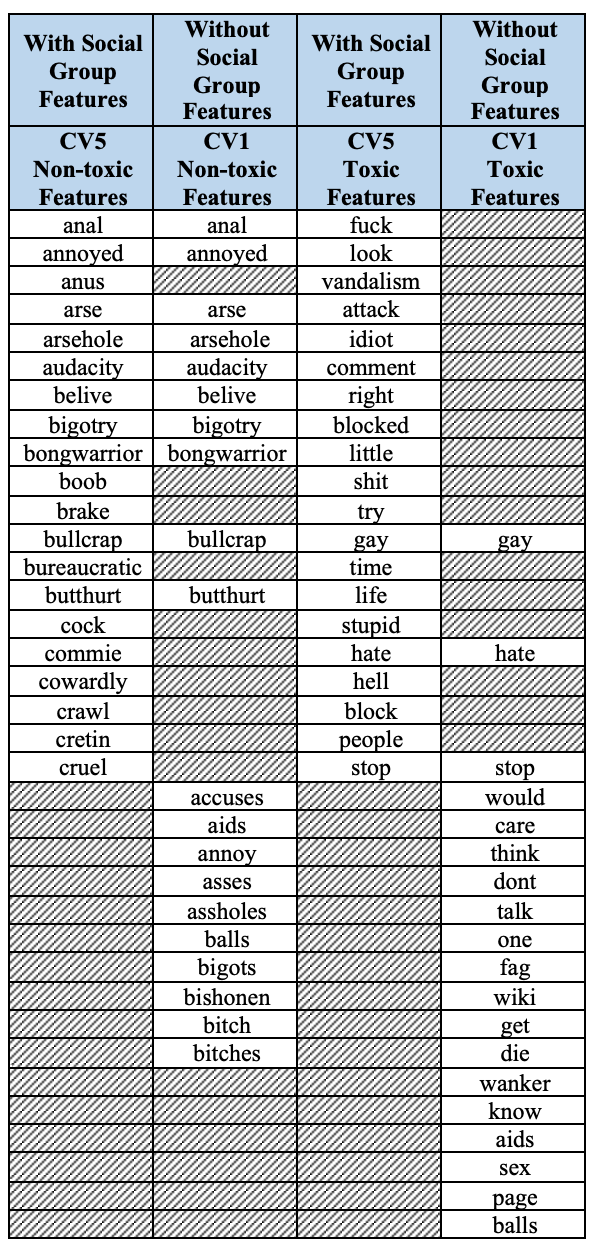
*Figure 43: Naïve Bayes Results Summary- with Social Group Features*



*Figure 44: Naïve Bayes Results Summary - without Social Group Features*

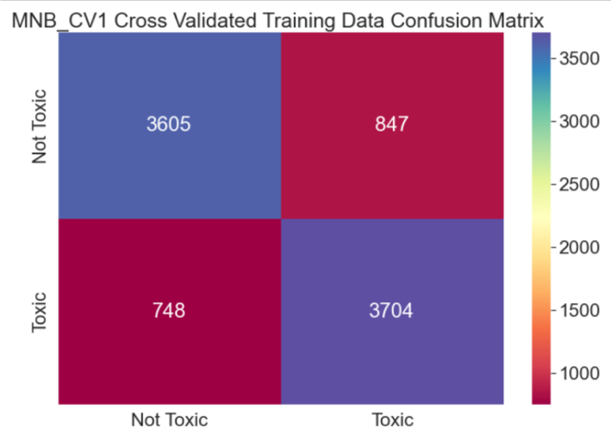
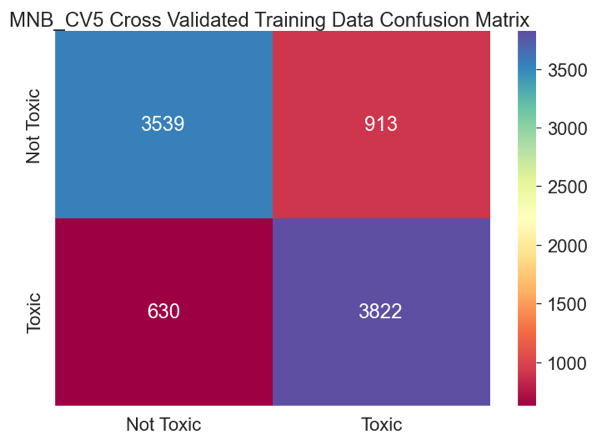
Table 9 contrasts the top 20 features by label for CV5 and CV1. Though discussed earlier in the analysis, the direct comparison between the two best Naïve Bayes models indicates that the removal of the Social Group Features impacted the top indicators for toxic comments the most. 50% of the non-toxic top features varied with the feature removal, while only 10% of the top toxic indicators aligned. When determining the quality of non-toxic indicators, the words “accuses”, “aids”, “balls” and “bitch(es)” are unique features resulting from the removal of Social Group Features. However, the features “cock”, “commie”, “cowardly”, “crawl”, “cretin”, and “cruel” are distinctive and more in line with what non-toxic language may entail. Except for “accuses”, the three remaining features may be better indicators of toxic content, especially when considering the homophobic trends in the comments and that “bitch(es)” is considered on the stronger side of toxic language. Similarly, “cock” as a CV1 feature also tends to be considered on the more offensive side of anatomy and would be expected to be a better indicator for toxic content.

