BAIT 508 GROUP PROJECT

SOCIAL MEDIA ANALYTICS

Jimmy(JinYe) Lu
18588392
Nelson Wong
31371131
Chang He
96628607

Table of Contents

Part 1. Data Collection

The word we picked for this project is 'NetNeutraility'.

We use the streaming method to collect all our 10,000 tweets since this key word is a heated topic among Americans in real time.

First, we import the TwythonStremer module from twython package and build our own subclass of TwythonStremaer called MyStreamer.

MyStreamer inherits the methods from TwythonStreamer and we override the methods on_success and on_error to fit our needs. On_success method will append tweets to the list when it is in English and will continue doing that until the number of tweets reaches 10,000.

```
# overriding
def on_success(self, data):
    # check if the received tweet dictionary is in English
    if 'lang' in data and data['lang'] == 'en':
        tweets.append(data)
        print('received tweet #', len(tweets), data['text'][:100])

    # if we have enough tweets, store it into JSON file and disconnect API
    if len(tweets) >= 10000:
        self.store_json()
        self.disconnect()
```

In addition, we define a new method store_json to store all the tweets as json file for future analysis.

```
# our new method to store tweets into JSON file
def store_json(self):
    with open('tweet_stream_{}_{}.json'.format(keyword, len(tweets)), 'w') as f:
        json.dump(tweets, f, indent=4)
```

Part 2. Preliminary Analysis

First, we load the file containing the 10,000 tweets collected into the workplace in order to analyze it.

According to initial analysis, the collected tweets data is a list containing 10,000 elements. Each element represents a tweet and it is structured as a dictionary which containing full information about the tweet.
In order to analyze the tweet content completely, we take a thorough look at the tweets. We find that there are two kinds of tweets. The first kind of tweet are long tweets that contain

additional key ‘extended_tweet’. And the full text of it is contained in this extended tweet. If the tweet is extended tweet we analyze the full text value in the ‘extended_tweet’. Another kind of tweets are the normal tweets which has limitation of 140 characters. Thus, for this kind of tweets we only need to analyze the text value in each element.

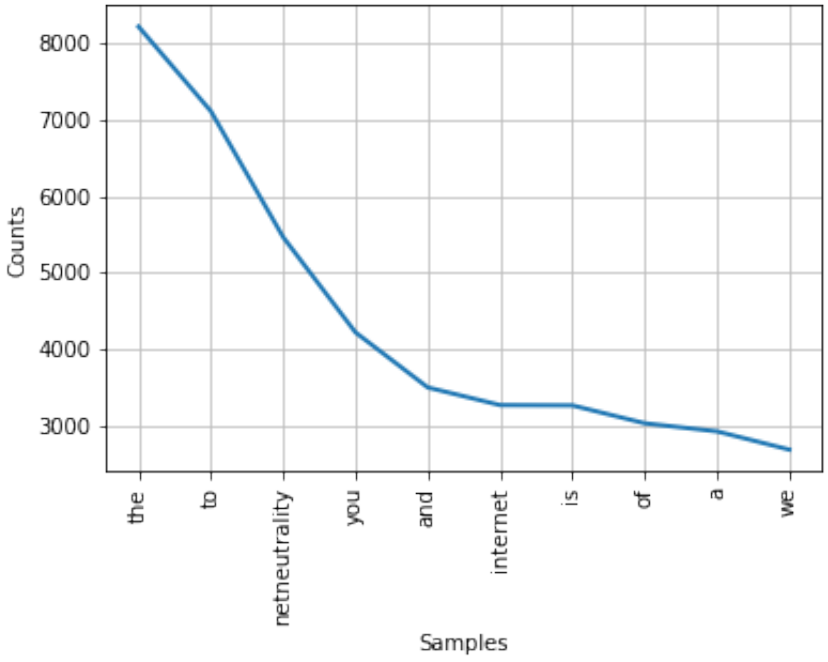
Also, in order to get the meaningful result, we filter the punctuation of the sentences and change the words into lower case.

```
p = string.punctuation
p+='…'
p+='—'
p+='–'
table_p = str.maketrans(p, len(p) * " ")
textdeal=text.translate(table_p).lower()
```

Finally, we can get the result of the ten most popular words in our tweets with nltk module.

The most popular words with stop words are:
‘the’, ‘to’, ‘netneutrality’, ‘you’, ‘and’, ‘internet’, ‘is’, ‘of’, ‘a’, ‘we’

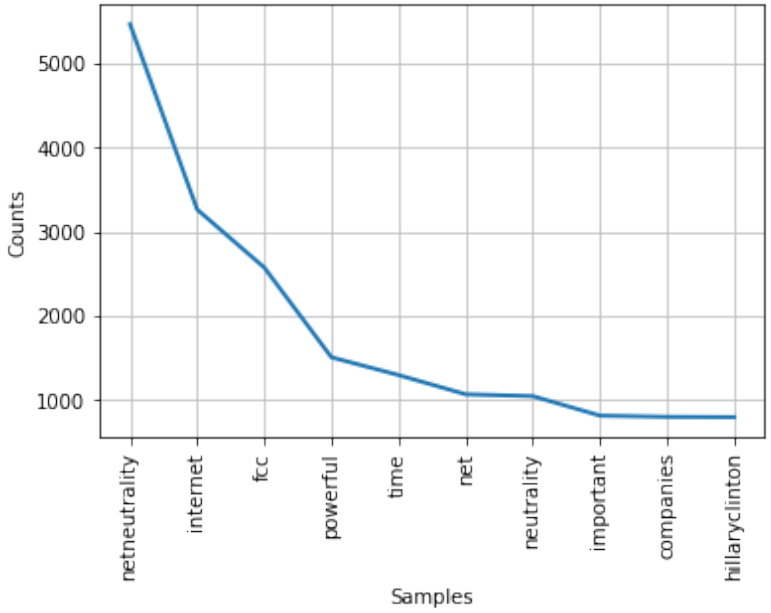
The graph below shows the popular extent of each top word in the tweets.



