Effect of Spam Detection Models on Real-time Tweet Sentiment

Lucas Lee, Tyson Tran, Yi Li

Abstract

Twitter is a highly influential social media platform that enables users to share and respond to short, real-time messages with a global audience. Despite its popularity, Twitter is not immune to the proliferation of spam content within conversations, particularly on sensitive and controversial social topics like abortion. These topics hold valuable insights as proxies to gauge the sentiments of the general public.

However, it has been difficult for researchers to model public opinion on Twitter in real-time while also accounting for the presence of politically charged or noise-infused tweets. Therefore, our objective is to build a pipeline that allows for streaming live tweet data and analyzing the change in sentiment by using spam-filtering models, Naive Bayes and a transfer learning model based on BERT. We will examine and compare the overall sentiment distribution of positive, neutral, and negative sentiment of all tweets and filtered tweets, as well as sentiment compound scores calculated by the NLTK Vader model. This analysis will allow us to better understand the impact of spam presence on public sentiment on social media.

1. Introduction

With the growth of technology since the 2000s, the growth of social media platforms has grown alongside. Social media platforms are a way for individuals to interact and communicate with each other. One type of social media platform that has grown exponentially are microblogging platforms such as Twitter and Facebook. These platforms are extremely valuable to everyone because it provides an outlet to gain more information for both public and private uses. It is a place that can hold opinions and can also be used by businesses or media outlets to get a general idea of people's opinions towards a specific topic. While these platforms have grown exponentially, the issue with spam has also grown exponentially. The main goal of spammers is to mislead real users through means such as malicious links or misinformation. That is in direct violation of the philosophy that Twitter is built on serving the public conversation [1].

Our objective is to understand how public sentiment on abortion will change after accounting for spam content. We believe that spam will skew the public sentiment on abortion more negatively because spam generally is used to mislead and misinform individuals. We choose to focus on abortions specifically because it is a controversial topic that contains a large amount of conversation. It is also a topic that contains lots of spam with the recent growth in public attention that it has received with the politicization of it.

To accomplish this task, we integrate spam detection into the sentiment analysis process with real-time tweets on abortion to gain insights on the effect of spam in real-time rather than having outdated results skewed by constantly changing opinions. By comparing the sentiments derived from both spam-filtered tweets and raw tweets content, we are able to gain a deeper understanding of the influence of spam on public opinion by exposing its prevalence and potentially misleading messages.

We will be using two different models for spam detection: a simpler Naive Bayes classifier and a more advanced transfer learning model based on Google's BERT. They are complemented by NLTK VADER for sentiment analysis. We will get more into this in the later sections.

2. Literature Review

Prior sentiment analysis research on Twitter (tweets text data, especially) has successfully built and compared machine learning models for analyzing tweets sentiment, in which they found ensemble models scored over 95% in accuracy, precision, and F1-score, outperforming supervised learning and convolutional neural networks models predicting sentiment using Twitter data[2]. While the study provided us with insights on model selection, its accessibility and generalizability can be further improved if combined with streaming data service, like Apache Pulsar.

Furthermore, related work on analyzing Twitter data has been through batches that analyzed tweets of certain timeframes [2, 3]. However, this lacks generalizability since the results of the sentiments are only applicable up to a certain time period, instead of tracking the continuous change in sentiment as the sentiments evolve.

With the introduction of the transformer architecture, it has dominated the field of Natural Language Processing using novel attention mechanisms with high levels of parallelism, essentially outperforming LSTMs and classical RNNs in this domain [6]. Furthermore, there have been studies that adopt existing large language models, like Google's BERT, for the task of spam classification. Specifically, a universal spam detection (USDM) transfer learning architecture for BERT has been proposed that achieved an average of 97% accuracy and F1 score of 0.96 across the Enron, Spamassain, Lingspam, and Spam text message classification datasets [7].

3. Description of data

For the purpose of this project, we acquired data from two distinct sources to develop machine learning models and address our research question.