When comparing the top features for toxic comments, “fag,” “wanker,” “aids” and “balls” are the most relevant indicators that differ from the results with the CV5 vectorizer. When weighing which comprehensive list of top features aligns best with the contents of toxic comments, CV5’s use of “fuck,” “vandalism,” “attack,” “idiot,” “shit,” “gay,” “stupid,” “hate” and “hell” has a slight advantage over CV1’s use of “gay,” “hate,” “fag,” “die,” “wanker,” “aids” and “balls,” especially when considering the value the words “would,” “dontm” “talk,” “wiki,” “get,” “know” and “page” actually add when identifying toxic comments. Given the frequency of auxiliary verbs, contractions, and terms unrelated to toxic identification, the quality of the features without the Social Group Features appears to be the better option for the task at hand.



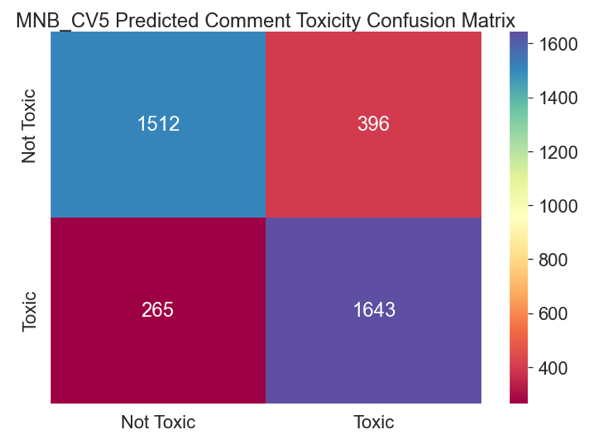
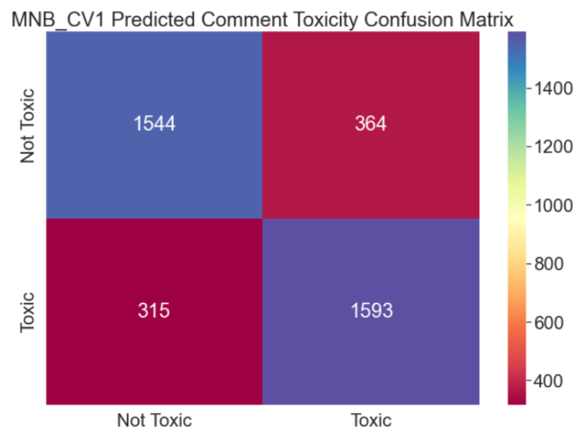
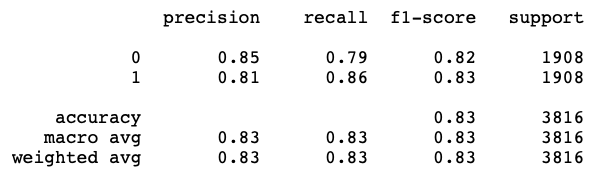
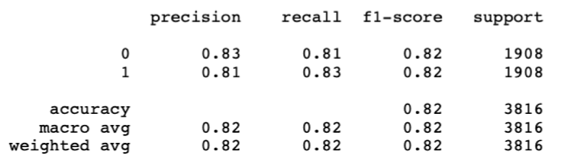
*Table 9: Best Models’ Top 20 Features Comparison*