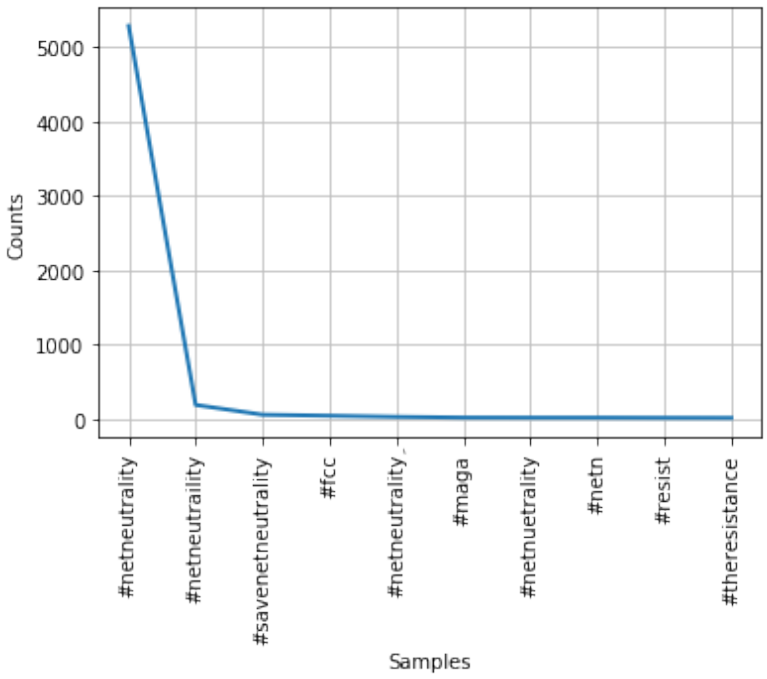
However, as shown above, many of the top words are stop words. Thus, we filter the stop words to get most popular meaningful words.

The most popular words without stop words in our tweets are:

‘netneutrality’, ‘internet’, ‘fcc’, ‘powerful’, ‘time’, ‘net’, ‘neutrality’, ‘important’, ‘companies’, ‘hillaryclinton’.
The graph shows the popular extent of each top word without stop words in the tweets:

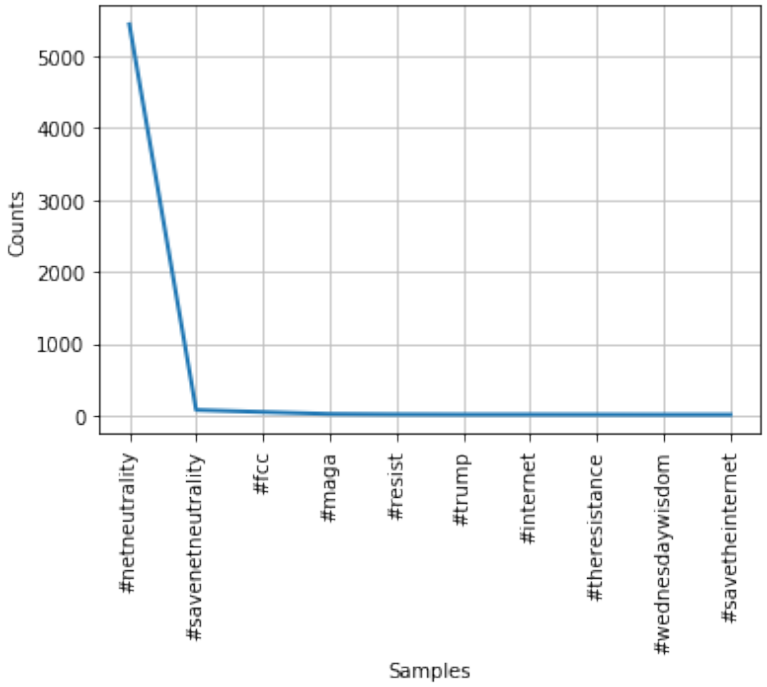


In order to get the most popular hashtags, we get the split words from the tweets and filter the words which the first character of it is equal to ‘#’ label. Also, note that since the ‘#’ label is used for filtering, we delete it from the default punctuation in order to keep this label in each word if the word has the ‘#’ label.
Using this method, we can get all the hashtags from the whole tweets.
The ten most popular hashtags are shown in the diagram below.



As shown above, there are some hot hashtags that are typos of 'netneutrality'. We filter these typo hashtags to obtain ten unique and meaningful hashtags that reflect the hot topics in tweets.

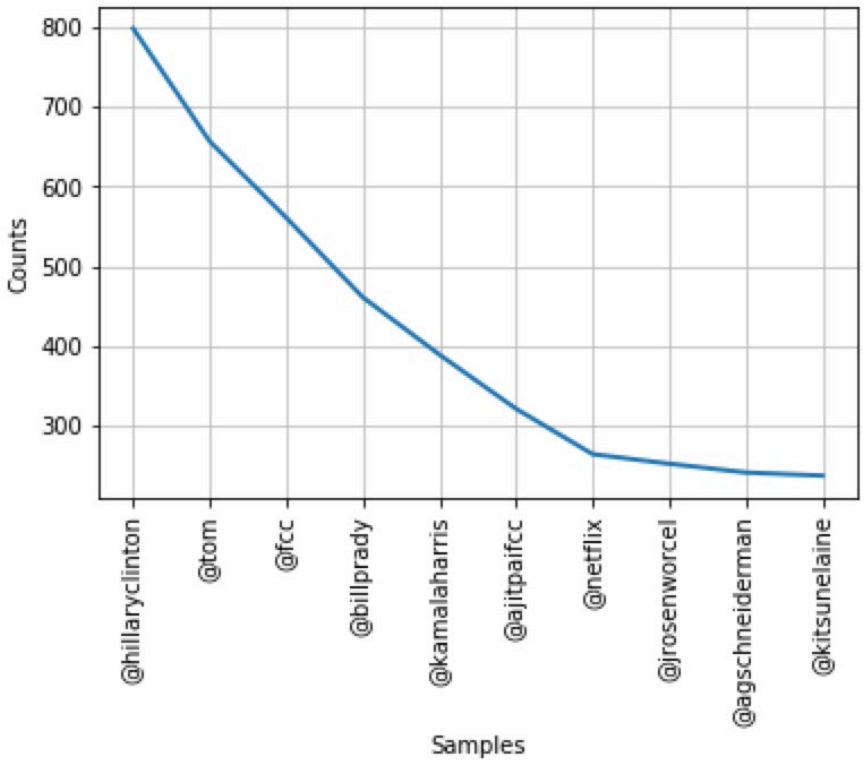
The most popular hashtags without typos are shown below.



Using the same logic dealing with the hashtags above, we filter the words which the first character of it is equal to '@' label. Also, note that since the '@' label is used for filtering, we

delete it from the default punctuation in order to keep this label in each word if the word has the '@' label.

The most frequently appearing usernames are shown in the diagram below.



We create a dictionary to store each person's tweets count. And we use the 'screen_name' of the user as the key because the screen name of every user is unique. Also, the value of the key is the number of tweets he/she sent.
Using this method, we obtained the number of each user's tweets. According to the dictionary we created, **the most frequently tweeting person** about the keyword is Greedsakiller. He/she has sent 27 tweets about the netneutrality.