For the training of our two models, we use a pre-labeled dataset from the University of Tennessee Machine Learning organization hosted on Kaggle. The dataset consists of data on 11,968 tweets with additional information about following and followers, activity, location of the user if they provided it, if the tweet was a retweet, and if the tweet was quality or spam.

To determine if a certain tweet was quality or spam, the dataset defines a spam tweet as those that are 'politically motivated', 'automatically generated

content', 'meaningless content', and 'click bait'. In general, these tweets are ones that do not contribute to the public conversation.

When looking at the dataset we want to be sure that it is a balanced dataset to make sure that each class of label is represented equally to prevent the model from being biased towards one class. As we can see in Fig. 1, the dataset represents close to the same number of quality and spam tweets.

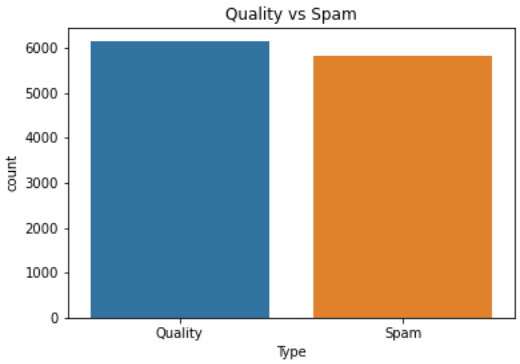

Figure 1: Quality vs Spam labels

The dataset also contains a lot of valuable information to decide what features are important to classify a tweet as spam or not. For example, we can look at the effect that an image has on a tweet being classified as spam or not by looking at the number of tweets that contain a picture versus being quality or spam. We are able to see if a tweet has an image if the tweet contains "pic.twitter". As we can see in Fig. 2, having a picture in the tweet is usually a more quality post rather than spam.

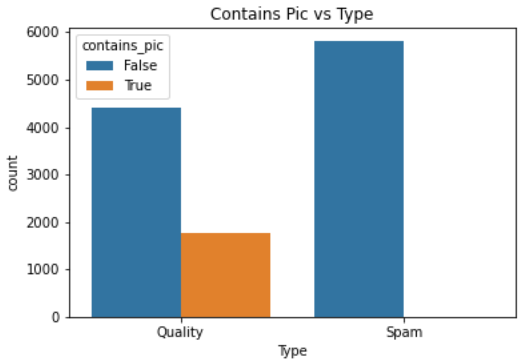

Figure 2: Picture vs Type of Tweet

To address the research question, we retrieved tweet information through Twitter API using streaming services with topic-related filtering (abortion). The collected information resembles the training dataset, having attributes such as number of followers, following, etc. of the account.

Rules under Twitter API for filtering the streaming Twitter data include an English language filter, which only considers English tweets (*lang:en*). This filter defines our target population to be English-speaking Twitter users and express their thoughts about abortion online. We also applied a keyword filter, which considers tweets containing '*abortion*' in its text content.

The dataset (figure 3) contains the information of 27,397 abortion-related tweets collected during a 10-day time frame in February.

Tweet	following	followers	actions	is_retweet	location
Over 2500 children will be killed today by abortion. This is the greatest bloodshed in America. What can you do today to stop it?	1124	317423	2117	0	Los Angeles, CA

Alabama law contains a massive loophole that allows abortions to still happen legally - Sign the petition to abolish abortion today! #AbolishAbortion #EAA #alpolitics	17	7	0	0	Alabama

Figure 3: Sample Dataset

4. Methods

4.1 Proposed Pipeline

Our framework for the pipeline (figure 4) has four main components: data retrieval, spam detection, sentiment analysis, and visualization.

This pipeline is facilitated through the use of Astra Streaming, which is built upon Apache Pulsar. Pulsar utilizes the publish-subscribe pattern: producers publish messages to a topic, and consumers subscribe to those topics and process the incoming messages. Additionally, we also employ the use of Pulsar Functions, which are lightweight processes configured by user-supplied logic that consumes messages from one topic and publishes to another according to said logic.

