Predicting Polarization From News Outlets Tweets

Stephanie Ramos
MSCAPP Candidate
University of Chicago
stephanieramos@uchicago.edu

Jesica Ramirez Toscano
MSCAPP Candidate
University of Chicago
jramireztoscano@uchicago.edu

Kelsey Anderson
MSCAPP Candidate
University of Chicago
kjanderson@uchicago.edu
github.com/advanced-ml-project/project

Abstract

Media bias has been a topic of interest for policy makers and politicians particularly since the 2020 US presidential election. Society appears to be increasingly polarized and hostile in online interactions. Online spaces have become the target of criticism and fears about cyberbullying, trolling, fake news and the potential for vulnerable individuals to become violently radicalized. With this in mind, we built different supervised machine learning models to test if the text extracted from tweets posted by news media accounts is predictive of the polarization in the comments they receive. We used logistic regression, recurrent neural networks and convolutional neural networks to make our predictions. Using these models and data extracted from Twitter, we were able to predict polarization in comments with 65% accuracy. Our research aims to contribute to creating a more civil and less polarized space for discourse on social media.

1 Background

Social media and the news are intimately intertwined in today's public arena. The sharing and resharing of events brings news to our fingertips with lightning speed. Issues of bias and polarization are studied in a wide variety of fields. Researchers are often concerned with several aspects of bias and polarization: the bias present in news media articles, social clustering along ideological lines which reinforces increasingly skewed versions of reality ("echo-chambers") and the erosion of meaningful civil discourse in the general public.

Social media spaces, contrary to popular belief, may be more pluralistic than expected (Tucker, et. al, 2018). People tend to have broad networks and are exposed to many viewpoints on political issues at low levels of depth through non-political threads. Explicitly political topic threads, however, are more polarized in nature, with the most emotional or sensational content experiencing higher levels of user engagement (Tucker, et. al, 2018). The people who are most active on social media sites and have the most followers also tend to hold the most extreme views (Hong Kim, 2016). Various researchers find that organically viral content is somewhat rare. Typically a traditional news outlet, public figure or social media influencer will highlight an issue and this action sets off a cascade of subsequent conversations (Tucker, et. al, 2018).

Since news is often accessed through social media, political news is often made visible in these channels by powerful institutions like news media and that these conversations tend to be the most polarized, it seems worthwhile to look at the factors that create or mitigate toxic or polarizing threads.

Preprint. Under review.

2 Related Work

A number of machine learning researchers have taken on projects related to media bias and the elements of language that might be classified as politically skewed. Scholars have also attempted to model polarity through community network structures and hate speech by using sentiment analysis. Our project draws on a number of these concepts to attempt a language model that explains resultant polarization.

Studies of media bias typically use the political leanings of news outlets themselves as ground truth for learning tasks (examples include: Jachim et. al, 2021; Lee et al., 2021; Hong Kim, 2016). These political designations tend not to be granular in nature, but rather large categories like "liberal," "moderate" and "conservative." As such, results from such studies are interesting and encourage our theoretical linkage of media bias with social polarization, but handle content labelling in too coarse a way to be useful for our purposes.

Most noticeably prior research has conceptualized polarization itself in different ways. Many define it as the formation of separate communities around a topic. For example, Guerra et. al examines three potential measures of community polarization: modularity, boundary node connections and popularity of boundary nodes. They use metrics computed on node graphs created with an open source automated graph visualization tool called Gephi to analyze six different datasets from Twitter, Facebook and blogs.

From the graphs they calculated the degree to which separate communities exist (modularity) using a formula shared with previous literature. Guerra et. al critique modularity, however, because it fails to distinguish between communities that are simply somewhat separate (in their datasets, for example, New York football fans and New York basketball fans) from communities where antagonism exists between groups (another sports example: between the supporters of rival Brazilian soccer teams). For polarization to exist, they argue, antagonism is an important distinguishing characteristic. To better capture this, they propose two measures of connections between groups: counting the proportion of in-group versus out-of-group connections in boundary nodes and calculating correlation between the popularity of a node and it's closeness to a boundary between groups.

The authors find that datasets they expected to exhibit polarization do show greatly reduced out-of-group connections on the boundary (Guerra et. al, 2013). They also find that, to a lesser extent, their hypothesis about popular nodes being internal ones for polarized networks and boundary nodes for non-polarized networks is generally true. The exception in this second metric was for US political blog posts. For news blogs, boundary nodes did tend to be in-group facing, however, among them there was a large number of popular boundary nodes. The authors explain this as the presence of news media blogs, which exist at the boundary because they are commonly referenced by both sides of a debate.

The Guerra et al. article fails to examine data at a lower level than network connections (i.e. retweets, follows or friend connections). So, while their findings are interesting, it is important to note that antagonism is semantic. A user might share a news story in order to promote it or mock it, depending on the caption or comment text they provide.

A similar graph-based approach was used by O. de Zarate et al. (2020), who measured controversy in 30 twitter datasets in 6 languages. To determine the "degrees of controversy", the data goes through four phases: First, the graph building phase, in which they build a conversation graph that represents a retweet-graph of a single topic discussion. Second, the community identification phase involves using a graph-clustering algorithm, Louvain, to determine the communities involved in the discussion. In the embedding phase, each user is embedded into a corresponding vector encoding syntactic and semantic properties of the posts. To perform this step the authors selected two models among the most advanced ones, namely Fasttext and BERT, which embed texts into fixed dimension vectors encoding semantically significance and meaning. Finally, to compute the controversy score, they select some users as the best representatives of each community's point of view. They used the HITS algorithm to estimate the authoritative and hub score of each user, and selected the central users, which correspond to the users with the 30% highest hub and authoritative score. Then the controversy score is defined by the distance between the central users and the rest of the users in each cluster.

This type of approach gives useful and promising results in measuring polarization, however it also requires the data being labeled to test the results. In this paper, the authors defined certain topics to be controversial, and from that they test if the data pipeline results were correct. So there isn't an established ground-truth.

Another approach to polarization revolves around the strength of feeling, or sentiment, present in a body of text. This is common in the exploration of toxic or harmful speech. For example, Salminen et al. (2018) collected comments from a YouTube channel and Facebook page of a major online news organization and created training data by identifying hateful comments using human judgment. Then they created a taxonomy of different types and targets of online hate, and trained machine learning models to automatically detect and classify the hateful comments in the full dataset.

They collected data from a major online news and media company that posted several videos per day on YouTube and Facebook and received thousands of comments per video. These comments were collected using YouTube and Facebook APIs. They pulled 137,098 comments from videos posted on YouTube and Facebook, in the period of July-October 2017, and focused only on English comments. They explored hate in the dataset by building a dictionary based on public sources of hateful words and a qualitative analysis. Searching with that dictionary, they found that 22,514 comments (16.4%) contained hateful content.

Then they proceeded to assign labels to a portion of the data. They considered linguistic attributes when annotating. Swearing, aggressive comments, or mentioning the past political or ethnic conflicts in a non-constructive and harmful way, were classified as hateful. To divide the comments into groups depending on their target and type, they used manual open coding, which uses human judgment for defining the categories. The taxonomy has 13 main categories and 16 subcategories (29 in total). The main categories include targets and language, 9 describing targets (political issues, religion, racism, etc.) and 4 the type of language (accusations, humiliation, swearing, promoting violence).

