Voter Sentiment through Political Stance Prediction and Topic Modeling using 2020 U.S. Election Twitter Data

ECON1680 Project 2 Draft 1

Author:

Lisa Ruixuan Li

Supervisor:

Prof. Amy Handlan

April 2023

Contents

1 Introduction

Twitter has become one of the most prominent platforms when it comes to voicing political opinions. Especially in periods of election, researchers can use Twitter posts to survey users' political leanings to better understand political polarization and voter sentiment over time and across topics. Specifically, we can detect political stances and label tweets as Democrat or Republican-leaning, which can help us understand how voters supporting different candidates behave differently on the internet. Further employing methods of deep learning, we can classify tweets relating to various categories of political discussion. This labeling method, called "topic modeling", allows us to cluster tweets into topics like healthcare, immigration, gun violence, foreign policy, just to name of few political topics of controversy.

In this research, I use machine learning and text analysis methods to study voter sentiments and political leanings from a sample of tweets from Oct 15, 2020 -Nov 8, 2020, a roughly two-week time period just before the 2020 U.S. Election date Nov 3, plus another week after the results of the election are announced. I will be mainly doing three things in this research: 1) Produce a word dictionary that best help classify tweets to Trump vs. Biden leaning; 2) Model tweets into different categories pertaining to different political topics; 3) Analyze sentiments and use of hate speech and other polar languages in different political stances and political topics.

These sets of sentiment analysis techniques greatly contribute to studying the economics of mass media and political economy, and produce a reusable dictionary of keywords that can help classify and label future tweets related to elections in the U.S.

2 Data

2.1 Dataset

The main dataset I'm using comes from Kaggle's *Donald Trump vs. Joe Biden 2020 dataset* (Hui 2021). It contains Twitter data over the period Oct 15 – Nov 8 in year 2020, grouped into two subsets according to the candidate names tagged in the post, 'hashtag Trump' and 'hashtag biden'. To collect the data, a list of related hashtags involving Donald Trump and Joe Biden is fed into the TwitterAPI query system, spitting out tweets highly relevant to the 2020 U.S. Election. This dataset contains a total of $970k$ mentioning Trump and $776k$ mentioning Biden. Over 90% of the tweets are sent by users not located in the U.S. or in languages other than English. Hence, I omitted these observations and only focus on the remaining $266k$ tweets representative of the U.S. voters who are active on Twitter.

The main variables I have available are as follows:

1. Tweets. The raw tweets are cleaned to be free of punctuation, pronouns, and hyperlinks. Further, the NLTK stopwords and Word-Lemmatization resources help me to clean the tweets.
2. The number of likes a tweet receives.
3. The number of retweets a tweet receives.
4. Location of the user, specific to state.
5. The number of followers of the user
3. Date of tweet posting

I'm also using another tweet dataset named 'Knowledge Enhance Masked Language Model for Stance Detection' published by Georgetown University (Kawintiranon and Singh 2021). Every observation of the dataset is also a tweet, and each observation is human-labeled with its political stance expression: opposing, neutral, or supportive of Trump or Biden. This dataset help classify the political stance of the tweets coming from my main dataset with $266k$ observations with the help of a dictionary produced by the logit-lasso regression. By the nature of logit-lasso regression, we can only produce the most accurate keyword dictionary if the number of observations labeled as Trump vs. Biden leaning are relatively equal. Out of the 2500 labeled tweets, 35% of observations are Biden-supporting or Trump-opposing, 33% of observations are Trump-supporting or Biden-opposing, and 32% of observations take a neutral stance. This prepares for the accuracy of the logit-lasso regression so that the word-list is not biased.

