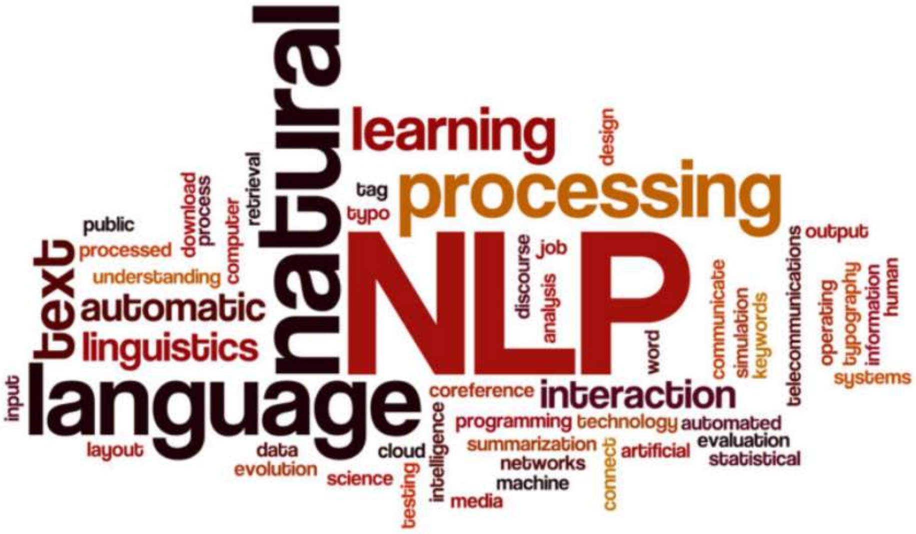


POLITICAL SENTIMENT ANALYSIS VIA WEBSCRAPING AND NLP

University of Denver - Ritchie School of Engineering

DS Tools 1 – Summer 2020

ABSTRACT

An exploration of overall political sentiment using Python webscraping and natural language processing (NLP) functionalities. Data was collected from Reddit and Twitter and subsequently cleaned and analyzed using existing Python data science libraries and packages.

Co-Contributors

Sawyer Jacobson
Aran Miller
Sameer Patel

Data Set and Motivation

For this project, the team decided to investigate general political sentiment based on analysis of user-submitted content to two of the most popular websites in the Unites States – Reddit and Twitter. In order to do so, the data would need to be analyzed through techniques common to the Natural Language Processing (NLP) field of data science.

The data collection, cleaning, and analysis process was conducted as follows:

1. Collection of buzzwords from two popular politically-aligned subreddits - /r/Liberal and /r/Conservative
    a. Store ~1,000 headlines from each subreddit using the Python praw library
        i. Headlines are first filtered to include the top (most upvoted) user-submitted posts over the past year
    b. Remove stop words using the Python NLTK package
    c. Identify the top 5 unique buzzwords for each political ideology based on usage frequency and domain knowledge
        i. Domain knowledge helps to eliminate shared buzzwords between the subreddits and helps to tailor retrieved Tweets to align with political sentiment more likely
    d. Create word cloud visualizations using the Python "wordcloud" package and matplotlib
2. Mine Tweet data by using Twitter API to return Tweets containing the topics (10 total) identified in Step 1
    a. Filter to only return English Tweets
    b. Extract Tweet text
    c. Remove certain types of irrelevant information to get a list of important words from the Tweets (eg. punctuation, emojis, stopwords, etc.)
        i. get_things()
            1. Used to collect data pieces from the 'entities' column of the tweet object
        ii. get_tweets()
            1. Takes the desired search topic and number of tweets as input.
            2. Uses a developer twitter API account to download tweets
            3. Returns a pandas dataframe
        iii. text_cleaner()
            1. A text cleaning function
            2. Expands contractions, removes stopwords, and lemmatizes a text data piece
        iv. clean_tweets()
            1. Function used for cleaning the text tweet data
            2. Removes "RT", "@" (user mentions), urls, extracts emojis, and performs text cleaning similar to text_cleaner()
        v. tweet_generator()
            1. Modifies the tweets to have the user_id and retweeted_status_id as columns
    d. Create word cloud visualizations using the Python "wordcloud" package and matplotlib

3. Conduct sentiment analysis on Tweet text using MIT's Valence Aware Dictionary and sEntiment Reasoner (VADER) Python package
    a. Remove duplicate Tweets from dataframe of Tweets returned in step 2
    b. Compute metrics such as individual Tweet compound score, mean sentiment score by topic, and weighted average sentiment score based on number of retweets
4. Graphically visualizing overall sentiment of the topics identified based on model performance
    a. Bar plot of average sentiment
    b. Bar plot of mean retweet count by topic