To perform the supervised classification, they developed three sets of feature categories. The feature categories are n-gram, semantic and syntactic, and distributional semantic features. Before feature extraction, preprocessing was performed, removing stop words from comments and stripping the tokens of any trailing special characters or space. They experiment with a set of machine learning algorithms to perform multilabel classification of the dataset. In this experiment, they used the 5,143 labels annotated using the taxonomy. This dataset was split into training and testing (33% for testing) the classification models. Five different models were used: Logistic Regression, Decision Tree, Random Forest, Adaboost, and Linear Support Vector Machine (SVM). For each model, they tuned the parameters using scikit-learn's grid search method in Python.

The average F1 score of the 21 categories and subcategories is used to report the overall performance as it combines both the precision and recall. The average precision of the best model, SVM, was 0.90, and recall 0.67 (avg. F1 score = 0.79). The recall score indicates the model is struggling to classify some categories, most notably religion (recall=0.3). Then they apply the classifier to the full dataset to classify the comments by the main categories of hate.

Most common is hate against the media (17.0% of instances), mostly against the particular news outlet which is referred to as "propaganda," "fake news," and "lies." Another popular target is the Armed Forces (16.9%), particularly the police. The most typical language type is humiliation (31.5% of language observations) but swearing (29.3%) is also common. Somewhat alarming is the share of promoting violence (18.0%), clearly indicating that many comments are toxic.

In an application more similar to our own, Aggarwal et al. (2020) used Tweets from popular media outlets and journalists in India to investigate how biased opinions are channeled and propagated via social media into a network of people following/consuming it.

On November 8, 2016, the government of India announced the demonetization of Rs. 500 and Rs. 1000 currency notes to curb the menace of black money. This provoked wide-ranging reactions from people, with some appreciating the move and some criticizing it. This study uses the Twitter API to fetch real time tweets from the verified handles of 10 media outlets and 10 journalists. A total of 4,475 tweets are collected for the subject of "demonetization." Some pre-processing was done in order to make the tweets suitable for further computation: URLs were replaced with a tag "URL", targets were replaced by the username, punctuation marks occurring at the start or end of a word were separated from the word to enable easier tokenization.

Sentiment scores were determined by analyzing the tweets collected using the Valence Aware Dictionary and sEntiment Reasoner (VADER). NLTK-VADER is a popular tool for sentiment analysis of social media texts. The sentiment scores obtained by VADER are used for determining the sentiment of text. VADER provides a sentiment score (PS) ranging from -1 (strongly negative) to +1 (strongly positive). This range is split into 3 regions, namely positive, negative and neutral. PS values ranging from -1 to -0.33 are classified as negative PS values while PS values ranging from 0.33 to 1 are classified as positive PS values. PS values lying between -0.33 and 0.33 are referred to as neutral PS values. In our research, the VADER scoring mechanism documentation (Hutto Gilbert, 2014) denotes sentiment cut offs at ±0.05 in order to flavor them positive or negative, rather than ±0.33, but Aggarwal et al.'s decision to use a broader definition of neutral is a likely a reflection of their desire to find particularly biased original source content.

The polar tweets form the basis for analyzing conditioning by media outlets. For each polar tweet, a list of user-ids of the users who 'react' to the given tweet is also extracted. The behavior of 'reactors' to polar tweets is observed to study the conditioning of opinions by the media outlets. Each user has prior opinions on issues. The prior polarity (p) of a person is calculated by the exponential moving average of the polarity score of its previous tweets. However, a polar tweet/news report by a media outlet can affect the user's opinion on the issue. The polarity in the opinion of an user might show a massive swing (reversing one's opinion) or a minor one (a small strengthening of prior opinion). The post-event polarity (p') is obtained by taking the arithmetic mean of the polarity scores of its tweets post the event for a period of 24 hours. An event E is said to have conditioned a consumer C if the value of p and p' are significantly different. The two opinions p and p' are assumed to be significantly different if the difference of p and p' is 20% more than the standard deviation in the PSs of C prior to event E.

A large fraction of tweets by media outlets/journalists are observed to be polar or subjective in nature. More than 40% of the tweets analyzed were found out to be polar. Also, a fraction as high as 41% of reactors were conditioned through such 'biased' polar tweets. This shows that an alarming number of people get conditioned by polar, subjective and biased news every day, something which goes past the naked eye of humans around the world.

The Aggarwal et al. paper provides a useful example of using VADER to assign tweets from popular media outlets and served as a guidance in constructing the labels for our dataset. It also provides evidence that there is a short term impact of media bias.

We moved forward with the understanding that our concept of using polarization as a target and trying to learn specific elements of news media speech which best predict it is somewhat novel, though strongly connected to previously researched areas.

3 Data

For the purposes of tracing media speech and subsequent comments, we pulled two Twitter datasets and a Kaggle dataset of archived articles scraped from the New York Times website. We grouped comments by original post and used a measure of the polarity across the body of each original tweet's comments to create a label of whether the original tweet created polarization within its comments.

3.1 News Media Datasets

Our project used three distinct data collection tasks to build a body of new media posts and replies to work from: two Twitter datasets and a Kaggle dataset. The Twitter data was downloaded using a python wrapper designed to harvest data from Twitter's REST API (JustAnotherArchivist, 2018). The Kaggle dataset was downloaded from a closed contest related to predicting content upvotes (Kesarwani, 2018).

For all three datasets, we cleaned the text by removing urls, social media tags, other special characters like emojis and tokenized it. We removed from our dataset posts with fewer than 5 responses, as we felt polarity might not be meaningfully measurable for small reply counts.

The Kaggle dataset contained 4,260 original posts matched with 884,890 comments from January to April of 2018. These are exclusively from the New York Times website. By our final polarity measure, this data is skewed towards being polarized, with about 85% of the conversations labeled as polarized.

The first set of tweets contained 42,516 original posts from three news outlets (The New York Times, Reuters and Fox News) along with the matched 3,854,911 subsequent user comments occurring between January and April of 2021. We selected these three news outlets because they are each well known with active Twitter followings and lie across the political spectrum. AllSides classifies Reuters as centrist, the New York Times as left-leaning and Fox News as right-leaning (AllSides, 2019). This data has 44% threads labeled as polarized by our final polarity measure.

Concerned that our data includes a broad array of news topics and feature importance may simply teach us which topics are most polarizing, rather learning anything about the approach to the coverage, we pulled an additional 21,053 tweets and 3,677,273 replies from 7 news sources specifically on keywords coming up in our initial models as polarizing. The key words used were: “Floyd,” “Biden,” “Trump,” “Chauvin,” “black,” “asian,” “arrested,” “killed,” and “attacked.” Tweets and replies were gathered from the New York Times , Fox News, Reuters, BBC, Daily Wire, CNN and Washington Post between October 2020 and April 2021. As expected, the second batch of tweets was more controversial, with 63% labeled as polarized by our final measure.