2.2 Summary Statistics

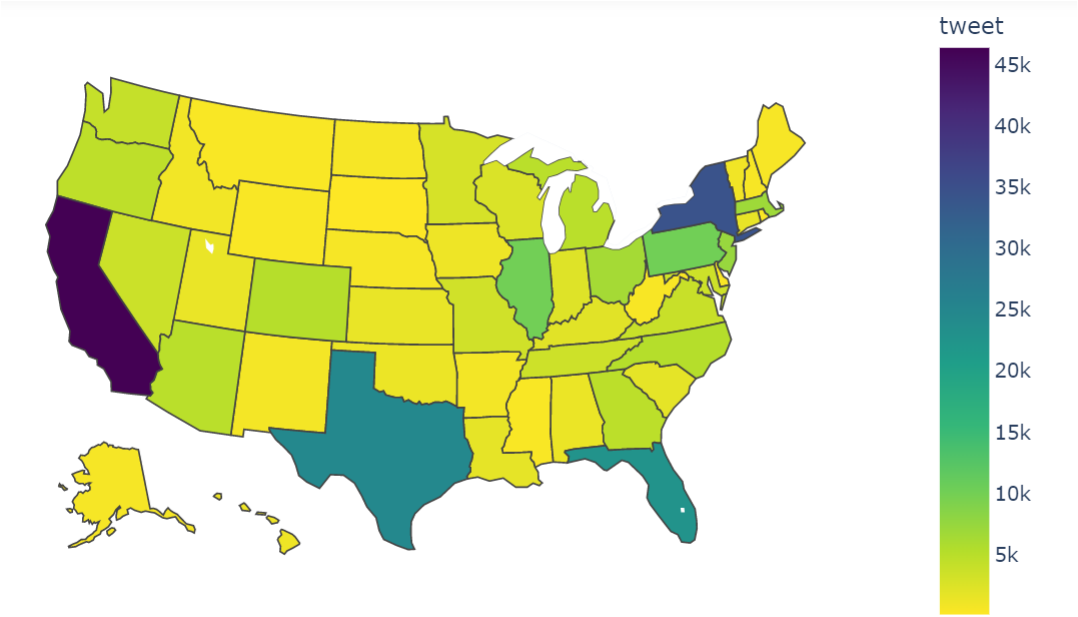

Figure 1: Geographic Distribution of tweets from period 10/15-11/08

Out of the $266k$ tweets for the 2020 Election spanning three weeks, the states with the most and least tweets posted related to Trump and Biden are mapped in Figure 1. Ranking the most represented tweet sources with more than $10k$ tweets for each state, $46,381$ are from California, $34,468$ are from New York, $24,659$ are from Texas, $22,735$ are from Florida, $12,117$ are from District of Columbia, $10,183$ are from Pennsylvania, and $10,157$ are from Illinois.

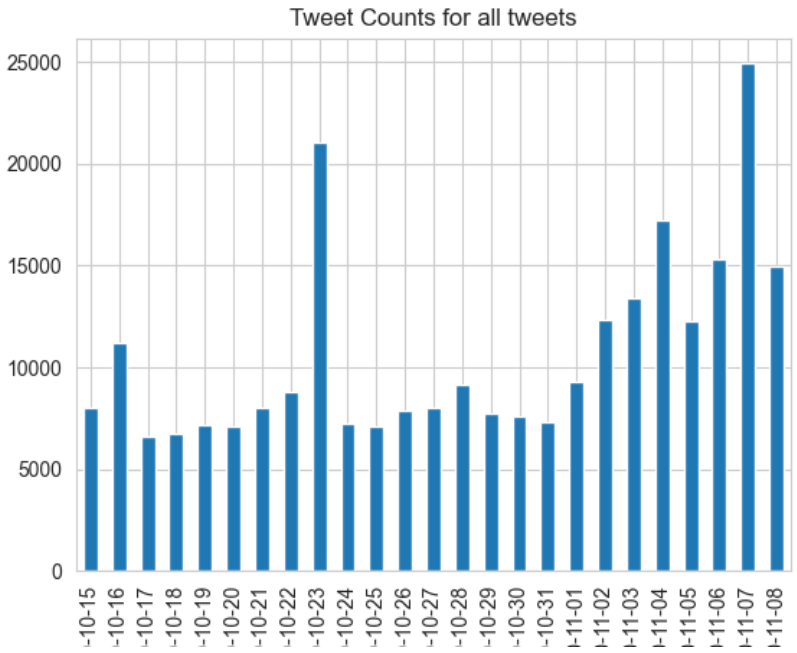

Figure 2: Number of tweets posted over time

For the span of roughly 3 weeks, the tweet counts remained relatively stable until the date 10/13, possibly due to the final presidential debate held in Nashville, spurring huge discussions over Twitter. The number of tweets swinged back to fewer than $10k$ per day, until the day before Eletion Day on 11/02. Tweet counts continue to rise until reaches a maximum of $25k$ on 11/07, where most social media outlets started announcing Biden as the 2020 Election winner.