The screenshot below shows the actual running result on Jupyter notebook:

```
In [9]: name=max(counter, key=counter.get)

In [10]: number=counter['Greedsakiller']

In [16]: print('The user '+name+' is the most frequently tweeting person, he has tweeted '
             +str(number)+' tweets.')

The user Greedsakiller is the most frequently tweeting person, he has tweeted 27 tweets.
```

To find **the most influential tweet**, we should find the retweet count, reply count and quote count of every tweets. After getting these data from each tweet's dictionary, we calculate the influence of each tweet and get the most influential tweet.

```
In [14]: max(influence)

Out[14]: 0
```

Unfortunately, as shown in the screenshot above, the maximum value of the influence is zero. It means that all of our tweets collected has zero influence. Because we collect the stream tweets, the tweets are the most updated one which the user posted right now. Therefore, all the tweets do not have any influence.
Thus, there is no most influential tweet in our sample.

Part 3. Word Cloud

We use the word cloud to visualize our data.
Based on the dataset we organized in the previous section, we further eliminate the meaningless words and clean up the data to help visualization.
When we compare different stemming methods, WordNetLemmatizer gives the best results as per below screenshot. For example, other stemming methods will change the word 'important' to 'import' while it actually means important here.

```
: for word in words:
    print('Word: {}\tLancaster: {}\tPorter: {}\tSnowball: {} \tLemma: {}'.format(
            word,
            ls.stem(word),
            ps.stem(word),
            ss.stem(word),
            wnl.lemmatize(word)
        ))
```

```
Word: netneutrality     Lancaster: netneut      Porter: netneutr        Snowball: netneutr      Lemma: netneutra
Word: rt        Lancaster: rt   Porter: rt      Snowball: rt    Lemma: rt
Word: hillaryclinton    Lancaster: hillaryclinton       Porter: hillaryclinton  Snowball: hillaryclinton
a: hillaryclinton
Word: you       Lancaster: you  Porter: you     Snowball: you   Lemma: you
Word: go        Lancaster: go   Porter: go      Snowball: go    Lemma: go
Word: girl      Lancaster: girl Porter: girl    Snowball: girl  Lemma: girl
Word: this      Lancaster: thi  Porter: thi     Snowball: this  Lemma: this
Word: important Lancaster: import       Porter: import  Snowball: import        Lemma: important
Word: costs     Lancaster: cost Porter: cost    Snowball: cost  Lemma: cost
Word: go        Lancaster: go   Porter: go      Snowball: go    Lemma: go
Word: amp       Lancaster: amp  Porter: amp     Snowball: amp   Lemma: amp
Word: powerful  Lancaster: pow  Porter: power   Snowball: power         Lemma: powerful
Word: companies Lancaster: company      Porter: compani Snowball: compani       Lemma: company
Word: get       Lancaster: get  Porter: get     Snowball: get   Lemma: get
Word: powerful  Lancaster: pow  Porter: power   Snowball: power         Lemma: powerful
Word: we        Lancaster: we   Porter: we      Snowball: we    Lemma: we
Word: can       Lancaster: can  Porter: can     Snowball: can   Lemma: can
```

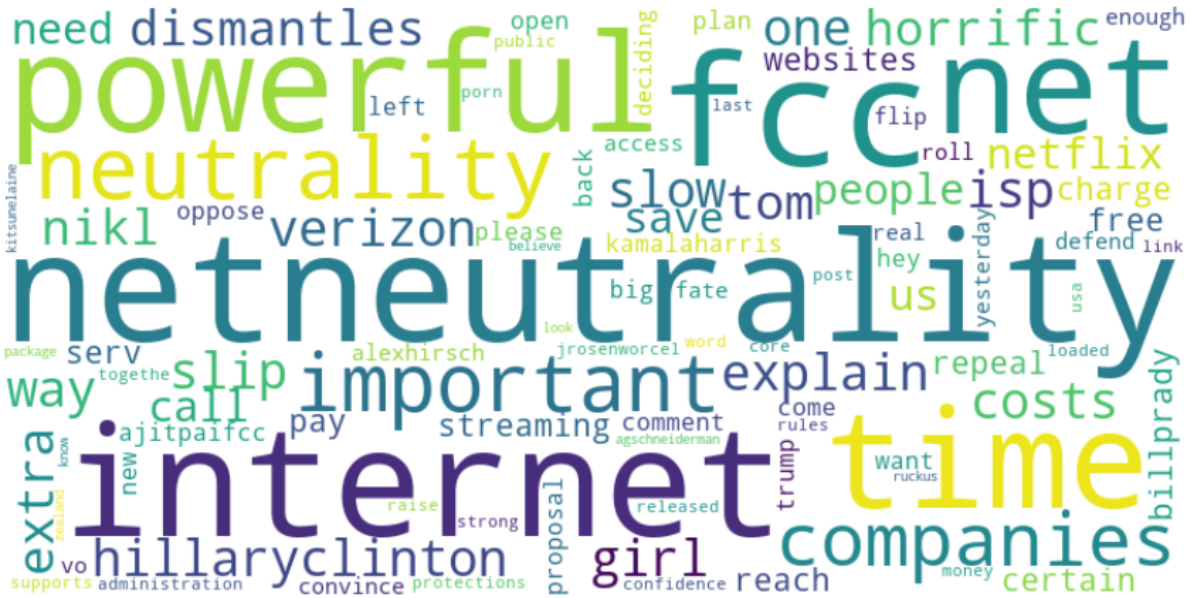
To make the work cloud, firstly, we need to convert our list of words into string.

```
# convert list to string
file1 =''
for word in st:
    file1 += ' {}'.format(word)
```

Further, we add below words that don't have any meaning to our stop words list.

```
stopwords = nltk.corpus.stopwords.words('english')
stopwords.append('go')
stopwords.append('amp')
stopwords.append('let')
stopwords.append('https')
stopwords.append('t')
stopwords.append('co')
stopwords.append('rt')
stopwords.append('without')
```

Last, import the word cloud module, generate the image and display the image as per our preference.
The result is as below:



Part 4. Sentiment Analysis

Sentiment analysis was done in two forms, once including URLs in the analyzed text and once without. All results are robust to removing URLs, hashtags and even net neutrality. For each iteration of sentiment analysis, the polarity and subjectivity of each tweet is evaluated and

stored in a list. The below histograms show the frequency of each bin of polarity and subjectivity and their respective tweet frequencies. Each iteration details the polarity and subjectivity analysis for the sum of all tweets, all extended tweets and regular tweets separately.