Since our datasets were not balanced between classification categories, we also experienced early difficulties in model performance around this and attempted various strategies to enforce class balance including using loss weights and dropping members of the overrepresented class.

The pulling of Twitter data was an extremely time consuming task. Since, in order to get our measures of polarity, we needed to retrieve approximately 100 replies for each original tweet the amount of data available to our model was significantly less than the number of tweets we retrieved. Our best estimate is that it took approximately 12 hours to pull each of the two batches of tweets.

3.2 Topic Modelling

Our initial concern related to models learning controversial topics versus controversial ways of presenting the news led us to add a topic component to our datasets. We created bigrams from the text data, created a bag of words (BoW) and then used Latent Dirichlet Allocation (LDA) to obtain the main topics of the set of tweets.

With LDA, for each tweet a subset of relevant words suggestive of a theme is identified by the learning algorithm. It then groups those with common themes together into a predetermined number of topics. Since the topic count is unknown at the outset, we tested topic modeling on a range of topic groupings from 8 to 20. The set of topics was relatively coherent when grouped at the 11 topic level. These categories were: ’Business’, ’Pandemic’, ’Protests’, ’Politics’, ’COVID’,’Violence’, ’Media’, ’Racial Violence’, ’International’, ’Economy’ and ’Other.’

Our original concept was to filter or control by topic to attempt to allow our model to learn about the way in which a topic is presented, rather than identifying the topics themselves as the most important determinant of polarity. Unfortunately our dataset was not large enough to pass sufficient training data into our models filtered down to topic level. To keep this general approach, but not restrict our dataset so drastically, we completed the second tweet data pull around highly controversial key words.

3.3 Creation of Polarization Metrics

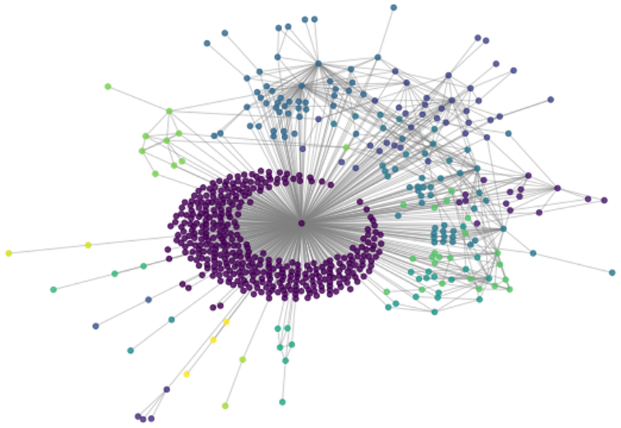
We attempted three different polarization metrics based on sentiment scores (as in Aggarwal et al., 2020) and one based on controversy (as in O. de Zarate et al., 2020). Our final dataset used a count based method related to sentiment score similar to that of Aggarwal and co-authors.

Since we could not find a dataset pre-labeled for comment controversy, our measure is, by necessity, constructed. Our initial approach was to attempt to create both a graph-based controversy score and a measure of polarity in sentiment to compare. If our models were able to predict similar results over both a semantic measure of polarity and a strength-of-sentiment based one, we could be more confident the patterns learned were truly predictive of an underlying content pattern.

For controversy scores, we attempted to use the same graph building method as O. de Zarate et al. (2020). However, due to the lack of enough time and data (enough responses per tweet), we weren’t able to build a measure of controversy from the communities detected by the Louvain algorithm in each tweet conversation. We believe this is still a great approach to measure polarization as it computes the embedding difference (context) between the center nodes of the two main communities in the tweet. For example, taking a tweet from Fox News Twitter stated: “New BLM demand seeks to permanently ban Trump from ’all digital media platforms”. We applied the Louvain algorithm and detected about 15 different communities, represented in Figure 1. Nodes are the Twitter users: FoxNews, user2, user4, etc. And the edges represent the messages. The colors represent the communities to which they belong. Following O. de Zarate et al. (2020), to compute a controversy score, we only take the two biggest communities, meaning the purple and blue communities. From those tweets, we can use pre-trained embeddings models (FastText, Bert) and retrain them to predict the communities in which each tweet belongs. Once the model is trained, we can estimate the controversy score by taking a weighted average of the euclidean distance of the users’ tweets embeddings with respect to the centroid for each community group. Intuitively, it represents how much the clusters are separated.

As previously mentioned, we lack enough data to follow this approach. O. de Zarate et al. (2020) computed the controversy scores with more than 100,000 tweet responses for each topic. In our methodology, we are not dividing by topic but by Twitter conversation from news outlets, and most of these conversations only have, on average, less than 100 replies. Note that the Twitter conversation shown in Figure 1 has about 1000 replies. Due to this limitation, we decided to abandon this methodology to measure polarization.

[1]



(a) New BLM demand seeks to permanently ban Trump from ’all digital media platforms

Figure 1: Community Identification of a Tweet Conversation from Fox News

For sentiment scores, we utilized the VADER scoring tool from Aggarwal et al. (2020). First we applied the scoring mechanism to each reply. VADER is a dictionary and context rule-based tokenizer which receives a string, assigns sentiment score between -1 and 1 to each word based on a vocabulary lookup (Hutto Gilbert, 2014). It then adjusts the sentiment based on the immediate word context, so is best for short text and short range dependencies. Sentiments for the full sequence are divided into positive, negative and neutral components and a weighted sum of these are returned as the overall sentiment of the string. The VADER calculation categorizes anything between -0.05 and 0.05 as sentiment neutral. Values below this are negative with increasing intensity and above this are positive with increasing intensity. VADER was trained on social media content, so despite its limitations it is well suited for a Twitter dataset where text lengths are tightly constrained to a 280 character limit.

[1] Nodes are the Twitter users: FoxNews, user2, user4, etc. And the edges represent the replies between users. The colors represent the communities to which they belong according to the Louvain algorithm. For the graph, we used a ForceAtlas2 layout.

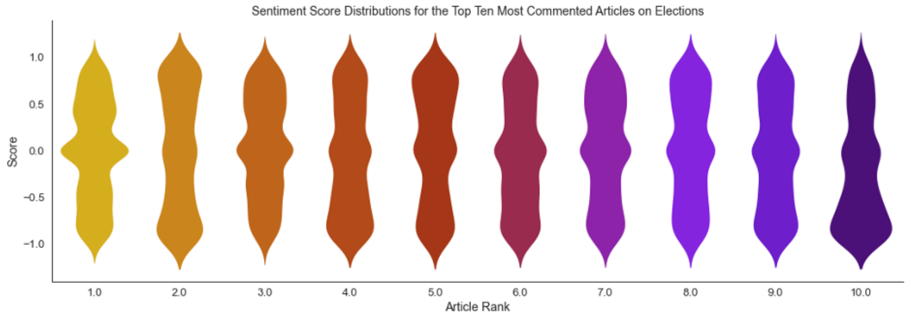
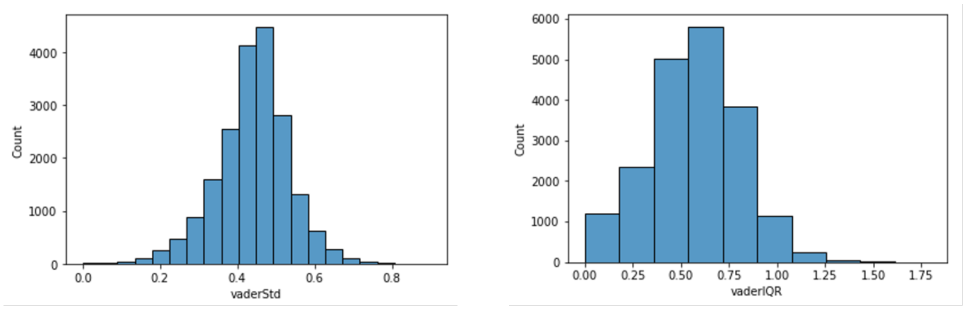