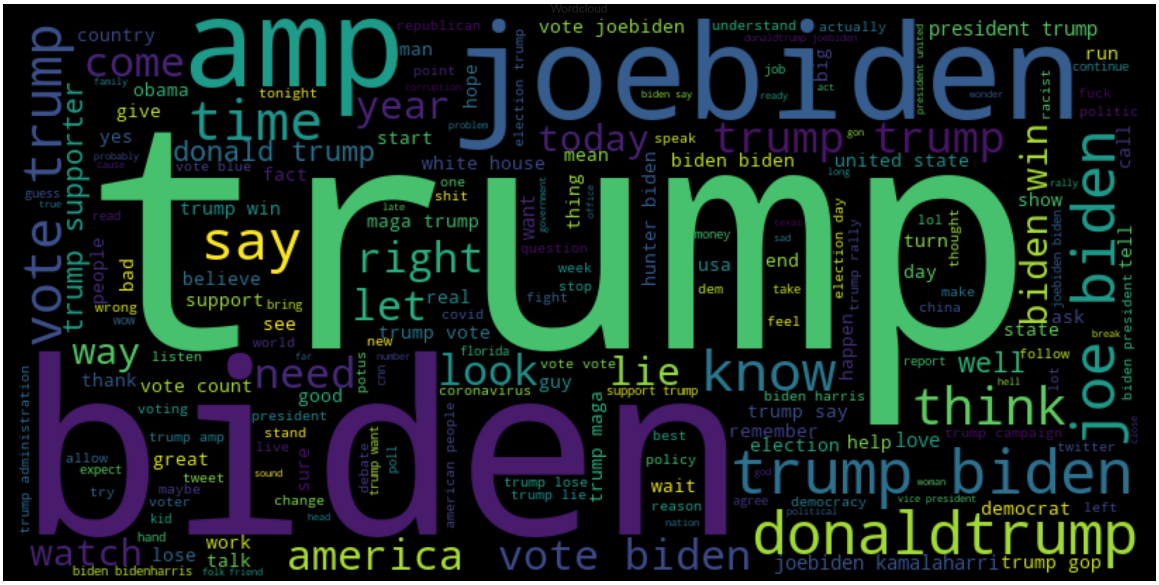

Figure 3: Word-Cloud for tweet text

The word-cloud in Figure 2 provides a glimpse of the most frequently mentioned words in the tweets. Apart from the names of the candidates, such as 'trump', 'donaldtrump', and 'joebiden', words related to the election are most common: 'amp' (standing for American President), 'win', 'election', 'america', 'count' are most common. The wordcloud is also proliferated with words that show impetus, urge, opinion, and motivation, such as 'time', 'right', 'know', 'think', 'way', 'bad', 'wrong', and 'lie'.

Figures 4-6 show the probability distributions for several parameters in the dataset. Figure 4 suggests that most users post fewer than 5 tweets over the time period of interest, whilst there are around 12% of tweets coming from users who post more than 5 tweets relevant to the election. About 80% of tweets receive no retweets at all, and 14% receiving 1 to 5 retweets. Very few observations receive more than $1k$ retweets, with aggregated probability close to 0 as shown in Figure 6. Similarly, around 57% of tweets receive no likes at all, 20% receive only one like, and 16% receive 3 to 5 likes. There are a few rare instances where one popular blogger receives many likes, but such extreme instances are not representative.