With a mean 82.6% and 82.0% model fidelity, Naïve Bayes was more effective when the Social Group Features were not removed. Figure 45 illustrates the aggregated confusion matrices for both models after 5 folds of cross validation. With a 10% false positive rate, both models demonstrated a propensity towards identifying comments as toxic when they were not. However, the false negative rates varied by 1%, making Naïve Bayes less likely to miss toxic comments when the Social Group features were not removed. Due to the ramifications of circulating toxic content, the online platforms need to minimize the risk of misclassifying toxic content as non-toxic.



*Figure 45: Naïve Bayes Training Performance Comparison – with and without Social Group Features*

Figure 46 illustrates the confusion matrices of the trained models’ performance on new test data that contains 3,816 unseen comments. Since cross-validation with five-folds was also implemented in these predictions, these matrices are also reflective of the aggregated results for all five folds. The classification\_report of the prediction was also generated to summarize the findings of the information presented in Figure 46. Though precision, the number of correct toxic comment identifications, is important, recall offers context on how well the model did within each label. At 81%, the recall for non-toxic comments was higher when the Social Group Features were removed. However, the trend seen in Figure 45 persisted, showing that when these features were not removed, the recall for toxic content was 86%, 3% higher than with its removal. The F1 scores, which are representative of a single metric that combines recall and precision using the harmonic mean, are also higher when the dataset contains the Social Group Features. Though both approaches result in “good” model performance, emphasizing the importance of not overlooking toxic comments put the CV5 at an advantage.

*Figure 46: Naïve Bayes Predictive Performance Comparison – with and without Social Group Features*

1. Support Vector Machines

The four Support Vector Machine models trained on the Wikipedia comments corpus used 70% of the data for training and 30% for testing. SVM model accuracies, precision and recall values are shown in Figure 47. The Wikipedia comments corpus is balanced meaning there are the same number of toxic and non-toxic comments. Therefore, 50% is the baseline when evaluating model performance. Three models out of the four models can effectively differentiate between toxic and non-toxic comments. The SVM Linear model performed the best at identifying patterns in toxic comments; the SVM Linear model accuracy is 62.11%. The SVM Radial Basis Function kernel model was almost as effective as the SVM Linear model and produced an accuracy of 61.81%. The SVM Sigmoid kernel model produced an accuracy of 56.76%. The SVM Polynomial model produced a model accuracy of 50.05%; it was unable to distinguish toxic comments from non-toxic ones. Additionally, the SVM Polynomial model produced a recall value of zero for the not toxic label although it had perfect precision for the toxic label. The SVM Sigmoid model has a higher recall than SVM Polynomial for the not toxic label, but the value is relatively low when compared to the models using Linear and RBF kernels. The SVM RBF model has a higher precision value for the not toxic label, and a higher recall for the toxic label when compared to the SVM Linear model. However, the Linear SVM model has a higher recall value for the not toxic label, and a higher precision value for the toxic label, resulting in a slightly better performance predicting on the test data than the RBF kernel model.

Table

Description automatically generated

*Figure 47: Support Vector Machine Model Accuracies, Precision and Recall values*

Reviewing Confusion matrices provides further insight into each model’s performance. Confusion matrices are shown in Figure 48. The SVM Polynomial model falsely labeled all test comments as not toxic, except for two which are correctly predicted as toxic. This supports the conclusion that the model was unable to differentiate between toxic and non-toxic comments. The SVM Sigmoid model labeled most comments as toxic and very few as not toxic. The SVM RBF model accurately labeled more comments as not toxic when compared to the Sigmoid and Polynomial models. The SVM Linear model accurately labeled the most comments non-toxic when compared with the RBF and Sigmoid kernel models. However, its accuracy in labeling toxic comments requires improvement.

Table

Description automatically generated

*Figure 48: Support Vector Machine Model Confusion Matrices*

The Cost values, C, associated with the best performing Support Vector Machine models are shown in Figure 49. The SVM Linear model produced the best accuracies using a low value of C. Increasing C did not affect model accuracy. The SVM models using RBF, Polynomial, and Sigmoid kernels required higher values of C. However, at a certain point, increasing C did not improve model accuracy.