Figure 2: Example of VADER sentiment score distributions for the 10 most frequently commented NYT articles about elections

We initially attempted to model distributions of sentiment scores for each original tweet and use measures of central tendency to differentiate between threads that were highly polarizing or not. We established categorical cutoffs for standard deviation related to our expectation of a random normal distribution as completely unpolarized and a bimodal distribution with modes at -1 and 1 as completely polarized. With this conceptual model, we binned sentiment score distributions into five evenly spaced categories ranging from no polarization to extreme polarization between standard deviations equal to or less than 0.33 to those greater than or equal to 0.67. The number of tweets in the most extreme categories, especially the low end, were unacceptably small, so we combined the lower and higher categories together, creating three bins: low, medium and high.



(a) Standard deviation

(b) Interquartile range

Figure 3: Distribution of Twitter dataset batch 1

We also used a similar method to categorize tweet sentiment polarity based on interquartile ranges. The interquartile range measure was in agreement with the standard deviation measure in most cases, only about 3% of original tweets were categorized differently using interquartile range versus standard deviation to calculate a measure of variability. Since there was so little difference between the two methods, we moved forward with testing using standard deviation.

Finally, as our tests continued, we noticed classification patterns leading us to believe the models were having difficulty distinguishing between tweets on the border between categories. We tested this by removing a small band of standard deviations from the center of our dataset and classifying original tweets in a binary manner: either high or low polarity. This resulted in much improved model performance, however is conceptually dubious because the removed swath of data was chosen somewhat arbitrarily. To better codify our binary classification problem we tried a new, count based, measure of polarity. Using the definitions of positive, negative and neutral sentiment from VADER (Hutto Gilbert, 2014), we counted the number of generally positive, negative and neutral tweets from the replies for each original post. If a post's neutral comments were outnumbered by both its positive comments and by its negative comments, we determined the post to be polarizing. This is similar to the Aggarwal et al. method of classifying news content, though it uses the original Hutto and Gilbert cutoff scores.

To better codify our binary classification problem we tried a new, count based, measure of polarity. Using the definitions of positive, negative and neutral sentiment from VADER (Hutto Gilbert, 2014), we counted the number of generally positive, negative and neutral tweets from the replies for each original post. If a post's neutral comments were outnumbered by both its positive comments and by its negative comments, we determined the post to be polarizing. This is similar to the Aggarwal et al. method of classifying news content, though it uses the original Hutto and Gilbert cutoff scores.

4 Methodology

Our approach to predict polarity is using neural networks models, particularly, a Recurrent Neural Network (RNN) and a Convolutional Neural Network (CNN). Both are state of the art models for text classification. Notably, RNNs can capture long-term relationships from arbitrary sized texts. CNN models can identify local aspects that are the most informative for the prediction task at hand, in our case, polarization. Lastly, we also used a traditional logistic regression model as a baseline model to assess the performance of the deep neural networks.

4.1 RNN as an Acceptor of Polarization

For our first neural network model, we created an RNN encoder feeding into a linear layer. We batched the text data in a method similar to that used by Trevett (2021) using the legacy bucket iterator from PyTorch and Backward Propagation Through Time (BPTT) as used in Chaudhary (2021). BPTT feeds sequences into the recurrent layer in ordered pairs and chains the error calculation across time in a way that is more computationally efficient than other common models for gradient descent over recurrent networks.

In this analysis, we use the encoded hidden state of the RNN at the end of the final output vector. The RNN is trained as an acceptor, meaning that after observing the final state, the model passes this state though one or more fully connected linear layers to decide on an outcome. For example, the RNN reads a tweet from a news outlet and based on the final state from the recurrent layers, decides the probability it will lead to a high or low polarization of twitter replies. The final output function defined as yn = O(staten) is fed into a fully connected linear layer to produce a vector of probabilities referring to the classes: High Polarization or Low Polarization. Then, in the backward phase, the error gradients are backpropagated through the rest of the sequence (the twitter text). This configuration is based on the original work of Hochreiter Schmidhuber in the use of LSTM (1997) and a subsequent fake news detection model trained by Patupat Cheng (2020). The following diagram shows the graphical representation of the model.

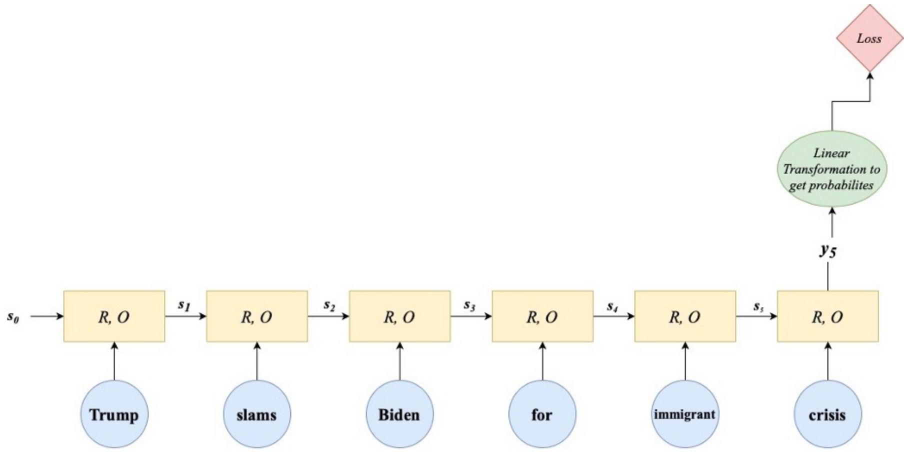


Figure 4: Acceptor RNN Training Graph

4.2 Bi-directional RNN as an Encoder of Polarized/Non-Polarized Dependencies

Another useful type of RNN for our prediction task is the bidirectional-RNN (biRNN), which is able to capture dependencies from both directions. Similarly to the RNN, the biRNN allows the encoding to arbitrarily look back to past information, but with the addition of also looking arbitrary to the future information within the sequence. The biRNN architecture works by having two separate states, the forward states (s_{if}) and backward states (s_ib) with two different RNNs. The first RNN is fed with the input sequence as normal ($x_{1:n}$), and the second RNN is fed with the input sequence reversed ($x_{n:1}$). For our implementation, similarly to the Acceptor RNN, we care about the final state (see figure 5). In this case we have two final states, forward y_{nf} and backward y_{nb}. (y_{5f} and y_{5b} in figure 5 example), and we concatenate them into one final output y_n. The y_n is fed into a fully connected linear layer to produce a vector of probabilities referring to the classes: High Polarization or Low Polarization. Finally, in the backward phase, the error gradients at position n will flow through the two RNNs. With this model, we are hopeful to find dependencies from both directions for the final prediction output.