Firstly, we have a producer that requests real time tweets from the Twitter API v2 using the filtered stream end point, retrieves relevant attributes, and publishes the formatted data to a Pulsar topic (Raw Tweet Topic).

Secondly, we have consumers subscribe to the Raw Tweet Topic. Two consumers will use ML models to filter out spam tweets then perform sentiment analysis. One would remain as a control group and perform the sentiment analysis directly. The result will be delivered to a data source (Google Spreadsheet, for example) and further analyzed and visualized on Grafana.

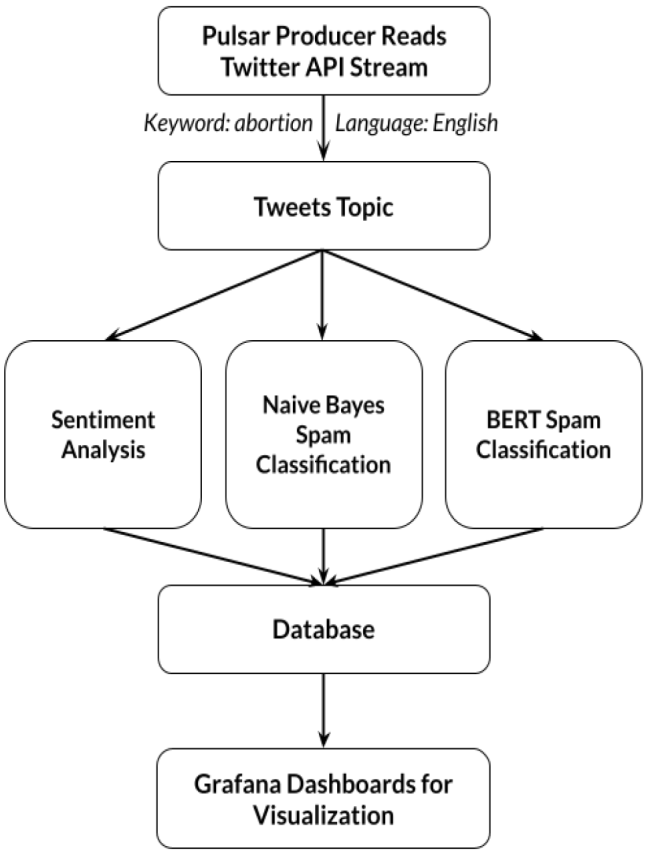

Figure 4: Streaming Pipeline

4.2 Data Retrieval

Twitter API offers two different types of streams: sampled stream and filtered stream. Sampled streams are not linked to any topic and it takes a 1% sample of all tweets in real-time. Filtered streams are linked to a topic by user-generated rules. The rules are used to match tweets that follow certain attributes. In our case, we used the filtered streams endpoint to match tweets in English and contain the keyword

‘abortion.’ Limiting the language to only English generates daily data granularity ~50k tweets. However, the selection of language introduces a potential risk to the representativeness of spam labeling and sentiment distribution.

To obtain specified information, such as number of account followers, following, and whether a tweet is retweeted, we reviewed all relevant attributes and included *public_metrics* of tweets.

With the filtered stream end point, we receive a response as a stringified JSON format. Within the Pulsar producer, we processed the nested data and sent out formatted JSON data with predefined Schema to Consumer for downstream classification and sentiment analysis.

4.3 Data Pre-Processing

Original tweet texts contain links, emojis, retweet marks, etc. that need to be cleaned for sentiment analysis. Below are a few steps that we take to preprocess tweet texts. More examples can be found under section 4.2.1.

