|  |
| --- |
| Sawyer Jacobson |
| Aran Miller |
| Sameer Patel |

DS Tools 1 – Summer 2020

|  |  |
| --- | --- |
| Political Sentiment Analysis via Webscraping and NLP  University of Denver - Ritchie School of Engineering | 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 |

# 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
   1. Store ~1,000 headlines from each subreddit using the Python praw library
      1. Headlines are first filtered to include the top (most upvoted) user-submitted posts over the past year
   2. Remove stop words using the Python NLTK package
   3. Identify the top 5 unique buzzwords for each political ideology based on usage frequency and domain knowledge
      1. Domain knowledge helps to eliminate shared buzzwords between the subreddits and helps to tailor retrieved Tweets to align with political sentiment more likely
   4. 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
   1. Extract Tweet text
   2. Remove certain types of irrelevant information to get a list of important words from the Tweets (eg. punctuation, emojis, stopwords, etc.)
      1. get\_things()
         1. Used to collect data pieces from the ‘entities’ column of the tweet object
      2. 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
      3. text\_cleaner()
         1. A text cleaning function
         2. Expands contractions, removes stopwords, and lemmatizes a text data piece
      4. 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()
      5. tweet\_generator()
         1. Modifies the tweets to have the user\_id and retweeted\_status\_id as columns
   3. 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
   1. 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
   1. Bar plot of average sentiment

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, conservatives or liberals are 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.

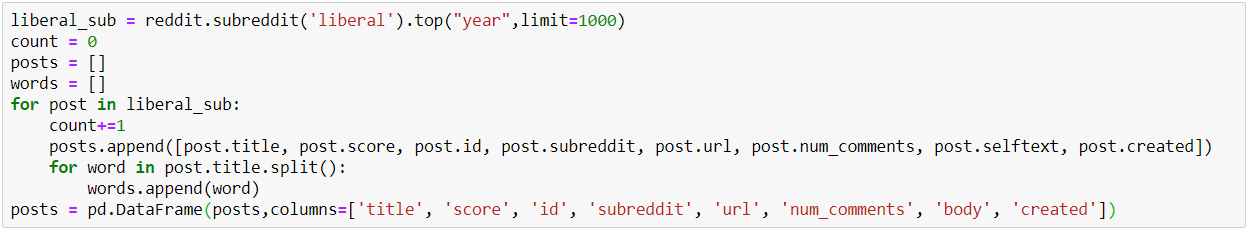
# Quality of Cleaning

## Data Cleaning and Type Conversion

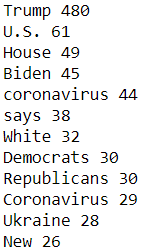
All forays into NLP begin with cleaning of the text data; including processes such as tokenization, stemming/lemmatizing, removal of stop words/accessory data (eg. excess blanks, emojis, etc.), and others. For this specific project, the praw library (<https://praw.readthedocs.io/en/latest/>) was employed 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:

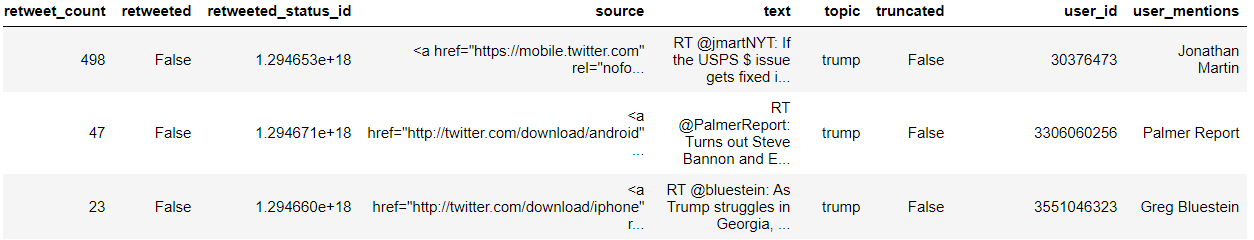




Stop words were then removed from post titles, and a frequency dictionary was created for important words from the combined titles (1000 posts for both the /r/liberal and /r/conservative subreddits).



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:



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 a role.

## 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:

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

Liberal:

A close up of text on a wooden surface

Description automatically generated

Conservative:

A close up of a logo

Description automatically generated

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

Liberal:

A close up of a sign

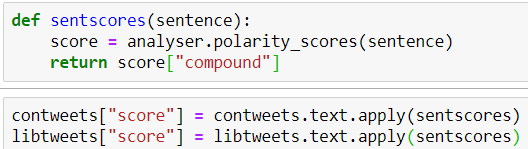
Description automatically generated

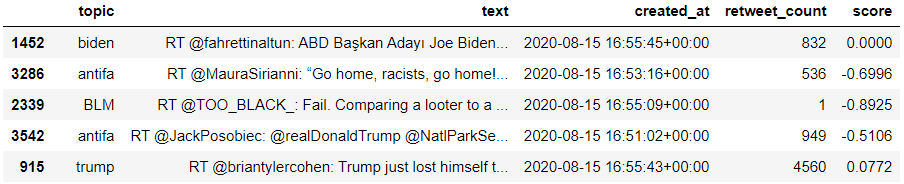
Conservative:

A picture containing green, table, sitting, wooden

Description automatically generated

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:





(insert section about mean score and weighted average score by topic, as well as interpretation of the results).

# Visualization

## Data Visualizations

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

A picture containing drawing

Description automatically generated

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

A screenshot of a cell phone

Description automatically generated

## Description/Interpretation of Visualizations

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

## Connection to Understanding Data Distribution

The specific line in the rubric is “What data visualization helped you understand about data distribution.”

My guess is he wants us to classify our data to a specific stats distribution? Any thoughts on what we need to write here would be hella appreciated. The data distribution is just text data so there really isn’t a distribution other than standard vocabulary. But I would say the word clouds help us understand to some extent. We removed the stopwords prior to that but if we didn’t, we would see tons of those. The wordclouds help us get a feel for the sentiment of the tweets on a high level. -Sawyer

## Outliers/Other Issues

I will update this section once our analysis is finalized.

# Conclusion

Based on the analysis and visualizations presented in the preceding sections, our group predicts …

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.