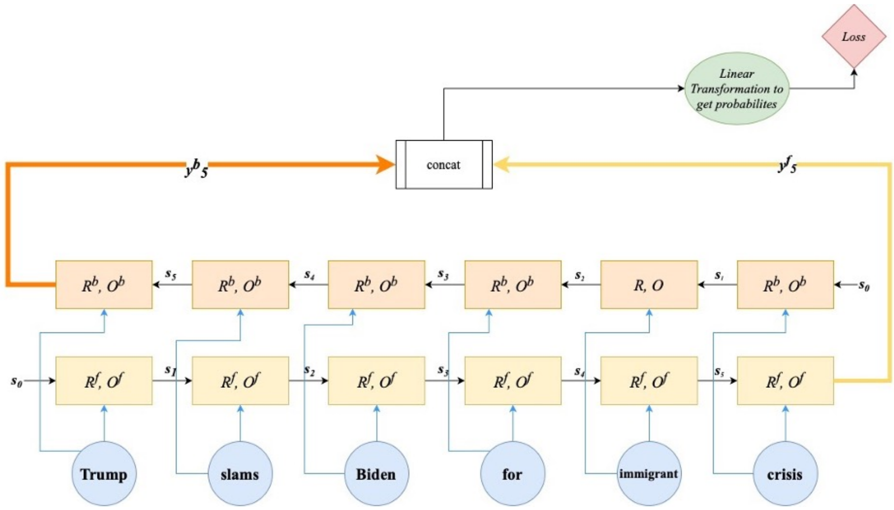


Figure 5: CNN for Sentence Classification

4.3 CNN for Sentence Classification

Convolutional Neural Networks (ConvNets or CNNs) are state of the art models famous for image recognition given their capability to identify relevant features from large data structures. For text classification, CNNs are very useful in identifying relevant ngrams "without the need to pre-specify an embedding vector for each possible ngram" (Golberg, 2017). The basic architecture of a CNN is to apply a nonlinear function (aka filter) over each instantiation of a k-word sliding window over a sentence. For example, suppose we are classifying a tweet with the following sentence: "Trump slams Biden for immigrant crisis". Just like with images, we can represent the text in 2 dimensions using word embeddings, in the following example, text is represented as a 6x6 matrix (6 words x 6 dimensional embedding). In our implementation, we use trained word embeddings of 400 dimensions per word.

$$x_{emb} \in R^{6x6}$$

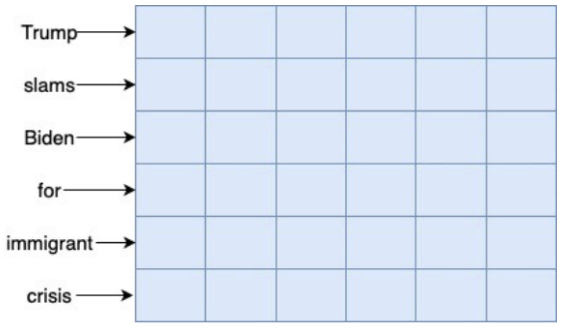


Figure 6: Embedding Diagram as an Image

Then, if we want a filter that covers two words (bi-grams) at a time, then the filter is 2x6 embedding dimensions. This filter will move down the image to cover all bi-grams. For each bigram a weight is associated with it. Then the next step is to use pooling to take the maximum value over a dimension (purple square). The idea is that the maximum value is the most important feature in determining the polarization of this tweet, which corresponds to the most-important n-gram within the tweet. In this "example case", the most important bi-gram is "Trump slams".

$$x_{conv_i} = Conv(x_{emb}) \in R^{1\,filter}$$

$$x_{pool_i} = MaxPool(ReLU(x_{conv_i}))$$

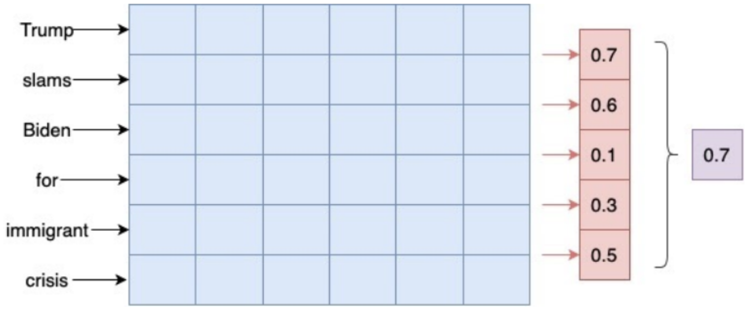


Figure 7: 1D Convolution with one bi-gram filter

In this example, we only used one filter. In our implementation (see table 1), we used about 100 filters with three different sizes: 3, 4, 5 (tri-gram, four-gram, five-gram). After the forward pass, the gradients that are propagated back, are used to tune the parameters in the filter function such that the function highlights the important aspects of the data. Intuitively, CNN is a feature-extracting architecture. With this, our objective is to extract the most important n-word groups (n-grams) that lead to the polarization in the tweet replies.

4.4 Logistic regression

As a baseline, we used a logistic regression model. With this model, we specified three different ways to transform the text into vectors with the following arquitectures: bag of words, bag of n-grams and term frequency-inverse document frequency (Tf-Idf). We used a grid of parameter settings to optimize the hyperparameters of the model. We tested a technique called Synthetic Minority Oversampling

Technique (SMOTE). This technique is used when datasets are unbalanced, it generates synthetic samples for the minority class. SMOTE didn't increase the accuracy of the logistic regression model. Based on the best validation accuracy, the logistic regression model with the Tf-Idf performed best.

4.5 Model Interpretability

Since our fundamental question involves the identification of tweet content for prediction, we used Integrated Gradient to understand the feature importance in our neural networks. Integrated Gradient is an interpretability algorithm used to understand how neural networks work. This algorithm, proposed by Sundararajan et al. (2017), attributes the prediction of a neural network to its input features. The gradient signals the neural network to increase or decrease a certain weight in the network during backpropagation and it relies on the input features to do so. Thus, the gradient associated with each input feature with respect to the output can be used to get a sense of the importance of a feature.

The integrated gradient along the i^{th} dimension for an input x and a baseline x' is defined as follows:

$$IntegratedGrads_i(x) = (x_i - x'_i)\int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} d\alpha$$

For text models, the baseline can be the zero embedding vector.

Integrated gradients satisfies two fundamental axioms that other attribution methods don't:

- Sensitivity - An attribution method satisfies Sensitivity if for every input and baseline that differ in one feature but have different predictions then the differing feature should be given a non-zero attribution.
- Implementation Invariance - An attribution method satisfies Implementation Invariance, if the attributions are always identical for two functionally equivalent networks. If an attribution method fails to satisfy Implementation Invariance, the attributions are potentially sensitive to unimportant aspects of the models.