- Remove links, hashtags, mentions
- Lowercase letter
- Remove extra spaces, punctuations

4.3.1 Data Pre-Processing Naive Bayes

When a tweet is extracted from Twitter, it comes in many different forms which would not work well as an input for our data. If we do not pre-process the data, the concept of “Garbage in, garbage out” would apply. Since we are using a Naive Bayes classifier, we want to remove the features of a tweet that are unique and unlikely to be seen in other tweets because it follows Bayes theorem which looks at probabilities of an event based on the occurrence of another event. In our case, it will be the probability that it is spam or not based on the probability of certain words appearing. As we can see in Fig. 5, the tweets contain a lot of different elements that are related to the use of social media, but are not related to the actual message that they are attempting to convey. Therefore, we only want to keep the base of the tweet and remove all other elements such as hashtags, links or symbols. We also want to make the structure of the tweet more uniform because tweets tend to be informal and have many errors such as multiple spaces between words. As we can see in Fig. 6, there is more uniformity between the tweets and still holds the message from the person who tweeted.

Tweet	Type
It's the everything else that's complicated. #PESummit #PXpic.twitter.com/Jsv6BAFQMl	Quality
Eren sent a glare towards Mikasa then nodded and stood up to go help his lovely girlfriend @SincerePyrrhic. Once he arrived in the kitchen_	Quality
I posted a new photo to Facebook http://fb.me/2Be7LiyuJ	Quality
#jan Idiot Chelsea Handler Diagnoses Trump With a Disease https://t.co/k8PrqcWTRI https://t.co/dRN35xtSJZ	Spam

Pedophile Anthony Weiner is TERRIFIED of Getting Beaten Up in Prison https://t.co/g3bU9Q4gAg	Spam

Figure 5: Before Processing Tweets

Process_tweet	label
its the everything else thats complicated	0
eren sent a glare towards mikasa then nodded and stood up to go help his lovely girlfriend once he arrived in the kitchen	0
i posted a new photo to facebook	0
idiot chelsea handler diagnoses trump with a disease	1
pedophile anthony weiner is terrified of getting beaten up in prison	1

Figure 6: After Processed Tweets

4.3.2 Data Pre-Processing BERT Transfer Learning

Before the raw tweet is fed into the transfer learning model, the raw text must be transformed to three tensors: numeric token IDs, input masks, type IDs. Additionally, padding, special tokens, and masks must also be added and created for feature homogeneity to BERT's original training data. This is done through using the corresponding preprocessor to the BERT encoder, and is simplified with the user of Tensorflow Hub's implementation of multiple downloadable preprocessors. It's worthwhile to note that since BERT was trained on documents that have a max token length of 128, all tweets are truncated to 128 tokens (words) during the preprocessing stage. While Twitter currently only supports 280 characters (roughly 56 tokens) it does have potential future plans to increase the character limit to 4000 characters (around 800 tokens). This requires a different padding and masking preprocessing mechanism for BERT for future replication projects.

4.4 Classification

To achieve our goal, we employed two different spam classification models, then each model will feed their results to NLTK VADER for sentiment analysis. We chose to implement two different spam classification models to compare the performance of the BERT transfer learning model by comparing it to the simpler Naive Bayes model. We also chose to use the pre-trained VADER model, which was optimized for classifying sentiments in social media microblogs.

4.4.1 Spam Classification Naive Bayes

Naive Bayes classifier is a probabilistic classifier based on Bayes theorem. It finds the probability that a tweet is spam given that it contains certain words. It finds the probability that each word is spam and multiplies it together to determine if a tweet is spam or not.

We choose to use a Naive Bayes classifier because it is a simple and effective model to work as a baseline solution to compare to a more advanced model. It is an effective solution because certain words are more likely to be used in a spam tweet over a quality one. We also chose to use a

multinomial Naive Bayes model over a Bernoulli Naive Bayes model because it works off of occurrence counts rather than the presence or absence of a word.