The metadata important to conducting this analysis includes the following:

- Reddit
    - Post title
        - Dictionary of important words and their frequencies
    - Number of upvotes
    - Number of comments
    - Creation date
- Twitter
    - Tweet text
        - Dictionary of important words and their frequencies
    - Creation date
    - Retweet count

Research Question

Are we able to determine which political group, conservative or liberal, is more positive/negative using sentiment analysis of Twitter data containing buzzwords specific to their respective boards/platforms? As briefly aforementioned in the Data Set and Motivation section, the input data necessary to answer this question comes in the form of Tweet text, and the output is a computation of sentiment score and subsequent visualizing of overall sentiment of the buzzwords identified from the subreddit webscraping.

Literature Review / Method of Addressing Research Question

NLP is an extremely complex field of data science, with a multitude of methodologies and models currently existing to perform specific types of analyses. In trying to stick to our goal of sentiment analysis, the team researched a multitude of techniques currently in use in this field.

Sentiment Analysis, also called Opinion Mining, tries to identify and extract opinions within a given text. The aim of sentiment analysis is to gauge the overall sentiment, quantitatively, of a piece of text based on the computational treatment of subjectivity in a text. Understandably, sentiment analysis is an extremely tricky endeavor, prone to errors given the subjectivity of natural language. Complications can arise in the form of multiple sentiments being expressed in the same sentence (multiple polarity), usage of emojis and emoticons, slang words, and degree of sentiment based on adverbs, to name just a few. Luckily, several open-source analytical packages have been created and distributed within the Python community to help the team towards answering the research question.

There are two broad approaches to sentiment analysis: purely statistical, and a mixture of statistics and linguistics. The latter approach incorporates grammar principles and various natural language processing techniques to train a model to ‘understand’ language. This approach requires a training set of data that has been pre-tagged as positive/negative/neutral to return results for new data. The former approach, known as Bags of Words (BOW) or lexicon-based sentiment analysis, was selected by the team in this project as the best way of analyzing Tweet text.

Most sentiment analysis approaches can further be split into one of two forms: polarity-based, where text is classified as either positive, negative, neutral, or valence-based, where the intensity of the sentiment is also considered. For this analysis, the team pursued a valence-based statistical approach of sentiment analysis through the Valence Aware Dictionary and sentiment Reasoner (VADER) package.

VADER is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media (eg. Tweet text). Foundationally, it uses a list of lexical features (e.g. words, emojis, punctuation, etc.) which are generally labelled according to their semantic orientation as either positive or negative to produce a “compound score”. The compound score is a metric that calculates the sum of all the lexicon ratings which have been normalized between -1 (extremely negative) and +1 (extremely positive). The beauty of VADER is that it operates on the entirety of Tweet text – meaning ordinary text cleaning processes (such as stop word removal, stemming/lemmatization, punctuation/emoji removal, etc.) are not necessary and are actually detrimental to the accuracy of the score returned by the algorithm. It is fully open-sourced under the MIT license (https://github.com/cjhutto/vaderSentiment#introduction).

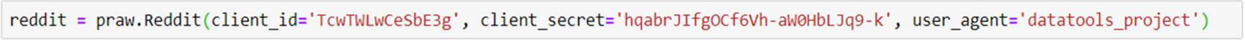
For this analysis, the team will use the VADER package to return individual compound scores for the Tweets returned for each topic. The team will then synthesize the results both quantitatively and visually as detailed in further sections.

Quality of Cleaning

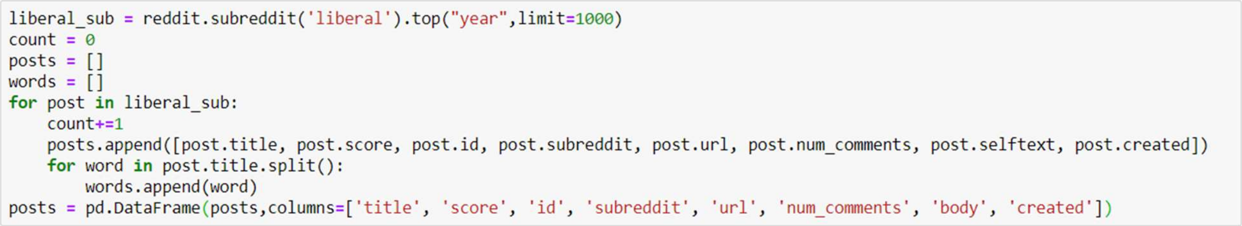
Data Cleaning and Type Conversion

All forays into NLP begin with cleaning of the text data; including processes such as tokenization, stemming/lemmatizing, removal of stop words/accessory data (eg. excess blanks, emojis, etc.), and others. For this specific project, the praw library (https://praw.readthedocs.io/en/latest/) was employed in combination with the nltk library to extract meaningful buzzwords from Reddit headlines. Samples of code detailing parts of the data cleaning process are shown below; given space constraints, full code will only be viewable in the attached Jupyter notebooks.