We applied this algorithm to our CNN and RNN. In figures 7 and 8 below, a set of tweets was randomly selected to illustrate how Integrated Gradient attribution modeling works over the context of a sequence.

5 Evaluation and results

Surprisingly, the best model to predict polarization was the logistic regression model using Tf-Idf to create the word vectors. The accuracy and f-1 score was about 65% and 71% respectively.

With the CNN, we ran two configurations, one with 100 filters for three different sizes: tri-gram, four-gram and five-gram, and the second one with 200 filters for the tri-gram filter and 100 filters for the four-gram and five-gram. In this sense, the models differ only in the number of filters, for the first one, the model assumes there are 300 different n-grams that are important, for the second one, the model assumes there are 400 different n-grams that are important for the prediction. Increasing the number of filters slightly increases accuracy and f-1 score to an accuracy and f-1 score of 62% and 63%.

The bi-RNN model is marginally better than the RNN model, with a 61% and 60% accuracy. The F-1 scores for both the RNN was 0.60 and 0.66 for the biRNN. Numerous adjustments were attempted on parameters such as learning rate, the number of linear layers and recurrent layers, the length of the BPTT chain, vocabulary size, drop out rate and layer sizes. We concluded that while adjustments to such parameters had marginal impacts on model performance, the most noticeable improvements came from two data related changes. First, more data greatly improved accuracy rates and reduced validation loss. Second, the model struggles to classify tweets that are neither strongly polarized nor particularly unimodal. When we reduced our problem from a multi-class to a binary classification task this improved model performance. We also attempted and abandoned the somewhat arbitrary strategy of dropping tweets with the middle-most polarization scores, which improved performance greatly, but was conceptually questionable. In sum, the model performance across the neural networks varied significantly with the way we measured polarity and the size of the dataset.

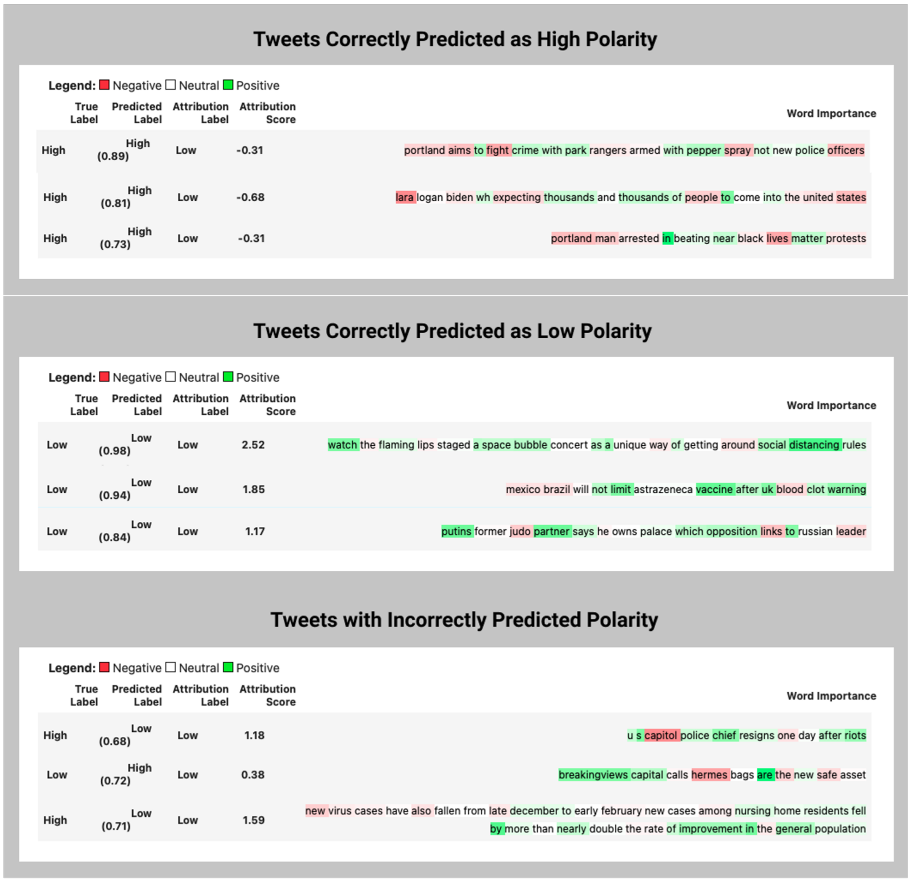
Table 1: Model Performance on Test Data

Model	Parameters	Test Accuracy	Test F-1 Score
Logistic Regression	C = 0.001 Feature extraction: Tf-Idf	65%	71%
RNN	LSTM layers:2 Drop out: 0.5 Learn Rate = 0.001 BPTT length = 64 Loss = Cross Entropy Optimizer = Adam	60%	60%
bi-RNN	LSTM layers: 2 Drop out: 0.5 Learn Rate = 0.001 BPTT length = 64 Loss = Cross Entropy Optimizer = Adam	61%	66%
CNN	Filter sizes: 3,4, 5 Number of Filters: 100 Drop out: 0.5 Learn Rate = 0.001 Loss = Cross Entropy Optimizer = Adam	61%	58%
CNN	Filter sizes: 3,4, 5 Number of Filters: 200, 100, 100 Drop out: 0.5 Learn Rate = 0.001 Loss = Cross Entropy Optimizer = Adam	62%	63%

5.1 Feature importance

To answer our fundamental question of what makes a news media post polarizing, we analyzed the textual feature importance from each of our models. For Logistic Regression we extracted the largest coefficients of the model and for the neural networks we used Integrated Gradient to estimate approximate word weights from model weights.

True to our original concern, all models did a better job of locating polarizing topics within the tweet corpus. Table 2 below shows the most significantly predictive terms from each model. Common shared terms like “Trump,” “Biden” and “president” seem directly tied to controversy surrounding the transition of presidential power in the United States during the period of time the tweets span. There are other surprises, however, that may better address our original research question. For example, multiple models identified tweets labeled as opinion columns or analysis stories tended to predict polarization. Follow up stories using words like “update” or “icymi” (which stands for “in case you missed it”) may also be more polarizing. Another odd discovery is the appearance of the word “here’s” which tends to start news media tweets with a certain tone, such as “Here’s why. . . ” or “Here’s how. . . ” This could provide some evidence that this specific tweet tone is somewhat polarizing.