Before training the model we split the data into a 75-25 split using "train_test_split" from the scikit-learn module to have data to test if the model is performing correctly. To train the model, we have to take the preprocessed data and transform it into a form that works with the multinomial Naive Bayes model. In our case, we use "CountVectorizer" from the scikit-learn module to transform the text to a matrix of token counts for unigrams. We use unigrams instead of bigrams or more because we are interested in the probability of each individual word being spam or not. To optimize the multinomial Naive Bayes model, we need to choose the best alpha value for the model, which we get by performing grid search. The best alpha value for the model was .5 and after testing the accuracy of the model, we received an accuracy of 79.48%. In Fig. 7, we can see the confusion matrix on how the model performs on its predictions by class. The testing dataset consisted of 2,992 tweets.

	Not Spam	Spam
Predicted Not Spam	1267	248
Predicted Spam	366	1111

Figure 7: Naive Bayes Test Data Confusion Matrix

4.4.2 Spam Classification BERT

The model architecture used is inspired by state of the art works of transfer learning on BERT for NLP tasks. [6, 7].,

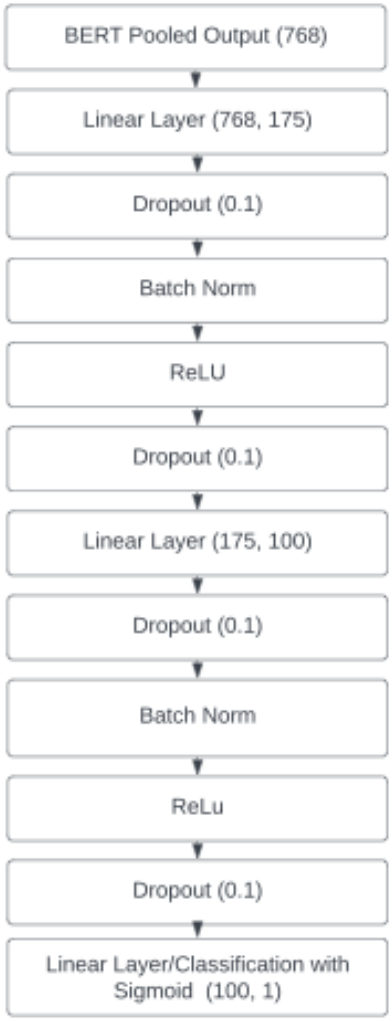

Figure 8: BERT Spam Classification Architecture

After the text has been preprocessed, it's fed into the BERT encoder (bert_en_uncased_L-12_H-768_A-12/4), which contains 12 layers, 768 hidden units, and 12 attention heads. In figure 8, the BERT output of size 768 is then reduced to dimension of 175, where batch dropout, batch normalization, with ReLU activation. Before feeding into the next layer, dropout is performed again. The next layer is the same with dimension of 100, and results in the next layer/prediction of spam or not spam of a tweet. Dropout is introduced to prevent

over-dependence of certain neurons and to improve robustness. Batch normalization is used to normalize the activations within the layer to mitigate internal covariant shifts to improve the model's stability. ReLU is employed to learn non-linear patterns within the data. Finally, a sigmoidal activation layer is used to cast the output to a probability between spam(1) or no spam(0). The architecture results in 85.3% accuracy, a significant improvement over the naive bayes model. As seen in figure 9, the main improvement is actually the reduction of false negatives — tweets that were predicted as non-spam that were actually spam. This hints that the attention mechanism provides meaningful contributions by semantically understanding individual tokens in relation to their surrounding text.

	Not Spam	Spam
Predicted Not Spam	1137	57
Predicted Spam	401	1387

Figure 9: BERT Spam Detection Confusion matrix

4.4.3 Sentiment Classification NLTK

We have three consumers that subscribe to the Raw Tweet Topic, where each receives the tweet data. Each consumer will then first perform spam classification, then classify the sentiment of the stream of tweets in one of three categories: positive, neutral, and negative.

One consumer directly classifies the sentiment of the tweest since it has no designated spam filtering mechanism. This provides the baseline for monitoring how spam affects the sentiment of the topic.

