**Readme for w205 Group Project Twitter Sentiment Analysis**

**By Manish Sannat and William OldFather**

**Prerequisites**:

1. Python 2.7 with Jupyter Notebook.
2. Mongo DB is installed in the box. (Run Notebook cell: “Data Storage: Mongo DB and JSON” and see it can connect to Mongo DB without any exceptions)
3. Following modules are installed:

Tweepy

PyMongo

Wordcloud

1. Get Twitter credentials and enter into file “config.py” (inside folder : “TwitterStream”) and credentials.py (inside folder : “alchemyapi-recipes-twitter”)
2. Get IBM Alchemy Free Trial key (“alchemy\_apikey”) and enter into credentials.py (inside folder : “alchemyapi-recipes-twitter”)

**Instructions**

1. **Start Mongo DB daemon**

Go to folder “alchemyapi-recipes-twitter”

Run in terminal

mongod --dbpath data/db --smallfiles --quiet &

1. **Twitter Data Extraction:**

Run batch program “twitter\_stream\_download\_geo.py” (folder : “TwitterStream”)

with following parameters for 60 minutes each:

python twitter\_stream\_download\_geo.py -q trump -l CA

python twitter\_stream\_download\_geo.py -q trump -l TX

python twitter\_stream\_download\_geo.py -q hillary -l CA

python twitter\_stream\_download\_geo.py -q hillary -l TX

1. **Data Storage**: Mongo DB and JSON: run the notebook cell to check that in the earlier step four collections are created in Mongo DB and number of tweets in each collection.

trump\_CA

trump\_TX

hillary\_CA

Hillary\_TX

Also, check JSON folder for presence of these four JSON files with the same names.

1. **Export JSON files**: Since we are not generating Big DATA for this project, to make the data analysis easier, we are creating JSON files at the same time we store the data to the Mongo DB. So, there is no need to export JSON file at this stage.

We expect to generate around 500MB of data for each run (total of 2 GB of data).

If needed, Mongo DB utilities can be used to extract the BULK data out and convert BSON to JSON format by using following commands in a terminal:

To dump data out of Mongo DB:

mongodump -d twitter\_db -o './TwitterStream/dump'

To convert BSON to JSON:

bsondump trump\_CA.bson >trump\_CA.json

1. **Pre-processing Tweets- Extract, Parse and Transform Tweets:** run this notebook cell to remove the stop words and tokenize the tweets.
2. **Trending I : Topic Modeling (most frequent terms and hashtags):** run the notebook cell . Change json file name in the following variable:

fname = './json/Trump\_sample.json' and run for all the four JSON files generated earlier.

1. For each JSON file in step 6 above, run the following notebook cell and check the output:

* Visualization - most frequent terms using Node.js

Go to html folder and double click on html file “freq\_terms\_chart.html”. It should show the most frequent terms bar chart.

Note: if you get any error with the html. Please start python web server in the Html folder for an unoccupied port (for example below port 8800).

python -m SimpleHTTPServer 8800 # Python 2

Go to this URL:

<http://localhost:8800/freq_terms_chart.html>

* Trending II : Word Cloud (both classic and with mask)

Notebook will show the workcloud upon run. The image files also get stored to the location : ./wordcloud

1. For following notebook cells:

* **Search a specific term's most frequent occurrences with other terms**
* **The 20 most frequent terms occurring together**

Change the json file name in the following variable:

fname = './json/Trump\_sample.json' and run for all the four JSON files generated earlier in step 2.

1. **Trending III : Hashtags Time Series:**

Change the file name in the variable below and point to the JSON file that you want to run. fname = './json/stream\_Trump\_big.json'

Also change the trending hashtags with the results from cell above:

Trending I : Topic Modelling (most frequent terms and hashtags)

#trending\_hashtags\_1 = '#trump'

#trending\_hashtags\_2 = '#hillarycoverageiscrap'

#trending\_hashtags\_3 = '#nevertrump'

Also, add key name similar to hashtags without starting with '#' in the dictionary match\_data. For an example, three hashtags: nevertrump, maga and job are used as key in the dictionary

#match\_data = dict(nevertrump=per\_minute\_i, maga=per\_minute\_s, job=per\_minute\_e)

#Plan to make this dynamic in the future release.