First, praw methods were used to extract Reddit data:

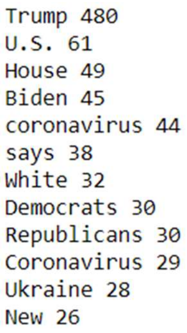


```
reddit = praw.Reddit(client_id='TcwTWLwCeSbE3g', client_secret='hqabrJIfgOCf6Vh-aW0HbLJq9-k', user_agent='datatools_project')
```



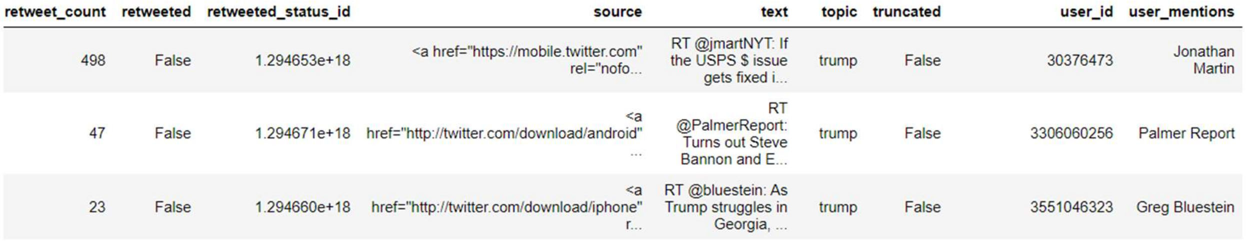
```
liberal_sub = reddit.subreddit('liberal').top("year",limit=1000)
count = 0
posts = []
words = []
for post in liberal_sub:
    count+=1
    posts.append([post.title, post.score, post.id, post.subreddit, post.url, post.num_comments, post.selftext, post.created])
    for word in post.title.split():
        words.append(word)
posts = pd.DataFrame(posts,columns=['title', 'score', 'id', 'subreddit', 'url', 'num_comments', 'body', 'created'])
```

Stop words were then removed from post titles, and a frequency dictionary was created for important words from the combined titles (1000 posts for both the /r/liberal and /r/conservative subreddits).



```
Trump 480
U.S. 61
House 49
Biden 45
coronavirus 44
says 38
White 32
Democrats 30
Republicans 30
Coronavirus 29
Ukraine 28
New 26
```

5 buzzwords per subreddit were selected from the list of frequently occurring words, which were then used to extract Tweets using Twitter API. The code to do so is oppressively long in the context of a written report, so only a sample of the returned dataframe containing Tweet information is displayed below:



retweet_count	retweeted	retweeted_status_id	source	text	topic	truncated	user_id	user_mentions
498	False	1.294653e+18	<a href="https://mobile.twitter.com" rel="nofo...	RT @jmartNYT: If the USPS $ issue gets fixed i...	trump	False	30376473	Jonathan Martin
47	False	1.294671e+18	<a href="http://twitter.com/download/android" ...	RT @PalmerReport: Turns out Steve Bannon and E...	trump	False	3306060256	Palmer Report
23	False	1.294660e+18	<a href="http://twitter.com/download/iphone" r...	RT @bluestein: As Trump struggles in Georgia, ...	trump	False	3551046323	Greg Bluestein

In order to generate word clouds for the Twitter data, the clean_tweets() methods was employed in order to return a list of the most meaningful words from the Tweet texts. Topic buzzwords were removed as well as they were inherently the most commonly occurring words. Since this analysis centers around text analysis, variable type conversion did not play much of a role. The only instance of type conversion conducted in this analysis was converting creation date information in the dataframes to datetime format using the pandas function to_datetime().

Tweets were filtered through the API to only include those written in English. In addition, duplicate Tweets were dropped from the pandas dataframe before conducting the sentiment analysis in order to eliminate issues with sentiment overstating.

Missing Values

Missing data (NaNs/nulls) was not an issue for this analysis, as data was scraped directly from the top 1,000 headlines (sorted by the most upvoted content over the year in both the /r/Liberal and

/r/Conservative subreddits), and post titles cannot be blank. Additionally, the Twitter API webscraping only returned Tweets which contained text specified by the topics returned from the Reddit data analysis, so blank Tweets would not be possible.