A second consumer utilizes the naive bayes model to filter spam tweets, then classifies the sentiments of the tweet.

A third consumer utilizes the BERT transfer learning model to filter out spam tweets and classifies the sentiment of the tweet.

We choose to use the pre-trained VADER for sentiment analysis since it was designed and trained with microblog-like data, just like Twitter [5]. VADER is a lexical based model that maps words to sentiment by building a dictionary of sentiments, also known as a lexicon.

Vader uses a large lexicon that includes emoticons and slangs which are fit for platforms like Twitter. For example, both "sux" and "sucks" have the same sentiment polarity score assigned by VADER. More specifically, it assigns a numerical sentiment rating of each term from -4 to 4 to account for polarities of sentiment.

VADER then evaluates the sentiment of a sentence or corpus through the normalizing the sum of the sentiment score for each lexical feature (words and emoticons), outputting a number between -1 and 1.

The sentiment scores of VADER also retain contextual information about the lexicons within the sentence through 5 novel heuristics. Namely, it uses punctuation marks, capitalizations, degree modifiers, usage of the word "but", and tri-grams to gain contextual awareness [4].

In the Pulsar consumer, we use the compound polarity score from VADER's, to assign a sentiment score to each incoming tweet. Since the compound score is the numerical representation of the polarity of a

sentence between -1 and 1, we classify the corresponding sentiment through the following:

- Positive: compound score >= 0.05
- Neutral: -0.05 < compound score < 0.05
- Negative: compound score <= -0.05

We choose to have text with a polarity score between -0.05 and 0.05 be rated as neutral to account for opposite terms not being able to cancel each other out fully due to the valence score not being gauged exactly the same from word to word.

5. Results

The accuracy of the Naive Bayes model trained on the Kaggle training dataset is 79.5%, and the accuracy of the transfer learning model based on BERT is 85.3% on the test dataset.

Using those two models, we predicted whether collected *abortion* tweets were spam or not.

Out of 27,397 *abortion* tweets, the spam classification results can be found below in figure 10. More than 69% of tweets were classified as 'spam' by both models, and 50% were manually classified as spam tweets.

Type	Spam (1)	Proportion	Quality (0)	Proportion
Naive Bayes	19,061	69.6%	8336	30.4%
BERT	21,837	79.7%	5560	20.3%
Manual Labeling (300 tweets)	150	50%	150	50%

Figure 10: Spam Classification

We then conducted NLTK sentiment analysis on all 27,397 tweets and filtered tweets. The sentiment distribution can be found in figure 11.

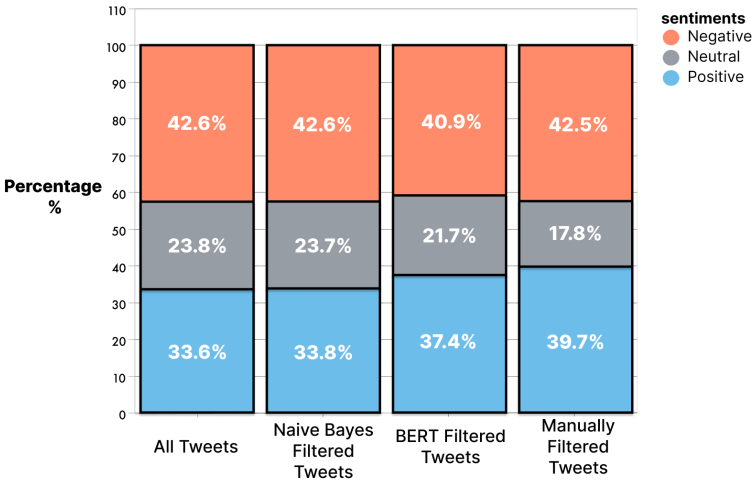

Figure 11: Sentiment Distribution of All/Filtered Tweets

The results revealed that negative sentiment was the most prevalent, constituting over 40% of all sentiments, followed by positive sentiment, while neutral sentiment was the least common. It also suggests a similarity on the sentiment distribution, regardless of whether spam is present or not. The maximum difference is 6%.