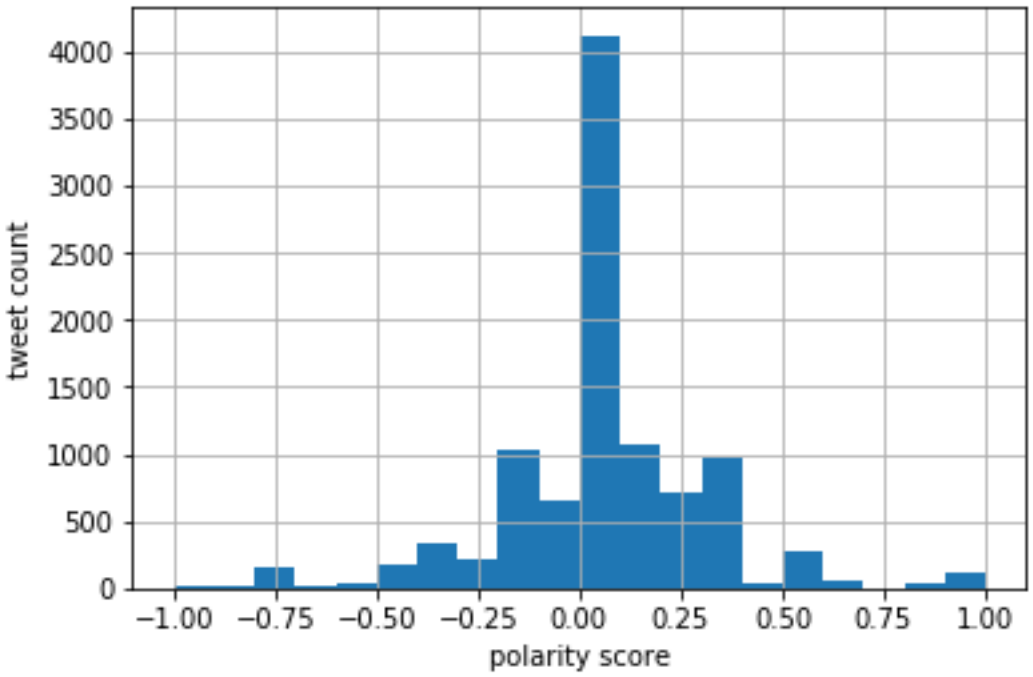
The procedure for sentiment analysis was as follows. For each dictionary in the list of tweets, the dictionary is checked if the key 'extended_tweet' exists. If so, the 'full_text' in 'extended_tweet' is extracted from the tweet and analyzed. Otherwise, the 'text' key value pair will be extracted and analyzed from the dictionary. This is done because every tweet has a 'text' key value pair, but only extended tweets have an 'extended_tweet' key. An extended tweet's 'text' key value pair would only show a part of the actual tweet, cutting off the full tweet with an ellipsis. Therefore, to extract the full text from every tweet, text must be extracted from the 'extended_tweet' key for extended tweets and the 'text' key for standard tweets. The full text of each tweet is analyzed using TextBlob's polarity and subjectivity methods.

Polarity Without URLs:

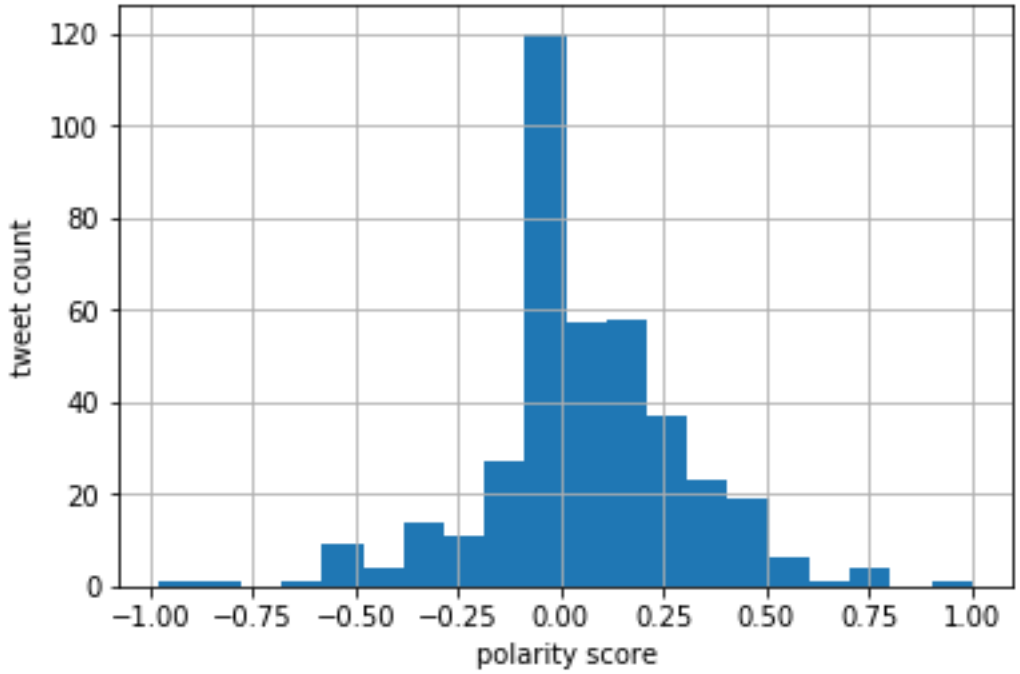
All Tweets:

Mean Polarity = 0.04650807188305188

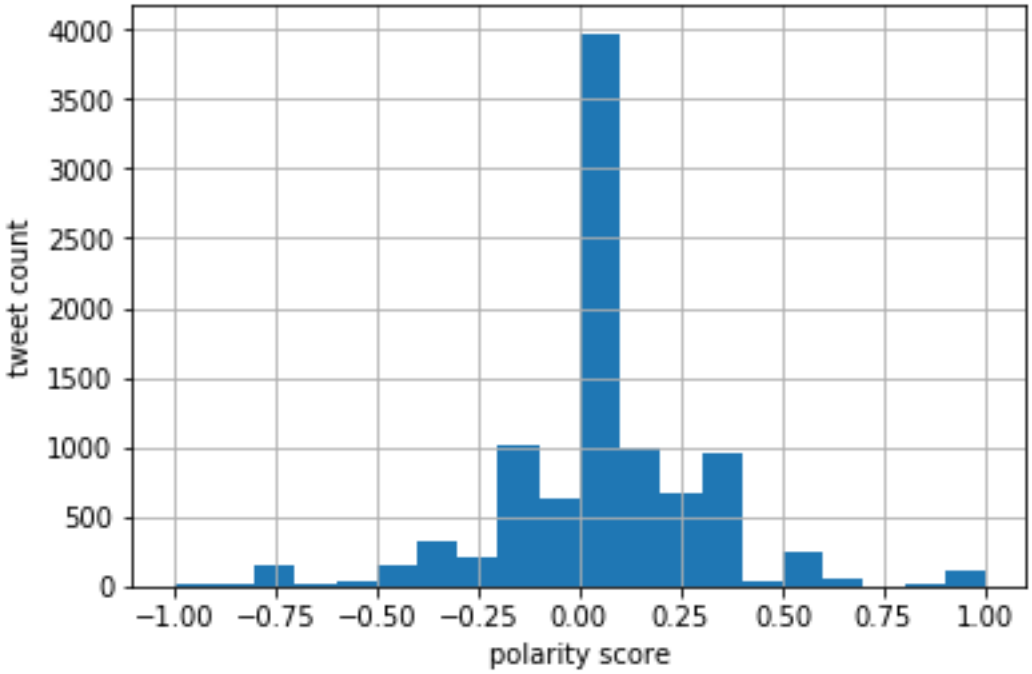
Median Polarity = 0.0



Extended Tweet:
Mean Polarity = 0.07118033461149224
Median Polarity = 0.03750000000000001



