1. Collecting and Storing tweets (Use tweet\_extract.py for this)
2. A new twitter application is made on https://apps.twitter.com after logging into the twitter account.
3. All the four parameters such as secret key, access token, etc are put into the code already provided. We used the streaming class which collects live tweets.
4. This code is run three times each for Hillary, Trump and Obama keywords to get 10000 tweets about each person.
5. The generated json files are then merged to get 3 json files for each person.
6. As json file is a list of dictionaries, each dictionary will be a tweet. Since we need only the “text” part of the tweets, we extracted it and stored it in 3 separate json files where each each item in a json file is a string i.e. the tweet.
7. Sentiment Analysis (Use tweet\_analysis.py for this)
8. The list of tweets is iterated over using a for loop and using TextBlob package the polarity and subjectivity scores are calculated and stored in 2 different lists.
9. Average of both the lists is calculated to get the aggregate scores which is shown in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Hillary | Obama | Trump |
| Polarity | 0.0698148080596 | 0.015709359005 | 0.074011094016 |
| Subjectivity | 0.342438902271 | 0.334487920914 | 0.309952640023 |

1. Trump has the most positive comments and Obama has the most negative comments.
2. Creating Word Cloud:
3. The package nltk was installed using nltk.download() command.
4. Using for loop each tweet is iterated over and converted to UTF-8 encoding.
5. Now the punctuation and numbers are filtered out from the string to get a better word cloud.
6. This string is split into a list words and each common word between the list generated by splitting the string and a list “stopwords” which is a part of nltk package, is removed.
7. The remaining words are appended to a new string, this string is fed into the WordCloud method by creating a new object.
8. Also we will use matplotlib package to generate and edit the desired word cloud.
9. Filtering with location (use tweets\_location.py for this)
10. We used the same streaming class used earlier to do this.
11. Using Google Earth, the coordinates for each of 7 given cities are generated and plug into the filter(last line in the code) and also the keyword is changed to the city name so we get accurate results from that specific location.
12. We get 7 json files with 1000 tweets each.
13. The same process mentioned above i.e. sentiment analysis and word cloud is done on these tweet files and results are generated and results are given below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Polarity | Hillary | Obama | Trump | Most positive | Most negative |
| Dallas | 0 | 0.1875 | -0.198800075586 | Obama | Trump |
| New York | 0.125 | 0 | 0.0590489718615 | Hillary | Obama |
| San Francisco | 0.211111111111 | -0.0791666666667 | 0.0562801932367 | Hillary | Obama |
| Atlanta | 0 | -0.833333333333 | 0.0700757575758 | Trump | Obama |
| Chicago | 0.392857142857 | -0.0964285714286 | -0.0375 | Hillary | Obama |
| Washington DC | 0.211111111111 | -0.175 | 0.114701704545 | Hillary | Obama |
| Los Angeles | 0.392857142857 | -0.075 | 0.119453463203 | Hillary | Obama |

|  |  |  |  |
| --- | --- | --- | --- |
| Subjectivity | Hillary | Obama | Trump |
| Dallas | 0 | 0.5 | 0.419538926682 |
| New York | 0.5 | 0 | 0.29567674513 |
| San Francisco | 0.551543209877 | 0.198611111111 | 0.34044168392 |
| Atlanta | 0 | 0.833333333333 | 0.297348484848 |
| Chicago | 0.517857142857 | 0.728571428571 | 0.184722222222 |
| Washington DC | 0.551543209877 | 0.3 | 0.377690972222 |
| Los Angeles | 0.517857142857 | 0.325 | 0.31369047619 |

Note: Since the tweets generated with location filter are not exactly related to Elections or Hillary, Obama and Trump, the word clouds are generating random and irrelevant words. Thus not added word cloud images for this one. The code however generates word clouds.

1. Topic Modelling.
2. The 30000 tweets collected earlier are stored in 3 separate files each for Trump, Hillary and Obama which is the corpus.
3. Used generate\_corpus.py to create three text files with tweets in it.
4. The files NMF.py and LDA.py generate the required topics for the given three files.