Besides categorical sentiment, we scrutinized the exact sentiment compound score distribution as well. Figure 12 and 13 compare the sentiment compound score (SCS) between all tweets (unfiltered) and filtered tweets by Naive Bayes and BERT.

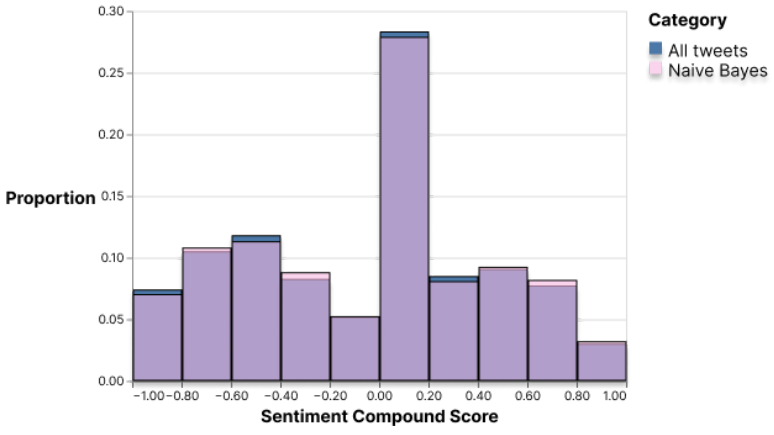

Figure 12: Comparison of SCS between all tweets and Naive Bayes filtered tweets

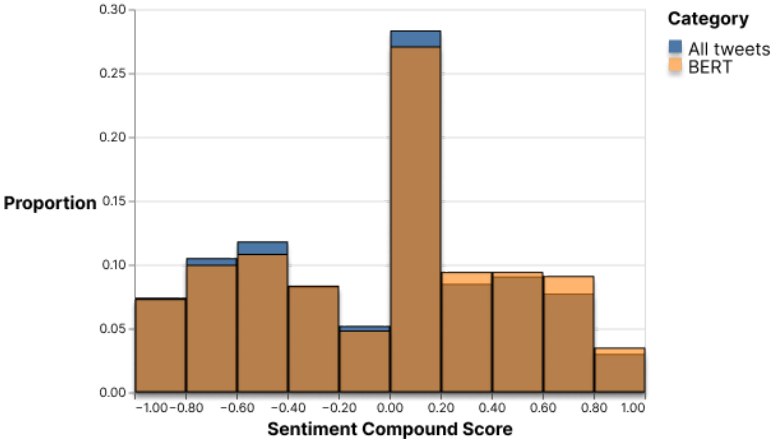

Figure 13: Comparison of SCS between all tweets and BERT filtered tweets

The distributions of SCS of both spam-filtered models show a similarity to the distribution of all tweets (unfiltered), suggesting the sentiment similarity of filtered tweets to that of all tweets.

Figure 14 shows the descriptive statistics of SCS under different scenarios.

Type	**Mean SCS**
All tweets	-0.070769
NB filtered quality tweets	-0.064845
NB filtered spam tweets	-0.073363
BERT filtered quality tweets	-0.042129
BERT filtered spam Tweets	-0.078061
Manually labeled quality tweets	-0.032911
Manually labeled spam tweets	-0.173945

Figure 14. Mean of SCS under different conditions

Figure 13 depicts the distribution of Sentiment Compound Scores (SCS) for all tweets, Naive Bayes (NB) filtered tweets, and BERT filtered tweets. However, it didn't obviously show that the distribution shift results in lower SCS values for filtered tweets compared to all tweets. The mean SCS for NB filtered tweets, BERT filtered tweets, and all tweets (non-filtered) are approximately -0.065, -0.042, and -0.071, respectively.

