Cole Thompson

MSDS 430 SEC 56

Introduction

The United States, and western democracies alike, face the grave threat of psychological influence campaigns and information warfare. Undergoing a constitutional stress test, our nation has seen extreme partisanship—intoxicating our collective view of civil participation in the new digital reality before us. The very objective of these campaigns is to sew division, and without truly understanding the strategic tactics behind these efforts, we will not escape the grasp of the deteriorative divisiveness that has brought governmental progress to a screaming halt.

The objective of this project is to provide informative material that facilitates a better understanding of these tactics-- to demonstrate the seriousness and many dimensions of this matter through a non-partisan, data-driven lens. For the project, I obtained a Kaggle dataset containing all Twitter activity originating from the 'Internet Research Agency'—a Russian company indicted for their very participation in the noted influence operations. The visuals produced by the application were specifically chosen with a non-data savvy audience on top of mind—in order to most effectively communicate my message in a universally comprehensible manner.

Program Overview

The program was organized using the object-oriented paradigm and contains three distinct classes having different functionalities. The first class, named "TweetDataHandler", was created to handle all data processing tasks for the program. The class was also designed with much consideration for the program's testability, as the data source was quite large and posed challenges in terms of efficient testing.

For example, I incorporated a configuration that drives how many files are imported to give an ability to import a single file (rather than all 9) to speed up execution time. Additionally, there are two other configurations that drive whether the processed file is saved to a specified directory or not; and also whether the program reads the saved file or re-imports the raw data and performs the text pre-processing steps. This was also done for efficiency, where t pre-processing steps would not need to take place after the initial run—if the configuration were set to save the processed file and read from that directory instead.

The configuration that determines whether the file is saved was made separate from the other configuration in order to give the flexibility of re-testing/optimizing the text pre-processing steps without having to overwrite the processed file (if there were no impactful changes to processed file output, of course). There are also configurations that drive whether status messages are printed to the console during execution, where after completing all testing—these messages can be turned off in order to make the program less verbose. The configuration dictionary also contains the messages written to the console, where the programmer can re-use the statements by providing the message's key to the function that handles the messaging.

The initial data processing task that takes place is the extraction of "hash tags" from the Tweets. This was done as a means of storing this data for later analysis, given my thought that "hash tags" would serve as an illustrative datapoint for highlighting the communicative strategy of these influence campaigns; while also knowing that I would be removing them from the Tweet content themselves for other text analytics tasks. After storing the hashtags, the dataframe is filtered for solely tweets in the English language. I initially had a method to translate the text, although it was taking a significant amount of time and I was unable to optimize it to a point that would be suitable for the project's submission. For this reason, I chose to solely use English Tweets for the other languages would be difficult to consume during analysis.

In following, the Tweet content is removed of all hyperlink URLs, given these serve little use for the intended analysis. It is important to note that the hyperlinks were removed at this point in the process for a later implementation would have been ineffective. For the 3rd following step is to remove punctuation, the regular expression used to identify the hyperlinks would not have worked if taking place after this step. The immediate step taken after removing the hyperlinks is to remove the "hash tags" (the hash tags were only extracted at this point, not removed)—which would also need to take place before removing the punctuation, given a regular expression is used here as well to identify text subsequent to the hash symbol. A similar consideration was made for the removal of stop words, words that are insignificant to the conducted analysis, to ensure that words like "don't" were properly removed before stripping the text's punctuation. After removing the punctuation, all tweets are made lower case and lemmatization is performed for a more effective means of pattern identification.

The 2nd class, called "TweetCalculator", serves the role of performing calculations on the processed dataset; and returns simple calculated values and text tables to the console. The current implementation allows the user to retrieve data about the troll profiles' Twitter followers and well as the number of that the profiles' are following. The user may call a function to retrieve basic descriptive statistics of this information or call another function to return the profiles having either the top number of followers, where they may specify an "Account Category"—such as "LeftTroll" or "RightTroll" (Note: the Kaggle dataset came with these categorical values already having been classified).

The 3rd class is named "TweetVisualizer" and holds functionality that generates plots and graphics about the processed twitter content. Many of the methods allow the programmer to provide an "Account Category" to see the specific plot in a filtered view for that parameter. The current implementation allows the user to generate: a word cloud plot of either the most frequent hashtags or terms contained within the Tweet content; a horizontal bar chart illustrating the number of Tweets/Retweets by "Account Category"; a Venn diagram showing the number of similarly used "hash tags" between two "Account Categories"; and a donut chart that reflects the number of positive, negative and neutral Tweets (which can also be filtered by "Account Category" as well).