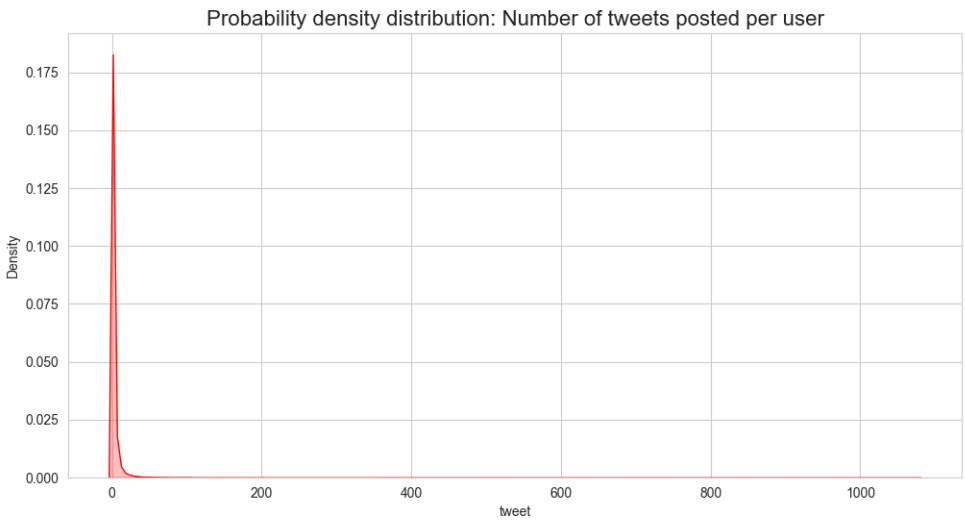

Figure 4: Probability Distribution: Number of Tweets posted per user

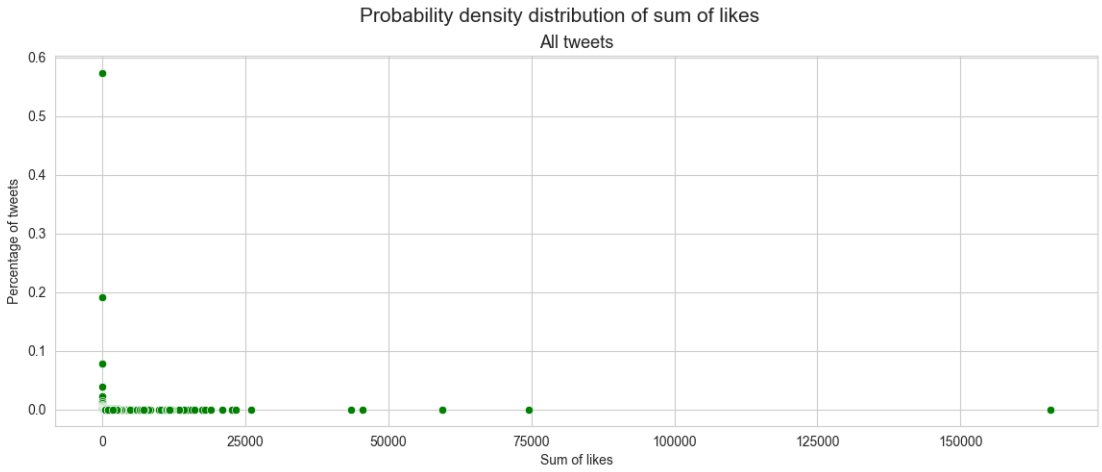

Figure 5: Probability Distribution: Number of likes received by each Tweet

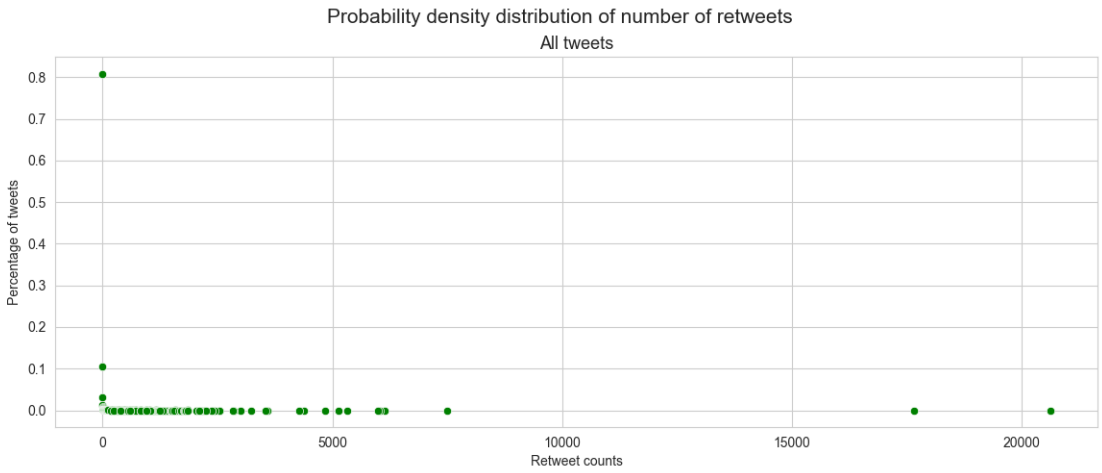

Figure 6: Probability Distribution: Number of retweets received by each Tweet

3 Methods

First, I attempt to classify tweets into Trump versus Biden leaning with the help of 2500 tweets that are already hand-labeled with candidate preference. A word dictionary that best predicts the political stance of this hand-labeled dataset is further used to classify my own dataset containing about $266k$ unlabeled tweets. Then, I run Latent Dirichlet Allocation (LDA) to separately on Trump and Biden leaning tweets to study what political topics each camp is more interested in, and whether the share of tweets related to one topic (e.g. healthcare) changes significantly each day before the election. Last, I employ a pre-trained algorithm named Twitter-roBERTa-base from HuggingFace to analyze the sentiment of each tweet on the share of positive, negative, and neutral words.

3.1 Classifying Political Stance

To classify the political leanings (Trump vs. Biden) of tweets for the $266k$ observations, I run a logit-lasso regression on 2500 existing Election2020 tweets. The data is already hand-labeled with political stance as supporting, opposing, or neutral of Biden or Trump. I further collapse the data into three categorical dummies: 1) Supportive of Trump or opposing of Biden, 2) Neutral stance, and 2) Supportive of Biden or opposing of Trump.

3.2 Topic modeling with LDA

Latent Dirichlet Allocation (LDA) is an unsupervised learning task that helps with classifying text documents into different topics. Instead of using a deterministic approach to classify each tweet, it assumes that each tweet is a mixture of topic probabilities. It also assumes each topic is a probability distribution for an underlying set of words.

This algorithm allows me to do two things.