Table

Description automatically generated

*Figure 49: Cost Values associated with the best performing Support Vector Machine Models*

The Linear Support Vector Machine model performed the best at labeling toxic comments on the Wikipedia corpus when compared with the other three SVM models, once again proving the simplest model is often the best model. The Linear model also ran the fastest when compared with Radial Basis Function, Sigmoid, and Polynomial kernel models. Figures 50 and 51 are visualizations produced using indicative features for toxic and not toxic labels generated by the Linear SVM model. “Boring” is a support vector often found in toxic comments while “dig” is one often found in non-toxic comments. The SVM Linear model accurately separates comments using both features. “Painful” is a support vector often found in toxic comments, while “bond” is found in non-toxic comments. The SVM Linear model was able to distinguish toxic comments from non-toxic ones using the words “boring,” “dig,” “painful” and “bond.”

Chart

Description automatically generated

*Figure 50: Support Vector Machine Linear Decision Boundary for “Boring” and “Dig”*

*Chart

Description automatically generated*

*Figure 51: Support Vector Machine Linear Decision Boundary for “Painful” and “Bond”*

1. Summary and Discussion

Machine Learning Algorithms can identify patterns to differentiate between toxic and non-toxic comments. However, doing so is not a trivial task. Bias is naturally built into datasets and, if not careful, these biases can work their way into models causing models to predict erroneous information. One Support Vector Machine model generated a list of indicative words for toxic comments containing the word “Khoi” (see Figure 52). Reviewing the Wikipedia comments corpus reveals the word is mostly used as a name when ending a comment. For example, “-Khoi.” Out of curiosity, the word “Khoi” was typed into Google. The image below is a screenshot of the definition that appeared once the search was complete. Labeling any comments as toxic due to the existence and frequency of the word “Khoi” would prove disastrous. This demonstrates the importance of actively reviewing features and, ideally, bringing in a subject matter expert before building models to decide which features *might* hinder the model’s performance by introducing bias. The process of manually removing unhelpful features is a double-edged sword. This process involves relying on subjective individual opinions and potentially introduces bias since everyone has different opinions on which features should be kept or removed. Additionally, most words have various meanings depending on the context. How to handle bias in a dataset and model is a challenging task that researchers have been trying to overcome for a long time.

Chart, bar chart

Description automatically generated

*Figure 52: Bias in Support Vector Machine model*

Graphical user interface, text, application, email

Description automatically generated

*Image 2: Definition of Khoi (Google) [[4]](#footnote-5)*

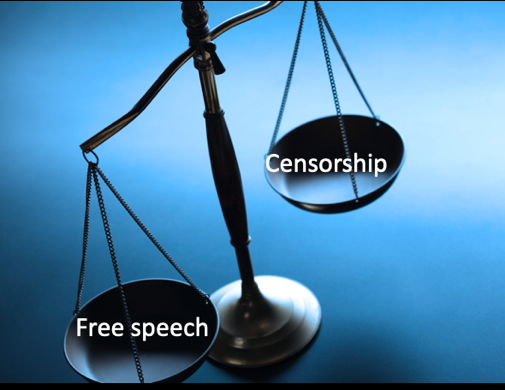
# Conclusion

The internet has grown from something that only a few people have access to, to something readily available for anyone with a smartphone and Wi-Fi. Social media has vastly changed how people communicate, indirectly introducing new forms of harassment. Children and teenagers are using the internet now more than ever, resulting in exposure to harassment at earlier ages and making it necessary to find ways to mitigate the toxicity and harassment found on the internet.

Before the internet days, children were distressed mainly by playground bullies. The internet has transformed that playground into an amusement park giving aggressors anonymous and instant access to multiple victims. As a parent, it is hard to control the content accessed by children on the internet. Additionally, it is easy to see the physical harm caused by a bully on a playground. Emotional distress is a lot harder to spot.

Censorship, as a means to stifle online toxicity, is a contentiously debated topic. Often interpreted as a violation of free speech, citizens and policymakers have struggled to synthesize this right into its digital equivalent. Britannica states exceptions to free speech include incitement, defamation, fraud, fighting words and threats, which to some degree, have all been evident in online toxicity[[5]](#footnote-6). Additional research concluding online toxicity is the virtual form of harassment further supports that there is no legal right to spread toxic content that falls within the exceptions to free speech.