Relevant Results:

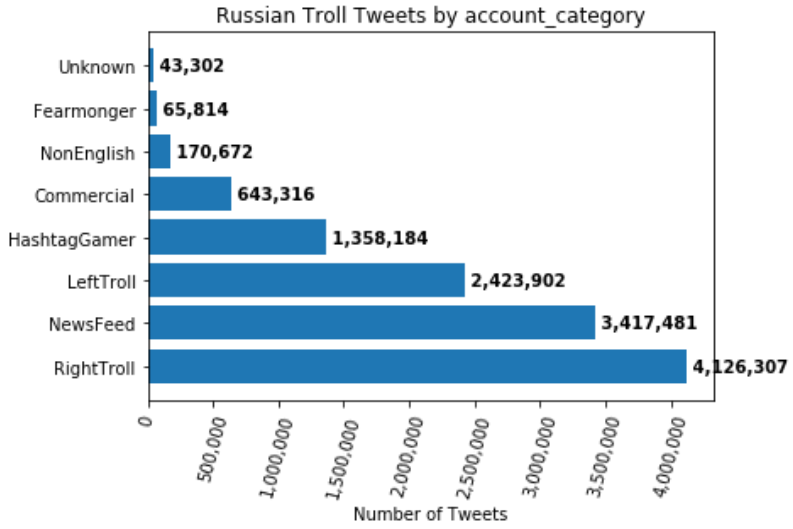
There were several interesting findings in utilizing the "TweetCalculator" class's methods. For one, if you notice below, the average "followers-to-following" ratio indicates that ,for each troll account, there are 1.53 followers for every person that they follow. I think this speaks to the success of their operations-- where they were able to efficiently build an audience for their influential messaging. You can also see that they have a large follower base for their messaging, with one profile bringing in as many 203,224 followers. That is a significant amount of people, and with precise targeting, could be an extremely effective way of generating impressions for the content they produce and share.

```
Russian Troll Tweets : Followers/Following
Followers Avg : 1394
Followers Median : 122
Following Avg : 910
Following Median : 192
Followers-to-following : 1.53

Top 10 followers - Overall

              author account_category  following  followers
14160      NOVOSTIMSK       NonEnglish       8798     203224
10586   KADIROVRUSSIA       NonEnglish          0      90648
14174      NOVOSTISPB       NonEnglish       2778      81161
18316     TODAYNYCITY         NewsFeed       9462      61038
10043     JENN_ABRAMS       RightTroll      12889      57462
7114       ELEVEN_GOP       RightTroll      30222      54664
18218  THEFOUNDINGSON       RightTroll      28824      41976
14137   NOVOSTICRIMEA       NonEnglish        998      38910
5681   COMRADZAMPOLIT       NonEnglish       1123      36509
19386  WASHINGTONLINE         NewsFeed      15458      31713
```

Another interesting finding came from a visual that illustrates the sheer number of Tweets (includes Re-Tweets) the troll accounts produce—and with the significant audience mentioned above—I believe this has a compounding effect in leveraging the medium to spread their messaging. The visual is broken down by the "Account Category" and shows that the "Right Troll" account created/shared as many as 4,126,307 tweets. Although, I find it most disturbing that the second-highest "News Feed" category came it at 3,417,418 tweets. This is particularly troubling given the fact that this category's inherent function was to provide "fake news", highlighting the widespread exposure of operation's misinformation tactics



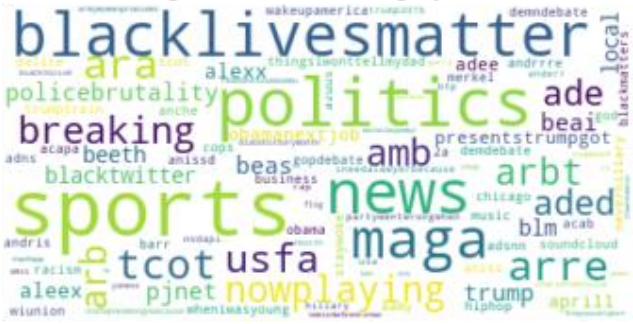
Furthermore, the "Hash Tag" and Tweet content wordclouds gives insight into how they engaged different users to sew division between different groups. the most frequent hash tags were either politically or socially oriented-- and used to induce provocation between different groups of people.