1. Extract topic from the corpus. I can customize the number of topics I want the algorithm to classify into. The algorithm will tell me the most frequent words associated with each topic. I can then name the topics on my own discretion
2. Label each document with the probability of it belonging to one topic. LDA will assign the topic labels to the tweets based on the probability distribution of each potential topic

3.3 Sentiment analysis using roBERTa

This is a roBERTa-base model trained on 58M tweets and fine-tuned for sentiment analysis with the TweetEval benchmark (Camacho-Collados et al. 2022). It is updated as of 2022, which includes a fairly recent collection of tweets to be trained on. This algorithm is also tuned for Twitter sentiment analysis, and can better handle with things like emojis and strong hate speech (Loureiro 2022).

RoBERTa can classify the tweets as showing positive, neutral, or negative emotions by sentiment percentages. In contrary to other algorithms given by NLTK that automatically label a tweet with one sentiment, it avoids the potential of mislabeling tweets in cases where the sentiment percentages of each category are very close. RoBERTa can also identify irony, hate speech, and offensive language.

4 Results and Expected Results

4.1 Political Stance (expected implementation)

I start with converting the cleaned hand-labeled dataset into a document-term matrix that has vector representations of the frequencies of words. Specifically, I use TFIDF as the text feature extraction method. This TFIDF matrix is the independent variable of interest. The dummy political stance variable (three values) is treated as the outcome of interest. The hand-coded dataset is split by a ratio

of 8 : 2, assuring that the train and test datasets contain almost identical shares of political stance dummies.

Using the logit regression with L1 penalty, I run the regression with optimal penalty term that produces the highest accuracy score for the testing data. The predicted political stance on the hand-labeled data could then further give me the highest coefficients on a set of words, which are the words most predictive of each political stance. Using such word dictionary, I can apply it to classify the $266k$ tweets in my main dataset.