Tweets Correctly Predicted as High Polarity

Legend: 🟥 Negative □ Neutral 🟩 Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
High	High (0.89)	Low	-0.31	portland aims to fight crime with park rangers armed with pepper spray not new police officers
High	High (0.81)	Low	-0.68	lara logan biden wh expecting thousands and thousands of people to come into the united states
High	High (0.73)	Low	-0.31	portland man arrested in beating near black lives matter protests

Tweets Correctly Predicted as Low Polarity

Legend: 🟥 Negative □ Neutral 🟩 Positive

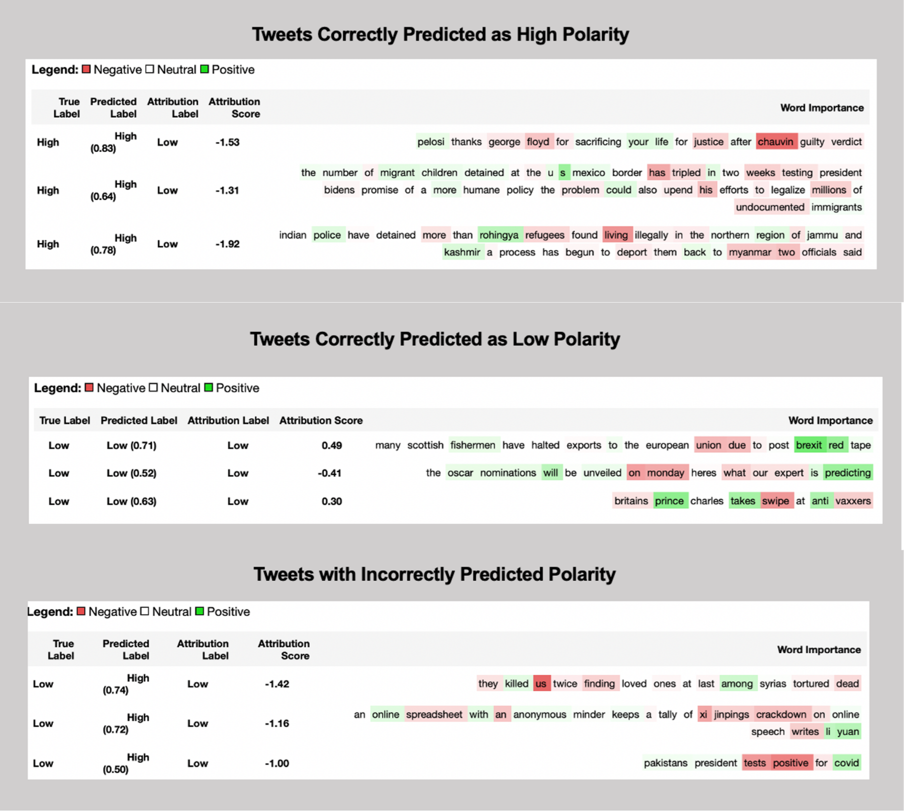
True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
Low	Low (0.98)	Low	2.52	watch the flaming lips staged a space bubble concert as a unique way of getting around social distancing rules
Low	Low (0.94)	Low	1.85	mexico brazil will not limit astrazeneca vaccine after uk blood clot warning
Low	Low (0.84)	Low	1.17	putins former judo partner says he owns palace which opposition links to russian leader

Tweets with Incorrectly Predicted Polarity

Legend: 🟥 Negative □ Neutral 🟩 Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
High	Low (0.68)	Low	1.18	u s capitol police chief resigns one day after riots
Low	High (0.72)	Low	0.38	breakingviews capital calls hermes bags are the new safe asset
High	Low (0.71)	Low	1.59	new virus cases have also fallen from late december to early february new cases among nursing home residents fell by more than nearly double the rate of improvement in the general population

Figure 8: Integrated Gradient Attribution Output Examples from BiRNN Mode



Tweets Correctly Predicted as High Polarity

Legend: 🟥 Negative □ Neutral 🟩 Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
High	High (0.83)	Low	-1.53	pelosi thanks george floyd for sacrificing your life for justice after chauvin guilty verdict
High	High (0.64)	Low	-1.31	the number of migrant children detained at the u s mexico border has tripled in two weeks testing president bidens promise of a more humane policy the problem could also upend his efforts to legalize millions of undocumented immigrants
High	High (0.78)	Low	-1.92	indian police have detained more than rohingya refugees found living illegally in the northern region of jammu and kashmir a process has begun to deport them back to myanmar two officials said

Tweets Correctly Predicted as Low Polarity

Legend: 🟥 Negative □ Neutral 🟩 Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
Low	Low (0.71)	Low	0.49	many scottish fishermen have halted exports to the european union due to post brexit red tape
Low	Low (0.52)	Low	-0.41	the oscar nominations will be unveiled on monday heres what our expert is predicting
Low	Low (0.63)	Low	0.30	britains prince charles takes swipe at anti vaxxers

Tweets with Incorrectly Predicted Polarity

Legend: 🟥 Negative □ Neutral 🟩 Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
Low	High (0.74)	Low	-1.42	they killed us twice finding loved ones at last among syrias tortured dead
Low	High (0.72)	Low	-1.16	an online spreadsheet with an anonymous minder keeps a tally of xi jinpings crackdown on online speech writes li yuan
Low	High (0.50)	Low	-1.00	pakistans president tests positive for covid

Figure 9: Integrated Gradient Attribution Output Examples from CNN Mode

Table 2: Integrated Gradient Estimation of Feature Importance*

Logistic Regression	RNN	biRNN	CNN
trump	**house**	**Trump**	**trump**
trumps	years	**President**	**president**
opinion	**analysis**	**Trumps**	**trumps**
biden	**capitol**	**Biden**	**biden**
president	vaccine	**coronavirus**	after
us	**writes**	**China**	writes
meghan	health	about	**police**
arrested	**bidens**	**House**	opinion
writes	report	election	military
amid	updates	White	analysis
analysis	**chinas**	against	**bidens**
impeachment	**myanmar**	**Capitol**	chauvin
democrats	**presidential**	**president**	**capitol**
icymi	death	**Bidens**	trial
book	campaign	people	democrats
republicans	heres	**Analysis**	officer
nuclear	**opinion**	**trial**	legal
bidens	being	**Myanmar**	americans
pardons	second	**impeachment**	shooting
challenges	states	American	years

*Common words between models in bold

Our models, though interesting to review, are not perfect predictors of polarity by any stretch of the imagination. For instance, the fake news classification method on which our LSTM model is based achieves a 77% accuracy rate on it's fake versus real news classification task (Patupat Cheng, 2020). Clearly the connection between original post and reply polarity, as we have defined it, is less easily predictable.

6 Limitations and Challenges

The main challenge that we encountered is related to data collection. The Twitter API has low rate limits which restricts our ability to download all the comments for a set of tweets. To overcome some of these limitations, we downloaded the data using a python wrapper designed to harvest data from Twitter's REST API (JustAnotherArchivist, 2018). Downloading tweets and comments using this method took a long time. This challenge was accentuated by the fact that, in order to construct the labels, we required around 100 rows of data/comments for each original tweet observation. Consequently, our initial dataset was large but it was reduced greatly after calculating the labels. We were able to add some additional tweets to the dataset after running the models for the first time and we saw improvements in their performance. Thus, we consider that our models could be improved by expanding the dataset.

The second challenge is related to model interpretability. Our hypothesis was that the tone or biased language of a tweet is predictive of the level of polarization in the responses. But it is difficult to separate the predictive power of topics (a tweet that talks about Trump is more polarizing than a tweet that talks about art) from the predictive power of words that indicate bias. However, our feature importance analysis suggests that the models are using both to make predictions. For example, the models predicted higher polarization for tweets containing words like "against" and "analysis".