While looking at the “Hash Tag” word clouds, one thing that really stuck out to me was the emphasis placed on creating a tense environment around race-related topics. Contained within the 100 most significant hashtags for all categories were “blacklivesmatter”, “charlotesville”, and “blackmatters”. While digging deeper into the most frequent “Hash Tags” for each specific category, one can see that a comprehensive race-related focus for each type of account, with those categories also containing similarly oriented “Hash Tags” , such as : “policebrutality”, “islamkills”, “blackmattersus”. Some examples are provided below to illustrate this concept:

Hash Tag Word Cloud : Category - None

Hash Tag Word Cloud : Category - Commercial

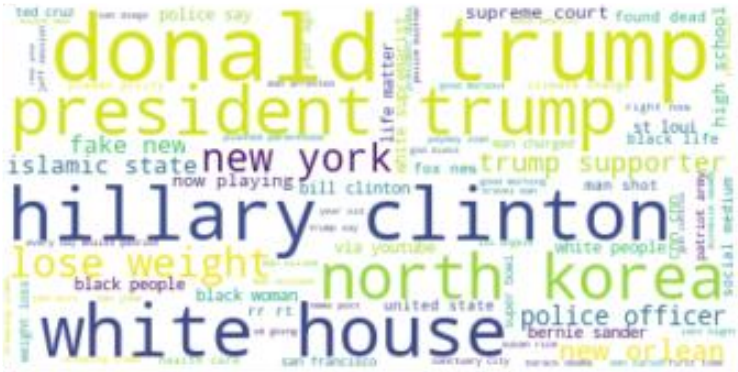
Hash Tag Word Cloud : Category - LeftTroll



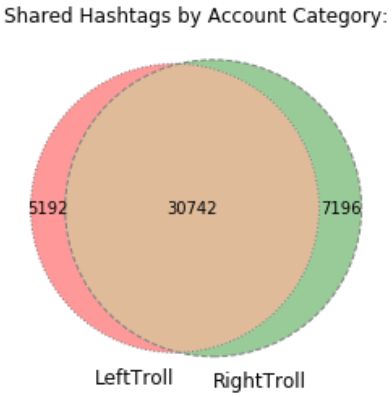
Hash Tag Word Cloud : Category - RightTroll

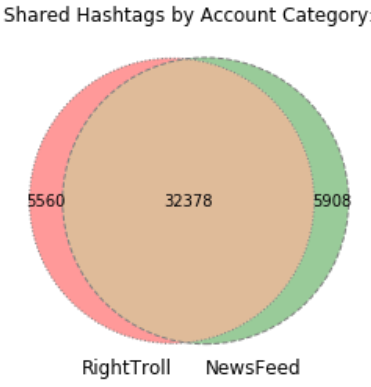
The word clouds highlighting the most significant terms indicates a very politically-centric focus, where some of the 100 most significant terms included “Donald Trump”, “President Trump”, “Islamic State”, “Hillary Clinton”, “North Korea, White House”, Trump Supporter”, “Supreme Court” , “Ted Cruz”, “and “Bill Clinton”. One can see from these terms themselves that they include political figures of both left and right ideologies, as well as significant points of discussion, which demonstrates their objective of polarizing different groups to induce division.