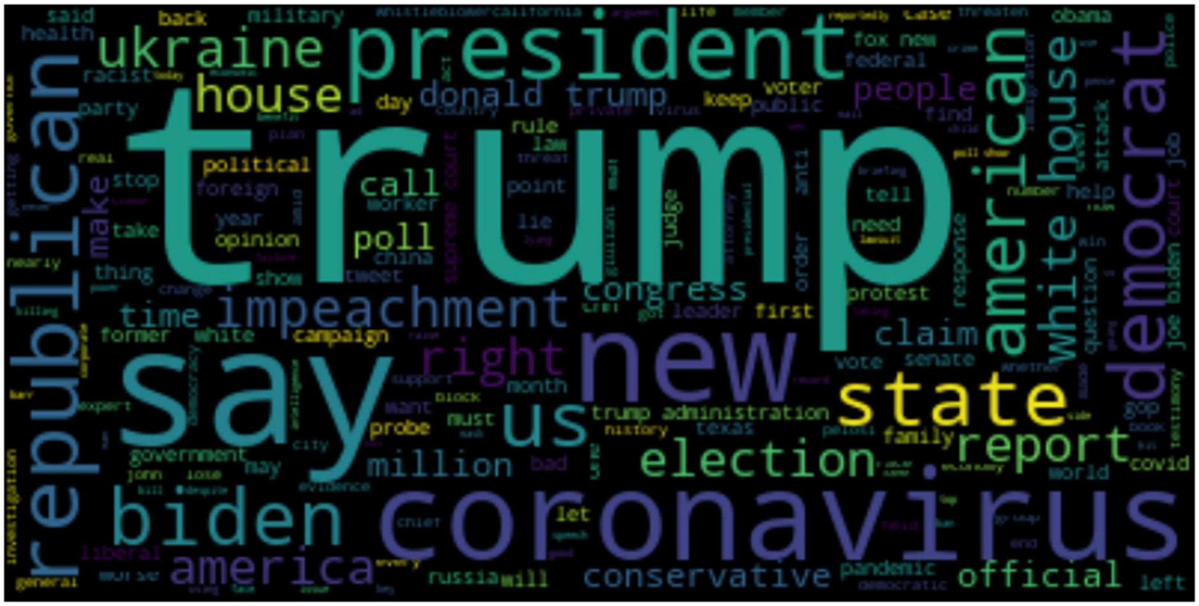
Attribute Creation, Summary Statistics, and Interpretation

As detailed in the Data Cleaning section above, a dictionary of counts was created showing the most frequently used words from /r/Liberal and /r/Conservative. From there, 10 unique buzzwords were identified (5 for each subreddit) and subsequently used to scrape Twitter data. The topics identified for each subreddit are:

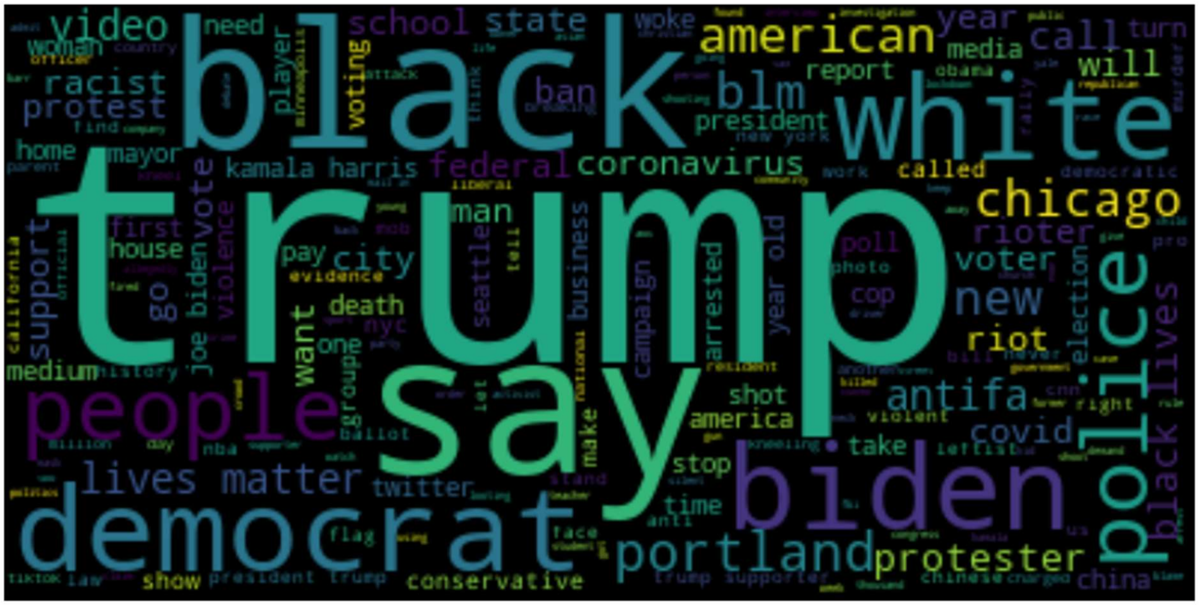
```
Conservative Topics:  ['trump', 'biden', 'blm', 'antifa', 'portland']
Liberal Topics:  ['coronavirus', 'democrats', 'ukraine', 'police', 'protest']
```