4.2 Topic modelling results (half-way)

I started with converting the corpus into a document-term matrix, setting the number of topics to 20. The results look like this:

```
[(0,
  '0.098*"trump" + 0.042*"covid" + 0.035*"life" + 0.028*"speech" + 0.027*"death" + 0.021*"matter"
+ 0.021*"dead" + 0.020*"hold" + 0.020*"coronavirus" + 0.020*"people"'),
 (1,
  '0.121*"trump" + 0.092*"supporter" + 0.044*"work" + 0.042*"american" + 0.039*"people" + 0.036*"h
elp" + 0.034*"million" + 0.033*"god" + 0.026*"family" + 0.026*"thank"'),
 (2,
  '0.077*"trump" + 0.045*"hate" + 0.042*"court" + 0.040*"away" + 0.024*"listen" + 0.023*"pandemic"
+ 0.023*"term" + 0.021*"friend" + 0.021*"conference" + 0.019*"kag"'),
 (3,
  '0.139*"trump" + 0.075*"win" + 0.066*"state" + 0.049*"know" + 0.046*"right" + 0.037*"need" + 0.0
35*"say" + 0.025*"fight" + 0.024*"want" + 0.021*"lead"'),
 (4,
  '0.127*"trump" + 0.056*"loser" + 0.054*"trumpislosing" + 0.029*"whitehouse" + 0.027*"cnn" + 0.02
5*"foxnews" + 0.021*"total" + 0.016*"long" + 0.016*"case" + 0.015*"nevada"'),
 (5,
  '0.173*"trump" + 0.054*"donald" + 0.026*"talk" + 0.019*"golf" + 0.016*"election" + 0.015*"turn"
+ 0.014*"trumpcrimefamily" + 0.014*"lose" + 0.014*"votehimout" + 0.013*"loss"'),
 (6,
  '0.125*"trump" + 0.062*"like" + 0.031*"look" + 0.024*"bad" + 0.022*"man" + 0.020*"good" + 0.017
*"people" + 0.016*"get" + 0.016*"history" + 0.015*"result"'),
 (7,
```

Figure 7: 20 topics and the most frequent words associated

```
 (7,
  '0.111*"trump" + 0.057*"lose" + 0.033*"year" + 0.026*"world" + 0.024*"fuck" + 0.022*"victory" +
0.020*"tell" + 0.019*"like" + 0.019*"time" + 0.016*"change"'),
 (8,
  '0.206*"president" + 0.060*"donaldtrump" + 0.033*"racist" + 0.029*"obama" + 0.029*"crime" + 0.02
7*"sad" + 0.025*"administration" + 0.020*"criminal" + 0.017*"prison" + 0.017*"office"'),
 (9,
  '0.099*"trump" + 0.053*"support" + 0.045*"love" + 0.026*"ask" + 0.024*"cheat" + 0.018*"senate" +
0.016*"accept" + 0.016*"question" + 0.015*"free" + 0.015*"dems"'),
 (10,
  '0.151*"trump" + 0.112*"election" + 0.031*"watch" + 0.025*"happen" + 0.022*"day" + 0.022*"live"
+ 0.019*"best" + 0.019*"night" + 0.018*"rally" + 0.017*"yes"'),
 (11,
  '0.257*"vote" + 0.073*"trump" + 0.072*"ballot" + 0.030*"blue" + 0.023*"mail" + 0.021*"today" +
0.019*"elect" + 0.018*"wait" + 0.015*"voting" + 0.014*"voter"'),
 (12,
  '0.124*"trump" + 0.035*"twitter" + 0.029*"tweet" + 0.020*"evidence" + 0.019*"press" + 0.016*"cor
ruption" + 0.016*"biden" + 0.016*"election" + 0.015*"patriot" + 0.015*"post"'),
 (13,
  '0.052*"medium" + 0.045*"trump" + 0.044*"news" + 0.028*"decide" + 0.027*"mean" + 0.026*"money" +
0.025*"fake" + 0.019*"refuse" + 0.019*"hell" + 0.018*"russia"'),
```

Figure 8: 20 topics and the most frequent words associated, continued

```
 (14,
  '0.261*"amp" + 0.080*"fraud" + 0.059*"trump" + 0.040*"steal" + 0.026*"word" + 0.020*"video" + 0.
019*"voterfraud" + 0.018*"trumpout" + 0.017*"team" + 0.017*"voter"'),
 (15,
  '0.063*"legal" + 0.050*"bidenharris" + 0.037*"lawsuit" + 0.033*"trump" + 0.023*"blm" + 0.022*"co
unty" + 0.020*"target" + 0.017*"meme" + 0.016*"national" + 0.014*"mark"'),
 (16,
  '0.128*"donaldtrump" + 0.117*"trump" + 0.054*"maga" + 0.052*"republican" + 0.048*"vote" + 0.044
*"america" + 0.041*"count" + 0.041*"election" + 0.039*"gop" + 0.027*"democrat"'),
 (17,
  '0.233*"trump" + 0.059*"lie" + 0.033*"claim" + 0.028*"fire" + 0.023*"truth" + 0.020*"told" + 0.0
19*"run" + 0.017*"people" + 0.014*"course" + 0.013*"campaign"'),
 (18,
  '0.086*"white" + 0.078*"house" + 0.033*"fact" + 0.026*"follow" + 0.023*"trump" + 0.021*"celebrat
e" + 0.021*"party" + 0.020*"check" + 0.018*"speak" + 0.016*"trumpsupporters"'),
 (19,
  '0.136*"trump" + 0.034*"pennsylvania" + 0.032*"counting" + 0.030*"georgia" + 0.029*"new" + 0.024
*"arizona" + 0.023*"michigan" + 0.023*"stop" + 0.021*"call" + 0.020*"florida"')]
```