Normal Tweet:
Mean Polarity = 0.045496113574182456
Median Polarity = 0.0

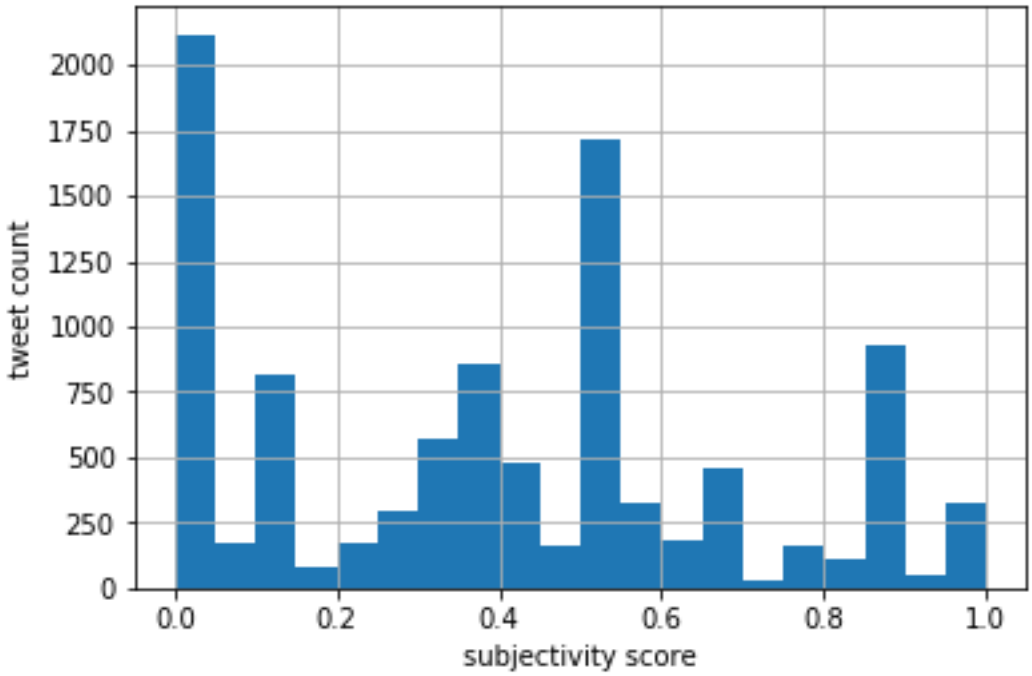


Subjectivity Without URLs:

All Tweets:

Mean Subjectivity = 0.3890913940082074

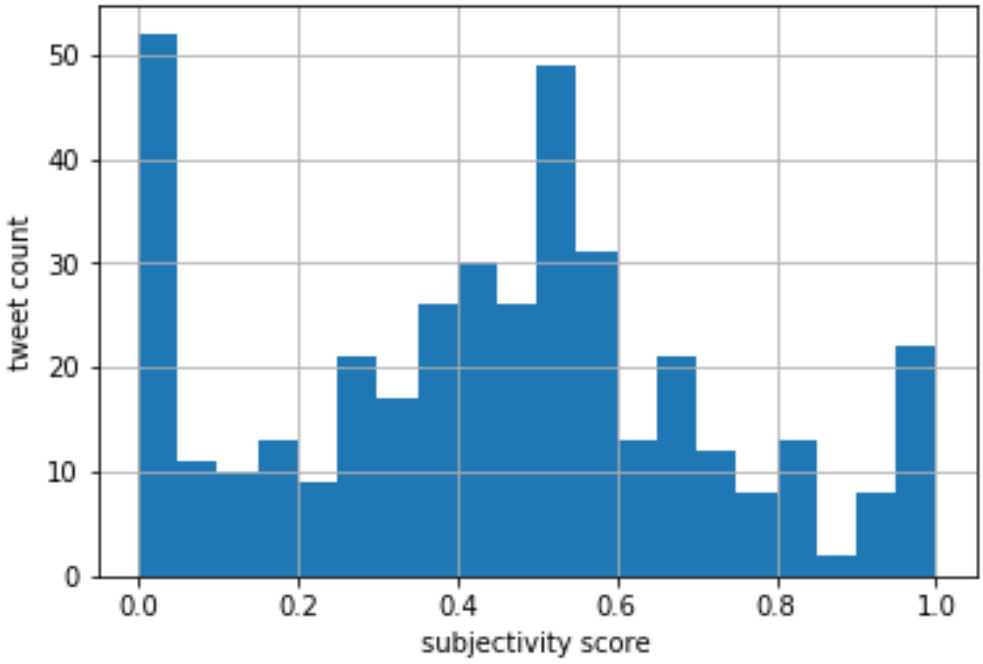
Median Subjectivity = 0.39999999999999997



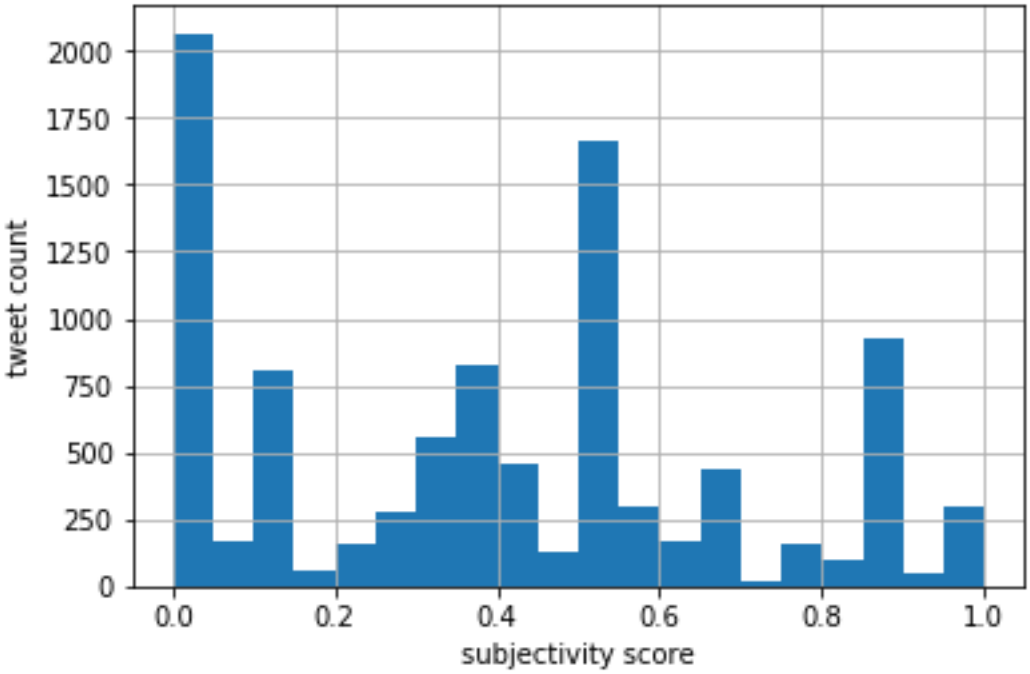
Extended Tweet:

Mean Subjectivity = 0.43926574974641963

Median Subjectivity = 0.4601190476190476

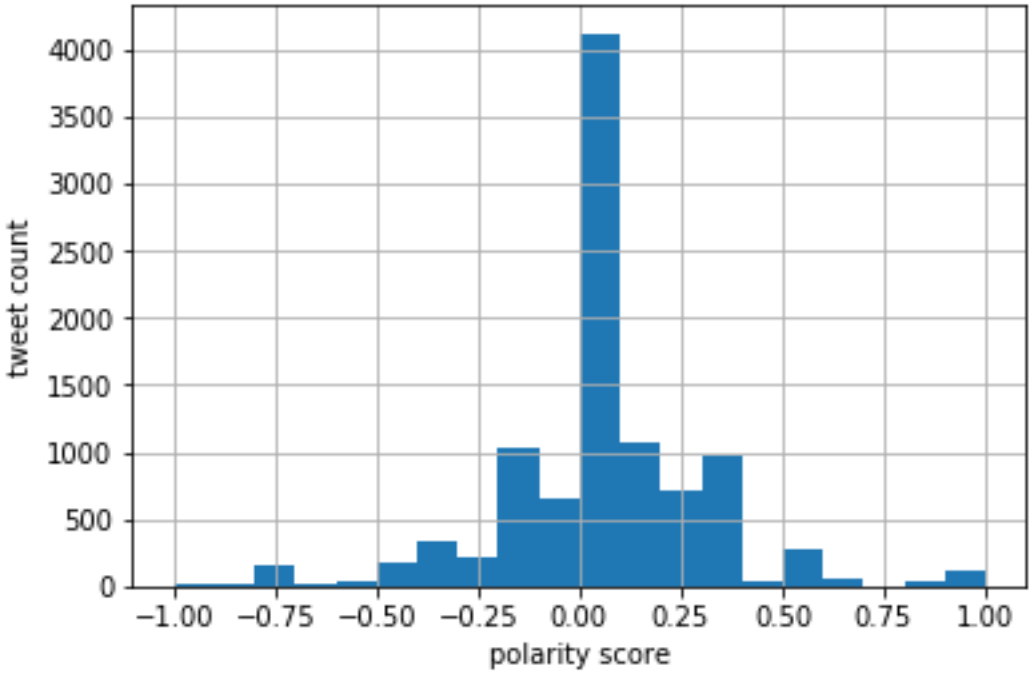


Normal Tweet:
Mean Subjectivity = 0.38703344104538673
Median Subjectivity = 0.39999999999999997

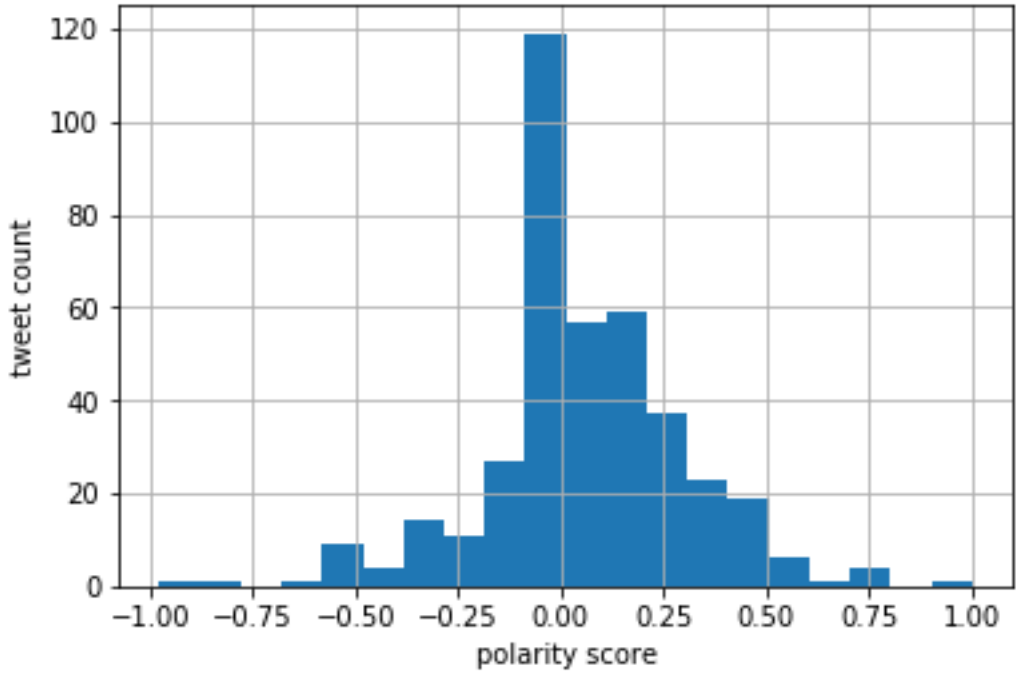


Polarity with URLs:

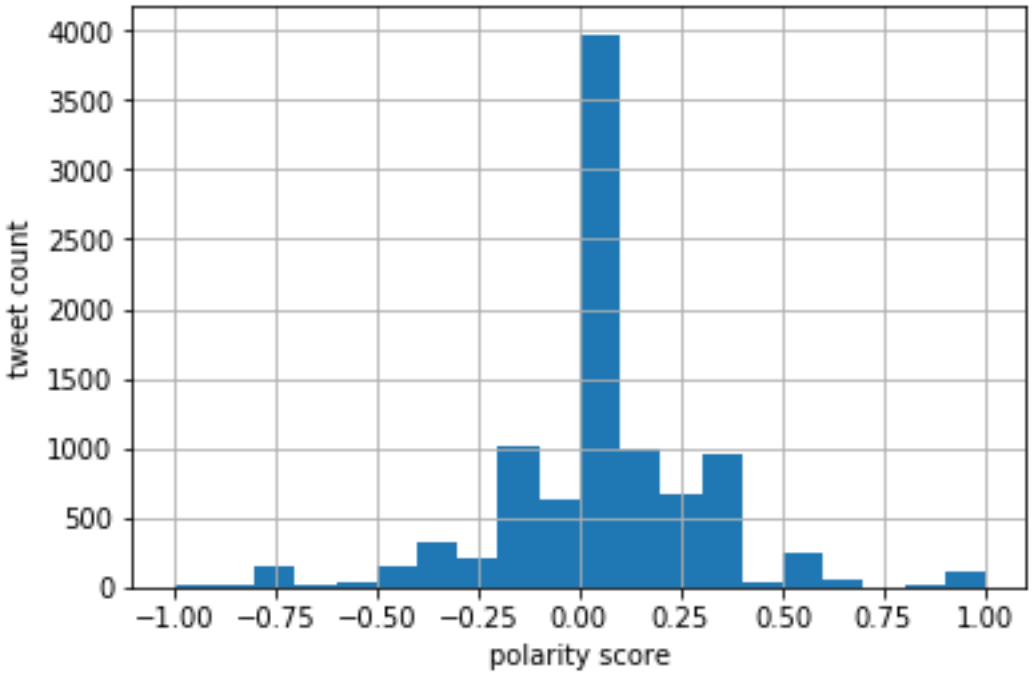
All Tweets:
Mean Polarity = 0.04661880849019532
Median Polarity = 0.0