Word clouds generated from the post titles of the /r/liberal and /r/conservative subreddits are shown below.

Liberal Subreddit Word Cloud

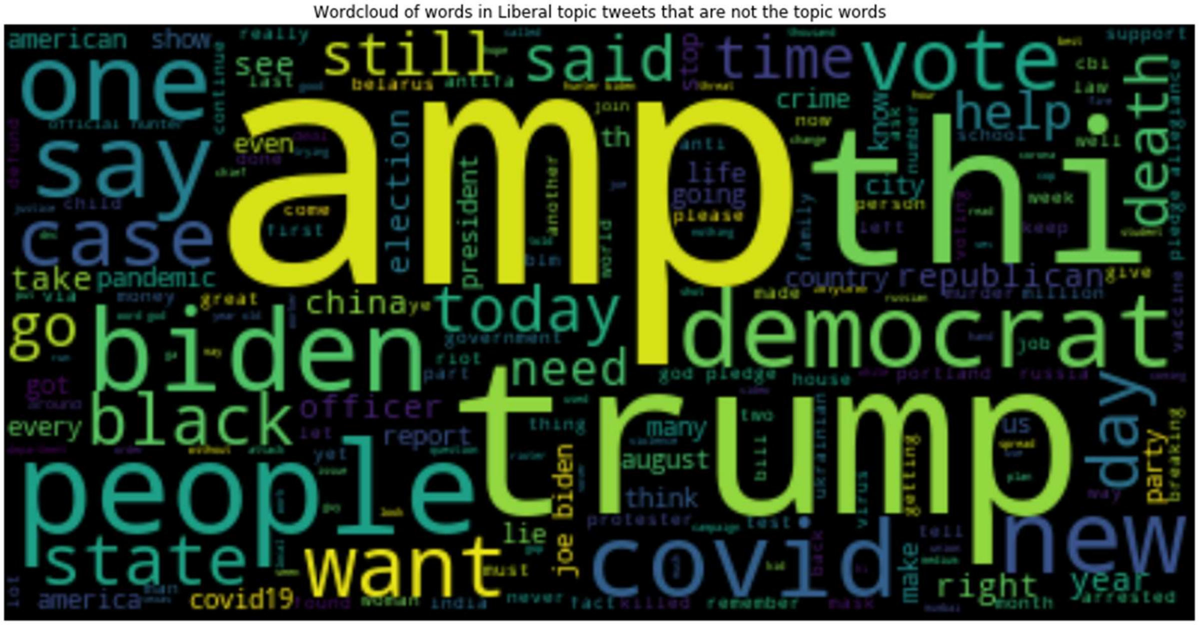


Conservative Subreddit Word Cloud

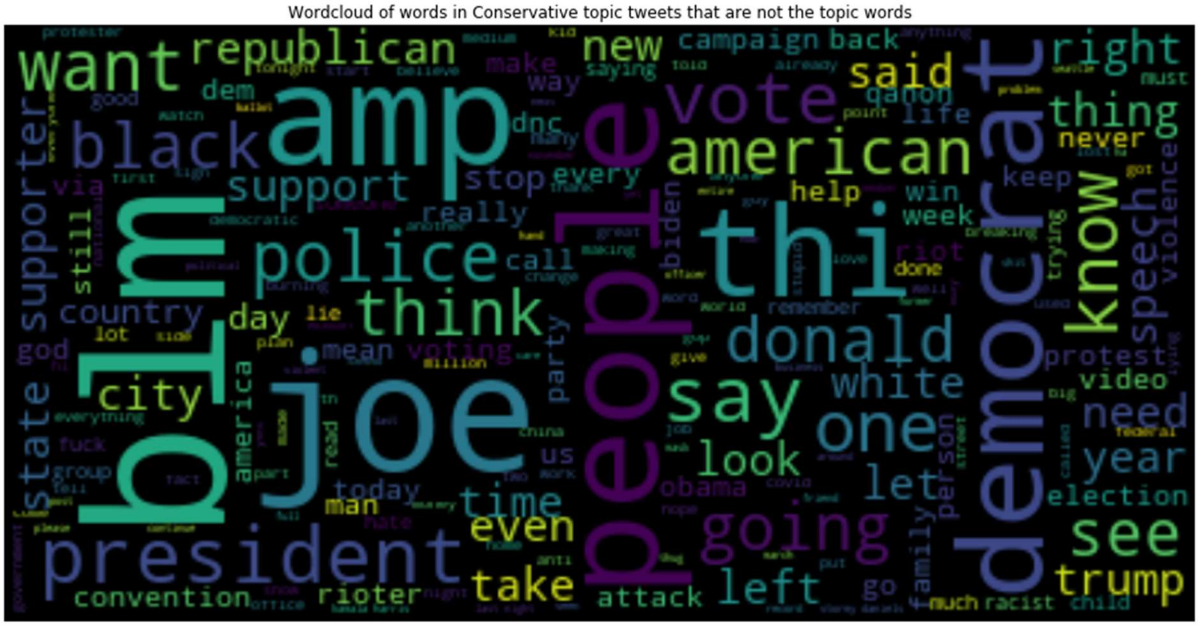


Word clouds generated from the liberal and conservative Tweets are shown below:

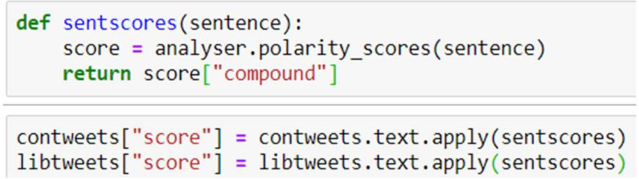
Liberal Tweet Word Cloud



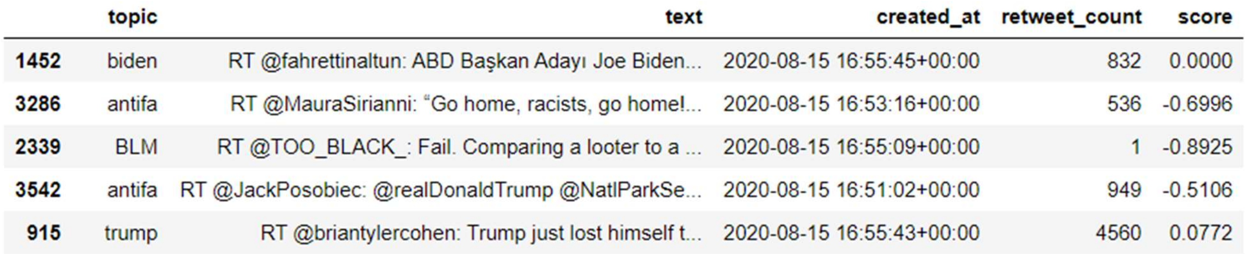
Conservative Tweet Word Cloud


Wordcloud of words in Conservative topic tweets that are not the topic words

The VADER package was then implemented on the original Tweet texts in order to quantify sentiment for Tweets pertaining to each topic. The "topic","text","created_at", and "retweet_count" attributes for both the liberal and conservative Twitter data were initialized into a new dataframe of only information pertinent to the team's analysis. A new attribute "score" was created by applying the SentimentIntensityAnalyzer polarity_scores() method to each individual Tweet. A sample dataframe showing compound scores is given below:

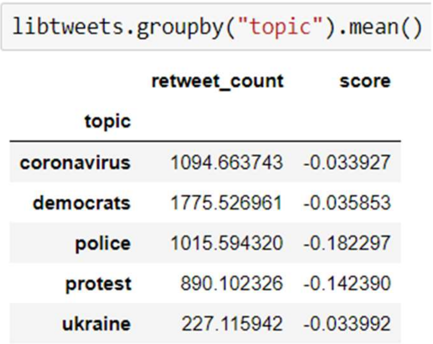

```
def sentscores(sentence):
    score = analyser.polarity_scores(sentence)
    return score["compound"]
```

```
contweets["score"] = contweets.text.apply(sentscores)
libtweets["score"] = libtweets.text.apply(sentscores)
```