7 Conclusions and Future Work

In this study, we used Twitter data from news media accounts to test if the text extracted from their tweets is predictive of polarization in comments. We used the comments for each tweet to build the labels (polarized or not polarized). To build the labels, we assigned sentiment scores to the comments using Vader and we built a measure of dispersion in the sentiments for each original tweet. The features were extracted from the text of the original tweets. We trained three different supervised ML models with multiple variations: Logistic Regression, Recurrent Neural Network and Convolutional Neural Network. Our best model was the Logistic Regression with which we achieved an accuracy of 65%. We used Integrated Gradients to get a sense of how our neural networks were working. We found that the neural networks predicted high levels of polarization for tweets that talked about topics related to politics as well as for tweets that used words that are related to bias like "opinion" and "against".

In the future, we would like to re-train our models using more data to improve performance. We would also like to restrict the data to a specific topic to see if we can isolate the predictive power of bias in the way a tweet is written. To be able to restrict tweets to a certain topic we would need a much larger dataset. Given the time necessary to harvest Tweets, we were unable to collect enough data within the scope of this project to track a single news item or sufficiently narrow topic to accomplish this. In the same vein, the lack of data limited our ability to measure polarity using a graph-based model following O. de Zarate et al. (2020). In the future, we would like to add this measure of polarity to assess how the dataset changes and whether or not models are sensitive to these changes.

Another approach we were unable to try within project time constraints would be to mask proper nouns, like people and places, which tend to be polarizing on the topical level, in order to predict over reporter tone in things like verb and adjective use. This could allow more subtle author voice patterns to surface and would be a good future extension of this work.

We feel that research on creating a more civil and less polarized space for discourse on social media is a worthy pursuit. Despite our research limitations, we still feel this is a promising avenue for informing public debate on media bias.

Division of Labor

	Jesica	Kelsey	Stephanie
Experimental Design	CNN Model selection, Controversy score conceptualization, Sentiment score conceptualization (secondary)	Sentiment score conceptualization (primary)	Original proposal concept.
Preprocessing Code	Controversy score building pipeline, Topic modeling pipeline, Scraping of the second batch of Tweets.	Tweet scraping pipeline, Sentiment score building pipeline, Scraping of the first batch of Tweets	Text preprocessing pipeline, Kaggle dataset.
Model-related Code	CNN models	RNN (LSTM) models	Regression models, Integrated Gradients for feature importance for all models.
Report	Methodology & Results sections	Related Work & Datasets sections	Abstract, Limitations and Challenges & Conclusions and Future Work sections & LaTeX document.
Learning / Growth Areas	Different architectures for NN models, and the intuition behind them, Model Building with PyTorch, particularly, I learnt how to build a CNN mode, Data extraction with Twitter API. The importance of having accurate labels.	Tweet extraction, Target feature engineering, Use of language models for classification, VADER scoring tool for sentiment scoring, The importance of balanced datasets to model validity.	Pytorch captum library (for model interpretability), New concepts: Integrated Gradients and Occlusion, Feature extraction using scikit-learn, Theoretical differences and differences in implementation of RNNs and CNNs.

References

[1] Aggarwal, S. Sinha, T., Kukreti, Y., Shikhar, S. (2020). Media bias detection and bias short term impact assessment. Array, Volume 6, 100025, ISSN 2590-0056, https://doi.org/10.1016/j.array.2020.100025.

[2] AllSides. (2019, February 21). Media Bias Chart | AllSides. AllSides. https://www.allsides.com/media-bias/media-bias-chart

[3] Chaudhary, A. (2021) An RNN Transducer-based Language Model. [Course Assignment]. Computational Analytics for Public Policy, University of Chicago, Chicago, IL.

[4] Ferreira, R., Freitas, F., Cabral, L. d. S., Lins, R. D., Lima, R., França, G., Favaro, L., Simske, S. J. (2014). A Context Based Text Summarization System. 11th IAPR International Workshop on Document Analysis Systems, Tours, France; IEEE. https://ieeexplore.ieee.org/document/6830971

[5] Guerra, P., Meira Jr., W., Cardie, C., Kleinberg, R. (2013, June 28). A Measure of Polarization on Social Media Networks Based on Community Boundaries. Proceedings of the International AAAI Conference on Web and Social Media; AAAI Publications. https://ojs.aaai.org/index.php/ICWSM/article/view/14421

[6] Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 8, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[7] Hutto, C.J. Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[8] Jachim, P., Sharevski, F., Pieroni, E. (2021, April 1). "TL;DR: Out-of-Context Adversarial Text Summarization and Hashtag Recommendation". ArXiv.Org. https://arxiv.org/abs/2104.00782

[9] JustAnotherArchivist. (2018). GitHub - JustAnotherArchivist/snscrape: A social networking service scraper in Python. GitHub. https://github.com/JustAnotherArchivist/snscrape

[10] Kesarwani, A. (2018, May 2). New York Times Comments. Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/aashita/nyt-comments

[11] Lee, N., Bang, Y., Madotto, A., Fung, P. (2021, April 1). Mitigating Media Bias through Neutral Article Generation. ArXiv.Org. https://arxiv.org/abs/2104.00336

[12] Ortiz de Zarate, J., Di Giovanni, M., Zindel E., and Brambilla, M. (2020) Measuring Controversy in Social Networks Through NLP. C. Boucher and S. V. Thankachan (Eds.): SPIRE 2020, LNCS 12303, pp. 194–209, 2020. https://doi.org/10.1007/978-3-030-59212-7$_1$4

[13] Patupat, A. J. L., Cheng, I.T. (2020). Performance Analysis of Different Word Embeddings and Transformers on Fake News Detection. Self Published | GitHub. https://itsuncheng.github.io/files/comp4211$_{p}aper.pdf$

[14] Trevett, B. (2021, March 21). pytorch-sentiment-analysis/1 - Simple Sentiment Analysis.ipynb at master · bentrevett/pytorch-sentiment-analysis · GitHub. GitHub. https://github.com/bentrevett/pytorch-sentiment-analysis/blob/master/1%20-%20Simple%20Sentiment%20Analysis.ipynb

[15] Salminen, J., Almerekhi, H., Milenković, M., Jung, S. G., An, J., Kwak, H., Jansen, B. J. (2018). Anatomy of online hate: Developing a taxonomy and machine learning models for identifying and classifying hate in online news media. In 12th International AAAI Conference on Web and Social Media, ICWSM 2018 (pp. 330-339). (12th International AAAI Conference on Web and Social Media, ICWSM 2018). AAAI press.

[16] Sundararajan, Mukund Taly, Ankur Yan, Qiqi. (2017). Axiomatic Attribution for Deep Networks.

[17] Wang, Xingyou and Jiang, Weijie and Luo, Zhiyong (2016). Combination of Convolutional and Recurrent Neural Network for Sentiment Analysis of Short Texts. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers (pp. 2428–2437). https://www.aclweb.org/anthology/C16-1229