Extended Tweet:
Mean Polarity = 0.07340926711088797
Median Polarity = 0.042072510822510824



Normal Tweet:
Mean Polarity = 0.04551997019157436
Median Polarity = 0.0

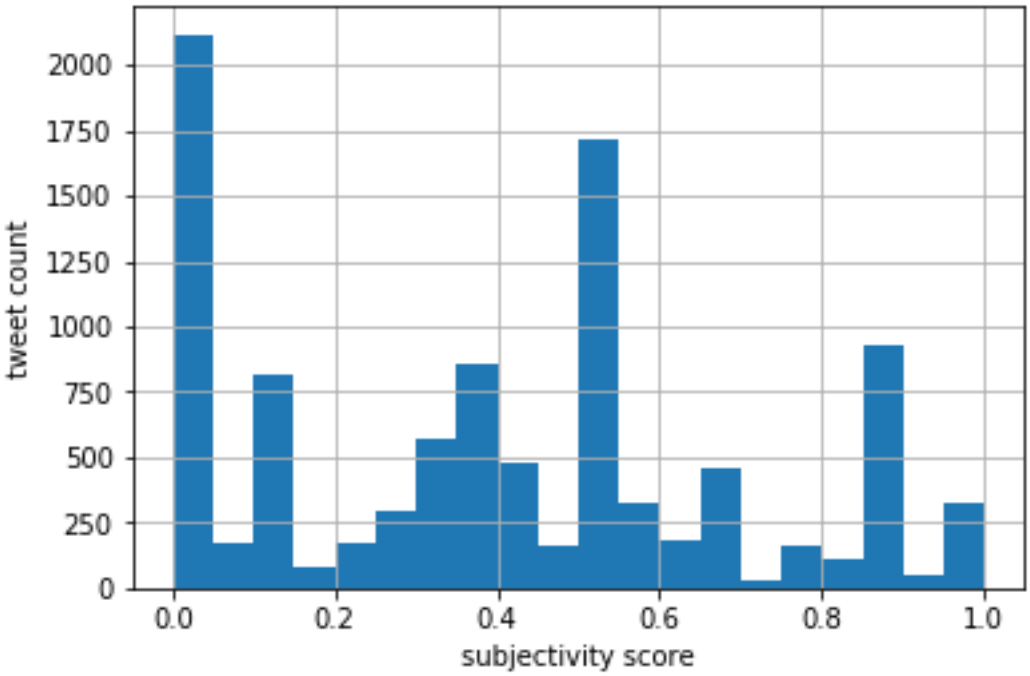


Subjectivity Without URLs:

All Tweets:

Mean Subjectivity = 0.38927556067487407

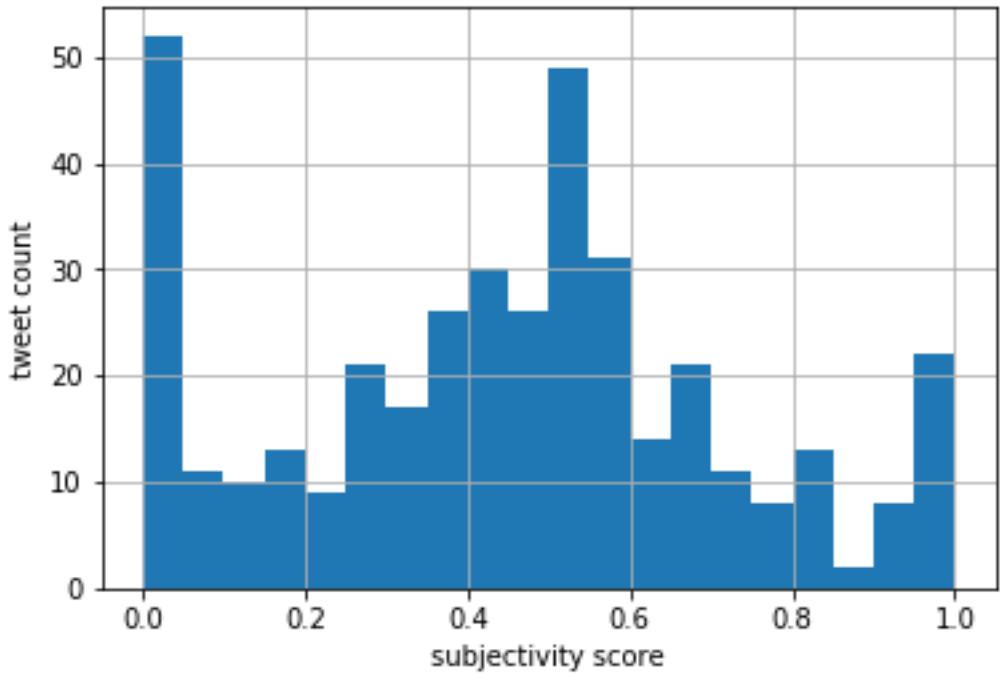
Median Subjectivity = 0.39999999999999997