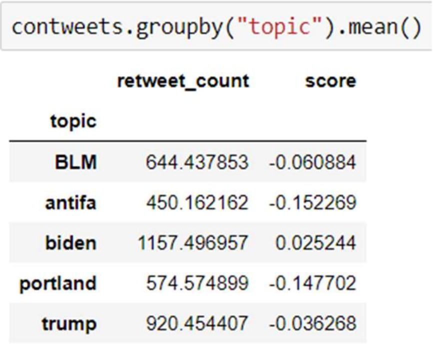
	topic	text	created_at	retweet_count	score
1452	biden	RT @fahrettinaltun: ABD Başkan Adayı Joe Biden...	2020-08-15 16:55:45+00:00	832	0.0000
3286	antifa	RT @MauraSirianni: "Go home, racists, go home!...	2020-08-15 16:53:16+00:00	536	-0.6996
2339	BLM	RT @TOO_BLACK_: Fail. Comparing a looter to a ...	2020-08-15 16:55:09+00:00	1	-0.8925
3542	antifa	RT @JackPosobiec: @realDonaldTrump @NatlParkSe...	2020-08-15 16:51:02+00:00	949	-0.5106
915	trump	RT @briantylercohen: Trump just lost himself t...	2020-08-15 16:55:43+00:00	4560	0.0772

The "score" column displays the average compound sentiment on a -1 to +1 scale for the text contained in the "text" column. The mean sentiment scores by topic are presented in the tables below:



```
libtweets.groupby("topic").mean()
```

topic	retweet_count	score
coronavirus	1094.663743	-0.033927
democrats	1775.526961	-0.035853
police	1015.594320	-0.182297
protest	890.102326	-0.142390
ukraine	227.115942	-0.033992



```
contweets.groupby("topic").mean()
```

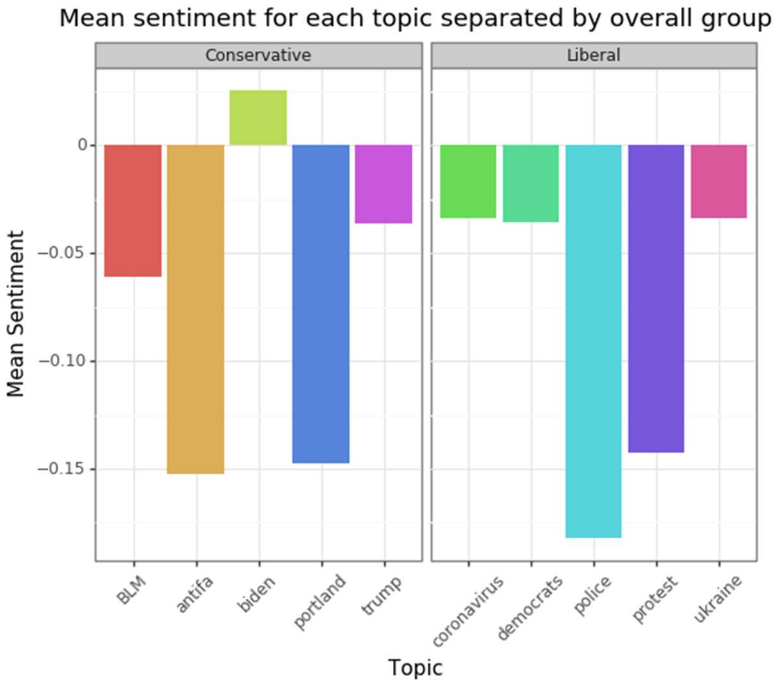
topic	retweet_count	score
BLM	644.437853	-0.060884
antifa	450.162162	-0.152269
biden	1157.496957	0.025244
portland	574.574899	-0.147702
trump	920.454407	-0.036268

As evidenced in the tables, the mean scores for most topics tend to be negative, while all of them center around 0 (as would be expected for an approximately-normal distribution of sentiments). This suggests that Twitter users tend to voice negative opinions more frequently than positive ones. The score for “Biden”, however, is slightly positive (0.025). This only represents a difference of ~0.06 from the Trump compound score, which makes it difficult to draw any conclusions regarding the 2020 elections based on this small sample of Tweet data.