Figure 9: 20 topics and the most frequent words associated,continued

With the 20-topics limit, the algorithm spit out a sizable collection of interpretive topics. Just to name a few:

The topic for index 0 could be clearly interpreted as a COVID-related topic. Index 4 is a topic related to Trump losing the election as demonstrated in trump-specific ballot results. Index 13 and 14 are clearly topics related to voter fraud and stealing votes. Index 19 is specifically about news about

ballot results in several big states.

Over the next few weeks, I expect to increase the number of topics fed into the algorithm and only retain ones that make the most sense to the researcher.

4.3 Sentiment analysis (code completed, requires combination with previous tasks)

I already finished running some preliminary sentiment analysis on a snippet of the dataset. Completing the task on $266k$ observations would take at least a few hours by computer. The sentiment analysis breakdown is expected to look like this:

```
{'negative': 0.54018086, 'neutral': 0.41085035, 'positive': 0.048968825}
{'negative': 0.76137674, 'neutral': 0.2061932, 'positive': 0.03243009}
{'negative': 0.17228034, 'neutral': 0.7520848, 'positive': 0.07563478}
{'negative': 0.36941049, 'neutral': 0.54789525, 'positive': 0.082694225}
{'negative': 0.31521976, 'neutral': 0.6122439, 'positive': 0.07253636}
{'negative': 0.91942877, 'neutral': 0.07099589, 'positive': 0.0095753735}
{'negative': 0.06223335, 'neutral': 0.37521285, 'positive': 0.56255376}
{'negative': 0.005730424, 'neutral': 0.22914432, 'positive': 0.7651252}
{'negative': 0.45147625, 'neutral': 0.47564164, 'positive': 0.072882116}
{'negative': 0.6664684, 'neutral': 0.29590663, 'positive': 0.03762497}
```

Figure 10: Sample of sentiment analysis using roBERTa

From here, we can see that each tweet is divided into sentiments with shares of positive, neutral, and negative sentiment. Using the classification results from political stance and topic modeling, I can further analyze sentiments according to differnt political stances and topics. For example, topics involving COVID-19 in support of Joe Biden might be more positive, while topics related to Voter Fraud for Trump supporters might be more polarized.

It is also expected that each tweet is put with three additional labels for irony, hate speech, and offensive language. Each label is a binary dummy detecting if the tweet reveals such characteristics. This label could be interpreted as a polarity measurement across political stances and topics.

5 Conclusion

This paper attempts to do three things: 1) Classify the political stance of election tweets with machine learning using pre-labeled tweets 2) Conduct topic modeling on the tweets and obtain a probability distribution of each tweet belonging to each topic, and 3) Analyze tweet sentiments and polarity in combination with stance and sentiment detection. Given the date the tweet is posted and parameters like retweeting and like counts, we can conclude with a potential correlation between public attention and tweet sentiment over different political topics.

6 References

Camacho-Collados, José, Kiamehr Rezaee, Talayeh Riahi, Asahi Ushio, Daniel Loureiro, Dimosthenis Antypas, Joanne Boisson, Luis Espinosa-Anke, Fangyu Liu, Eugenio Martínez-Cámara, Gonzalo Medina, Thomas Buhrmann, Leonardo Neves and Francesco Barbieri. "TweetNLP: Cutting-Edge Natural Language Processing for Social Media." Conference on Empirical Methods in Natural Language Processing (2022).

Kawintiranon, Kornraphop and Lisa Singh. Knowledge Enhanced Masked Language Model for Stance Detection. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4725–4735, Online. Association for Computational Linguistics. (2021).

Loureiro, Daniel, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke and José Camacho-Collados. "TimeLMs: Diachronic Language Models from Twitter." Annual Meeting of the Association for Computational Linguistics (2022).

Manch Hui. U.S. Election 2020 Tweets. January, 2021. Distributed by Kaggle.
https://www.kaggle.com/datasets/manchunhui/us-election-2020-tweets