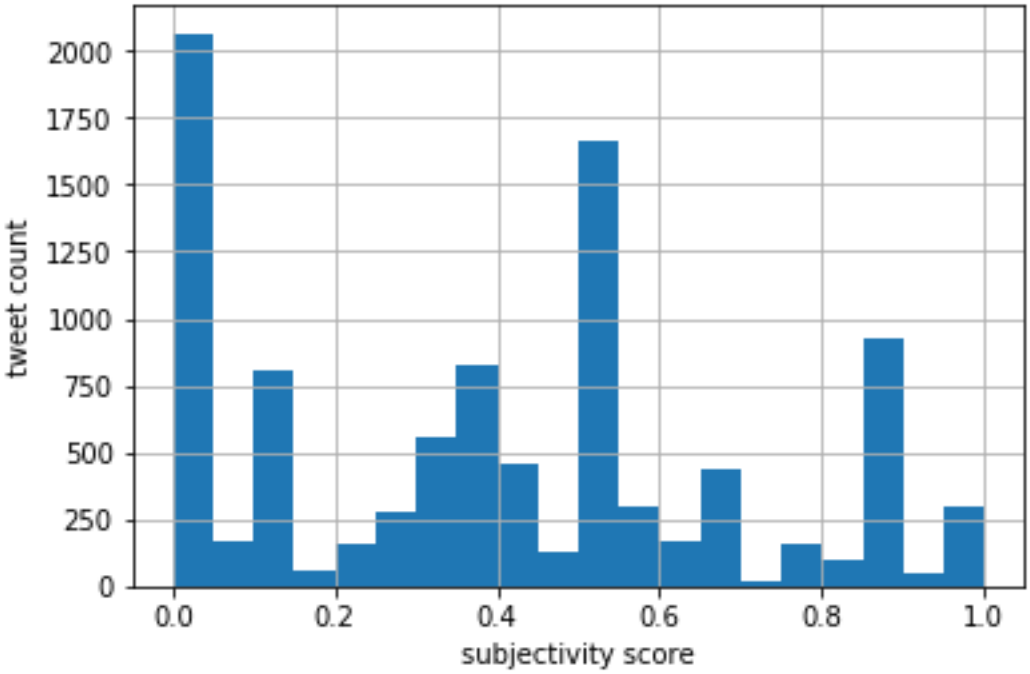
Extended Tweet:

Mean Subjectivity = 0.4408097429782301

Median Subjectivity = 0.4601190476190476



Normal Tweet:
Mean Subjectivity = 0.38716183302262314
Median Subjectivity = 0.39999999999999997



Part 5. Insights

The frequency of words in the tweets leads us to 'Hillary Clinton', 'Internet' and 'FCC'.

Based on our research, the Federal Communications Commission announced on Nov 21, 2017 that it planned to wipe out the net neutrality regulation that ensure equal access to the internet.[i] This was proposed by FCC Chairman Ajit Pai, which is one of our most @person. This announcement became a huge topic on social media as most people were condemning this action. People believe the action to kill net neutrality is against the freedom of speech and powerful companies could benefit substantially from this. All the people on the list of most frequently appearing usernames, including Netflix, have expressed support for net neutrality, except for Ajit Pai. Twitter users were retweeting the supportive tweets, and hence their usernames have high appearance frequencies. Ajit Pai, on the other hand, is being tweeted at by requests to maintain net neutrality and expressions of disapproval.

Values for polarity tend to be centered around 0. This is surprising because an overwhelming amount of news regarding net neutrality conveys hate towards the Federal Communications Commission and their decision to repeal net neutrality. Celebrities and social media sites, such as Hillary Clinton and Reddit, called for action to stop the FCC from repealing net neutrality. Therefore, polarity was expected to be strongly negative.

The first tweet in the list of tweets appears to be strongly negative. It expresses strongly worded malcontent towards the government and details ways the government ruined quality of life. It also calls for action against the FCC. As a result, a human interpreting the tweet would see it as a strongly negative message.

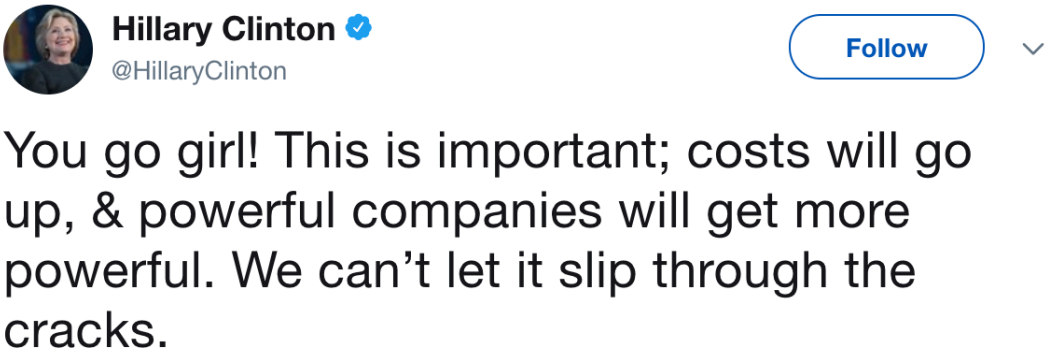
```
Fucking government bastards... Stop them from censoring us again! They ruined TV, Movies, Music, and most recently Video
Games. And the internet is in their scope sights now...

Support #NetNeutraility and stand up. Enough is enough! https://t.co/cKeLio7kPJ
```

However, TextBlob's analysis only assigned the tweet a polarity level of -0.05, a very neutral rating. Consequently, measurement error likely exists in TextBlob's algorithm such that the distribution of polarity scores is shifted to the right: thereby providing a very neutral distribution of messages when it is expected to be negative.

Subjectivity scores, on the other hand, are more agreeable under TextBlob. Subjectivity evaluates the extent to which the message is opinionated or emotionally driven, rather than by objective purposes. The aforementioned tweet had a subjectivity score of 0.51. This is a fair evaluation because the tweet includes a lot of opinionated and emotional content even though it has an objective call to action. The distribution for subjectivity scores appear to be trimodal. One subset of tweets being highly objective, one subset clustered around a score of 0.5 and one subset clustered between 0.8 and 1. The distribution of tweets could be clustered into three distributions, and the text for each distribution evaluated independently. Consequently, the subsets of tweets with more subjectivity may be more negative in polarity.

By construct, net neutrality, internet and their variants must be among the list of most frequently tweeted words. The keyword used for the twitter search was netneutrality. Therefore, each tweet must include the term netneutrality at least once. Since net neutrality only pertains to the internet, internet must also be amongst the most frequent words. Internet is the object that net neutrality refers to and is thus not very informative in our list.

Past the top word frequencies of netneutrality, net, neutrality and internet, many of the most frequently tweeted words belong to Hillary Clinton's tweet encouraging the defence of net neutrality.



This also explains why Hillary Clinton is the most frequently appearing username.

Hillary Clinton's original message offered support by encouraging defenders of net neutrality, therefore having an overall positive tone. TextBlob assigned her message a polarity value of 0.375, and a subjectivity level of 0.875. Since TextBlob evaluates Hillary Clinton's message each time it is retweeted, the distribution of polarity is thus skewed to the right because many people conveyed their discontent through Hillary Clinton rather than scorn in their own words. This also explains the cluster of tweets with high subjectivity scores. However, this does not explain the relatively neutral polarity for the mentioned tweet above.

In conclusion, by doing basic text analysis on topics we are interested in in twitter, we could briefly understand key elements of the topic, the impacts of tweets, and people's opinion towards it. In other words, with proper interpretation, we could use this technique and information for decision making and prediction, especially for topics, like stock price and fashion, where social media opinion plays an important role.

---

[i] Restoring Internet Freedom, retrieved from https://www.fcc.gov/restoring-internet-freedom, accessed on Dec 2, 2017