The idea of ending online anonymity to deter online toxicity has also been entertained by anti-online toxicity advocates. While the prevention and removal of toxic content that is not protected under free speech is paramount, this approach would directly violate citizens’ right to free speech. In McIntyre v. Ohio Elections Commission, the Supreme Court ruled that anonymity is a fundamental component of free speech. Aside from legal rationale, this suggestion would not address the actual harm behind toxic online content. While requiring real names to be associated with online content might dissuade some aggressors from posting harmful content for fear of legal retribution or general consequences, the victims would still be susceptible to encountering these messages- they would still be vulnerable to the adverse health impacts online toxicity causes.



*Image 3: Free speech vs Censorship (source:* *https://unsplash.com/s/photos/scale)*

Companies who run online platforms need to take appropriate actions to balance free speech and content moderation. By referencing the truly protected rights of free speech, the disassociation of online behaviors with what is allowed outside the digital world can be mitigated. Though coherent regulatory guidelines are ideal, online platforms cannot wait idly for this to manifest. With bias prevention in mind, legal and linguistic subject matter experts should be contracted to curate a list of common words in subsets of toxic content that are not under the provisions of free speech. Subject matter experts would be able to offer insight on what constitutes true indicators of illegal toxicity, despite the different backgrounds and perspectives on toxicity.

# References

1. (n.d.). Retrieved September 20, 2022, from

<https://www.google.com/search?q=khoi&oq=khoi&aqs=chrome..69i57j46i131i199i433i465i512j46i131i433i512j46i175i199i512l2j0i433i512j46i175i199i512j46i512j0i512.1511j0j7&sourceid=chrome&ie=UTF-8>

1. ENC2045 computational linguistics. (n.d.). Retrieved September 20, 2022, from

<https://alvinntnu.github.io/NTNU_ENC2045_LECTURES/nlp/topic-modeling-naive.html>

1. Heart-breaking cyberbullying statistics for 2022. (n.d.). Retrieved September 20, 2022, from

<https://dataprot.net/statistics/cyberbullying-statistics/>

1. OpenWeb. (n.d.). Toxic online comments: How they happen, and how to stop them. Retrieved September 20, 2022, from

<https://www.openweb.com/blog/toxic-online-comments-how-they-happen-and-how-to-stop-them>

1. Permissible restrictions on expression. (n.d.). Retrieved September 20, 2022, from

<https://www.britannica.com/topic/First-Amendment/Permissible-restrictions-on-expression>

1. Rising levels of hate speech & online toxicity during this time ... - l1ght. (n.d.). Retrieved September 21, 2022, from

<https://l1ght.com/Toxicity_during_coronavirus_Report-L1ght.pdf>

1. Toxic definition & meaning. (n.d.). Retrieved September 20, 2022, from

<https://www.merriam-webster.com/dictionary/toxic>

1. Toxicity in the streaming world- causes, effects and solutions. (2021, July 12). Retrieved September 20, 2022, from

<https://www.richardweechambers.com/toxicity-in-the-streaming-world-causes-effects-and-solutions/>

1. Visualising topics as distributions over words. (n.d.). Retrieved September 20, 2022, from

<https://bl.ocks.org/AlessandraSozzi/raw/ce1ace56e4aed6f2d614ae2243aab5a5/>

1. Pew Research Center, January 2021, “The State of Online Harassment” [↑](#footnote-ref-2)
2. https://alvinntnu.github.io/NTNU\_ENC2045\_LECTURES/nlp/topic-modeling-naive.html [↑](#footnote-ref-3)
3. http://bl.ocks.org/AlessandraSozzi/raw/ce1ace56e4aed6f2d614ae2243aab5a5/ [↑](#footnote-ref-4)
4. https://www.google.com/search?q=khoi&oq=khoi&aqs=chrome..69i57j46i131i199i433i465i512j46i131i433i512j46i175i199i512l2j0i433i512j46i175i199i512j46i512j0i512.1511j0j7&sourceid=chrome&ie=UTF-8 [↑](#footnote-ref-5)
5. https://www.britannica.com/topic/First-Amendment/Permissible-restrictions-on-expression [↑](#footnote-ref-6)