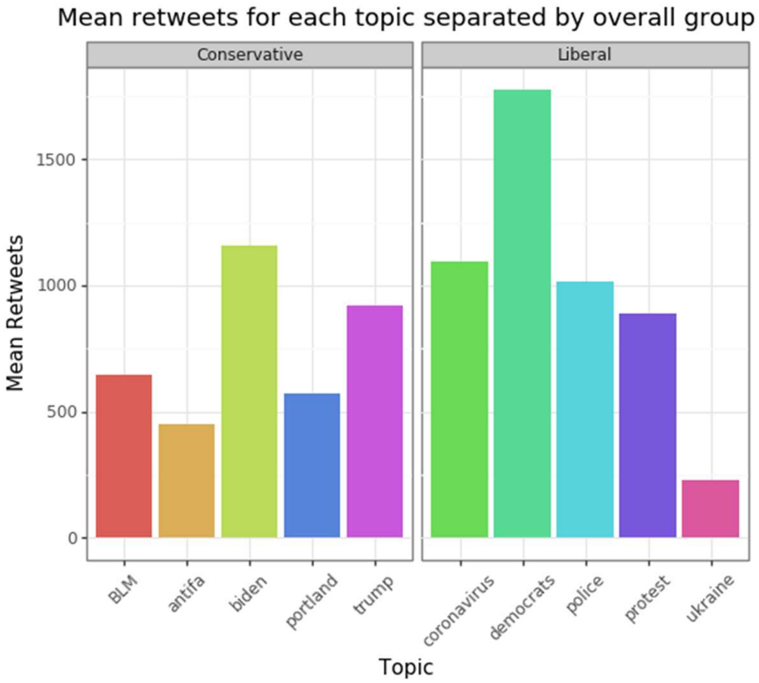
Visualization

Data Visualizations

A graph of average sentiment relative to the buzzwords identified is presented below.



A graph of average number of retweets relative to the buzzwords identified is presented below.



Description/Interpretation of Visualizations/Conclusions

From the above graphs, the team arrived at the following conclusions regarding overall sentiment of the specified terms:

- The topics identified are regarded in a mostly negative light according to the VADER compound scoring algorithm (and for just the sample of Tweets returned at the time of this analysis)
- Liberal topics (such as “democrats”, “police”, “protest”) tend to be more-discussed, based on retweet count, than conservative topics (such as “antifa”).
- The only positive score was for “biden”, but the difference was not large enough to draw a defensible conclusion regarding the upcoming election
- The most-negative score was for “police”, which the group expected given the recent political climate
- The above analysis opens up several considerations for future analyses:
    - Increasing the Tweet text sample size
    - Increasing the time period under which analysis can take place
    - Inclusion of more buzzwords
    - Inclusion of buzzwords that are more indicative of the political ideology under which they are nested

Connection to Understanding Data Distribution

None of the topics tended too far from the mean score of 0 on the -1 to +1 scale. This would suggest that the individual compound sentiment scores would be approximately normally distributed around 0. Future forays into NLP/Sentiment analysis would include a much larger set of Tweet data in order to more accurately represent the population sentiment for each of the topics identified.

Outliers/Other Issues

The primary issues encountered when conducting this analysis include dealing with duplicate Tweets and dealing with non-English Tweets.

For non-English Tweets, an option within Twitter API allowed the team to filter out most of them, but many were still able to slip through the cracks (especially Tweets concerning coronavirus).

Duplicate Tweets were an issue the team had some difficulty coming to an agreement on. On the one hand, the goal of this analysis was to gauge overall sentiment regarding the topics identified, so duplicate Tweets don't pose a problem given that they represent multiple peoples' opinions. On the other hand, the multiplicative attribute of including duplicate Tweets is already captured, for the most part, by the Retweet Count. There was also the issue that duplicate Tweets were almost always retweets themselves, and since our Twitter extract pulled Tweets over a relatively short time period, duplicates could severely complicate the end analysis. In the end, the decision was made to exclude duplicates from the sentiment analysis, which allowed for a more defensible representation of the compound sentiment scores for each topic.

Works Cited

Burchell, Jodie. "Using VADER to Handle Sentiment Analysis with Social Media Text." *Standard Error Full Atom*, 2017, t-redactyl.io/blog/2017/04/using-vader-to-handle-sentiment-analysis-with-social-media-text.html.

Gupta, Shashank. "Sentiment Analysis: Concept, Analysis and Applications." *Medium*, Towards Data Science, 19 Jan. 2018, towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17.

Hutto, CJ. "Cjhutto/VaderSentiment." *VaderSentiment*, GitHub, 2020, github.com/cjhutto/vaderSentiment#introduction.

Munir, Samira. "Basic Binary Sentiment Analysis Using NLTK." *Medium*, Towards Data Science, 27 Mar. 2019, towardsdatascience.com/basic-binary-sentiment-analysis-using-nltk-c94ba17ae386.

Pandey, Parul. "Simplifying Sentiment Analysis Using VADER in Python (on Social Media Text)." *Medium*, Analytics Vidhya, 8 Nov. 2019, medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f.

Pascual, Federico, et al. "Twitter Sentiment Analysis with Machine Learning." *MonkeyLearn Blog*, 4 Aug. 2020, monkeylearn.com/blog/sentiment-analysis-of-twitter/.

Tanner, Gilbert. "Scraping Reddit Data." *Medium*, Towards Data Science, 12 Feb. 2019, towardsdatascience.com/scraping-reddit-data-1c0af3040768.