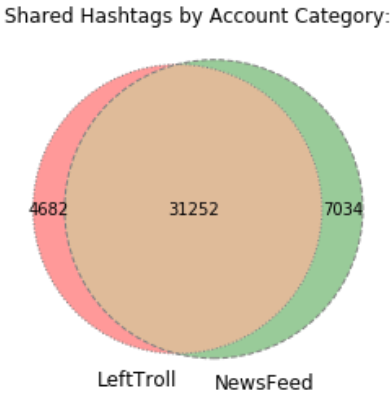
Tweet Content Cloud : Category - None



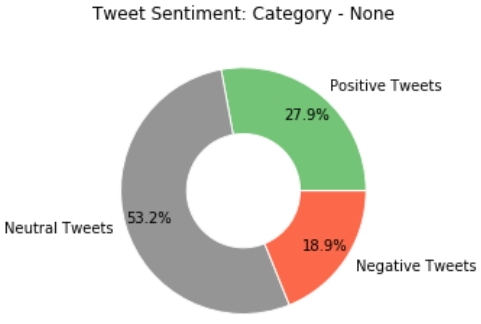
Furthermore, the Venn diagram visual reflects methods used to create common touch-points between groups to create tension, where “Hash Tags” were used to provoke sensitive and controversial discourse among them. The below Venn diagram indicates that a wide majority of “Hash Tags” were used in Tweets produced by both the “Left Troll” and “Right Troll” accounts. Using these common “Hash Tags” gives the operation a strategic ability to build conversation about certain topics from both perspectives and promote divisive conversation. Looking at the below you can derive that 71.28% of the “Hash Tags” used by the “Left Troll” and “Right Troll” accounts were common between the two types of accounts. The same notion extends to the relationship between the “News Feed” troll accounts and the “Left Troll” and “Right Troll” accounts, signifying resonance in the communicative messaging to this end as well.

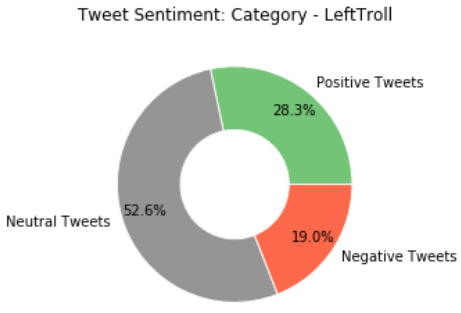


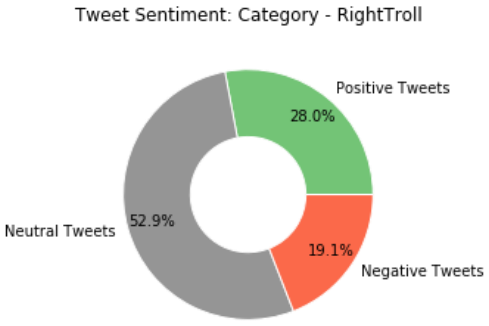




The donut charts containing the breakdown of “Positive”, “Neutral” and “Negative” sentiment for each “Account Category” yielded interesting but unexpected results. Tweets were classified as “Positive” if the TextBlob polarity score was above 0, classified as “Negative” if the score was below 0, and classified as “Neutral” if equal to 0. The most confusing aspect was that there was a consistent pattern that there were more “Positive” tweets than “Negative” tweets for each category. I found this specifically odd given the nature of their objective, to sew division and polarize groups, which I would assume to be inherently negative conversation. Although, I do think it’s possible that they may have sought to use their messaging to build emotion within each group of followers, by using positive sentiment, as a means of building a base of followers and polarizing their views in this way. I would like to explore another Sentiment analysis tool to see if the calculation would yield different results.







Project Notes:

- Canvas does not allow upload for files of the size of my dataset, therefore, my submission will only contain one of the files, the one containing the least amount of data. Because the single file only has a limited sample, it is expected that you would see very different results when running the program.
- Additionally, to run the notebook using the single file, you must update the below dictionary value to 1 in the "TweetDataHandler" class:

```
self.data_sources = {
            "troll_tweets" : [
                    ["\\data\\IRAhandle_tweets_",".csv",1],
                    ['external_author_id',
                     'author',
                     'content',
                     'region',
                     'language',
                     'publish_date',
                     'harvested_date',
                     'following',
                     'followers',
                     'updates',
                     'post_type',
                     'account_type',
                     'new_june_2018',
                     'retweet',
                     'account_category']]
            }
```

- Another consideration that must be made is that the program will retrieve the CSV file from the relative path of the program, at a folder titled "data". Therefore, a file directory with this name must be created at the location of the script, and contain the data file, in order to retrieve the data for processing. Another folder within the "data" folder itself must be created, and named "Processed", in order to save the processed file-- if the "isSavePreproc" configuration in the "TweetDataHandler" class is set to True.
- You may also clone the GitHub repository, containing all CSV files and folders, at the below link. The repository in the GitHub repository consists of several .PY files, which does differ from the format of the submitted IPython notebook file.
    - Link: https://github.com/CDT7185/trolly_tweedle
- The Kaggle dataset was retrieved at the below link:
    - Link: https://www.kaggle.com/fivethirtyeight/russian-troll-tweets