The observed pattern, with spam-filtered tweets exhibiting lower SCS than unfiltered tweets, aligns with the SCS of manually filtered tweets, which average around -0.033. Furthermore, the discrepancy in SCS between BERT filtered tweets and all tweets (0.029) is notably larger than that between NB filtered tweets and all tweets (0.006). This marginal difference for the NB spam classifier implies that a user employing an NB-based spam filter on Twitter would likely perceive no noticeable change in sentiment within the discussions compared to not using any filter.

By examining the mean SCS for abortion-related tweets, BERT model demonstrates generalizability beyond the training dataset. The observed mean SCS of BERT filtered tweets (-0.042) closely aligns with the manually filtered (ground truth) mean (-0.033), with a difference of 0.009. Comparatively, the difference between NB and manually filtered tweets is 0.03, suggesting that BERT's spam classification

is a more desirable approach than utilizing the NB spam classifier.

6. Discussions

We trained models on a collected dataset of tweets that covers a variety of topics and used those to predict completely new data.

In order to find the *ground truth* for abortion-related topics, we manually labeled 300 tweets. While our manual labeling process is useful in providing ground truth, it may be subject to bias due to the subjective nature of evaluating spam content. Figure 15 refers to a reply tweet to an abortion-related tweet, which is manually labeled as a spam tweet that falls into the 'politically motivated' category of spam.

Figure 15: Manually labeled spam tweet

Our group's evaluation standards may vary among members and deviate from those in the training dataset, potentially leading to inconsistencies in labeling. We also recognize that our manual labeling process incorporated supplementary information, such as context and user profile details, which is unavailable to text-only models. This additional data may contribute to subjectivity, potentially affecting the objectivity and generalizability of our findings.

Despite these limitations, our study demonstrates the potential of utilizing a spam-labeled, general dataset encompassing a wide array of topics to predict spam content within a specific domain. Furthermore, the alignment of our manually labeled data with the overall trends in the dataset lends credibility to our results.

Our next step will focus on incorporating more topics and features to the existing models.

7. Conclusion

Our analysis of streamed tweets on abortion-related topics reveals that a significant proportion of tweets contain spam content, as determined by both Naive Bayes and transfer learning models. Despite the prevalence of spam, we found that the sentiment distributions of all tweets and manually/model filtered quality tweets were quite similar, with the majority expressing negative sentiment with a maximum sentiment difference of 6%. Furthermore, examination of the sentiment compound score suggested that the presence of spam did not have a significant impact on the overall sentiment towards abortion-related topics. However, upon closer inspection, we observed a slightly more negative sentiment associated with tweets containing spam, as indicated by the mean sentiment compound score. This finding highlights the potential for spam to subtly influence public sentiment and underscores the importance of identifying and filtering out such content in future analyses.

References

1. https://about.twitter.com/en
2. Employee Sentiment Analysis Towards Remote Work during COVID-19 Using Twitter Data. (2022). *International Journal of Intelligent Engineering and Systems (Vol. 15, Issue 1)*. https://doi.org/10.22266/ijies2022.0228.08
3. Social Media Users' Opinions on Remote Work during the COVId-19 Pandemic. Thematic and Sentiment Analysis. (2020). *Information System Management (Vol. 38, Issue 4)*. https://www.tandfonline.com/doi/full/10.1080/10580530.2020.1820631?scroll=top&needAccess=true
4. Exploring Public Sentiment on Enforced Remote Work during COVID-19. (2021). *Journal of Applied Psychology (Vol. 105, No. 6, p. 797-810)*. https://psycnet.apa.org/fulltext/2021-56704-001.pdf
5. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media 8 (1):216-25*. https://doi.org/10.1609/icwsm.v8i1.14550.
6. Transformers: State-of-the-Art Natural Language Processing. (2020). https://doi.org/10.18653/v1/2020.emnlp-demos.6
7. Universal Spam Detection using Transfer Learning of BERT Model. (2022) https://doi.org/10.24251/HICSS.2022.921