Run the cell and then open html file “time\_series\_all.html” inside the folder “html”.

Note: if you get any error with the html. Please start python web server in the Html folder for an unoccupied port.

python -m SimpleHTTPServer 8800 # Python 2

Go to this URL:

<http://localhost:8800/time_series_all.html>

1. **To run Twitter Sentiment Analysis using IBM Alchemy API:**

Go to folder "alchemyapi-recipes-twitter"

Run recipe.py on a terminal with two arguments.

python alchemy\_sentiment\_analysis.py 'trump\_TX.json' 250

python alchemy\_sentiment\_analysis.py 'trump\_CA.json' 250

python alchemy\_sentiment\_analysis.py 'hillary\_TX.json' 250

python alchemy\_sentiment\_analysis.py 'Hillary\_CA.json' 250

First argument parameter is the JSON file stored in the JSON folder and was generated from the earlier run (there should be four files in the JSON folder).

Second argument parameter is number of tweets for sentiment analysis.

IBM Alchemy API free account allows for free 1000 call each day.

This can be divided into four JSON files with each 250 calls.

The program will run Sentiment Analysis and store the tweets with coordinates and sentiment score to the Mongo DB collection <<sentiment\_name of JSON file>> without extension .json.

This will also create JSON file <<sentiment\_ plus name of input JSOB file>>in the folder “map”.

With each run enter the candidates positive and negative score in the file:

“alchemyapi-recipes-twitter/results.csv”.

Here is one example for Trump in CA batch program run.

###############

# Stats #

###############

SENTIMENT BREAKDOWN

Number (%) of positive tweets: 163 (67.63%)

Number (%) of negative tweets: 45 (18.67%)

Number (%) of neutral tweets: 33 (13.69%)

Corresponding entry in the file “alchemyapi-recipes-twitter/results.csv”.

|  |  |  |  |
| --- | --- | --- | --- |
| Trump - CA | 67.63 | 18.67 | 13.69 |

Note: There is possibility that the program may throw following error:

Traceback (most recent call last):

File "alchemy\_sentiment\_analysis.py", line 359, in <module>

main(sys.argv[1], int(sys.argv[2]))

File "alchemy\_sentiment\_analysis.py ", line 38, in main

unique\_tweets = dedup(raw\_tweets)

File "alchemy\_sentiment\_analysis.py", line 241, in dedup

for tweet in tweets:

TypeError: 'NoneType' object is not iterable

This is most likely due to lack of tweets in the file that contains GEO coordinates information.

Please run the tweepy batch program again (Step# 2) and collect twitter for a longer duration and then run the program alchemy\_sentiment\_analysis.py with the larger file.

1. **Extract Sentiment score JSON files from Mongo DB for Sentiment Map.**

Run following scripts in a terminal (with Mongo DB daemon running in the background)

To dump data out of Mongo DB, run following from a terminal

mongodump -d twitter\_db -o './dump'

cd dump/

cd twitter\_db/

To convert BSON to JSON, run following from a terminal (location where Mongo DB is dumped in above step) for the. bson files starting with prefix ‘sentiment\_’

bsondump sentiment\_trump\_CA.bson >sentiment\_trump\_CA.json

bsondump sentiment\_trump\_TX.bson >sentiment\_trump\_TX.json

bsondump sentiment\_hillary\_CA.bson >sentiment\_hillary\_CA.json

bsondump sentiment\_hillary\_TX.bson >sentiment\_hillary\_TX.json

Store these four JSON files to folder "**map**"

1. Generate Geo JSON files for positive and negative sentiments:

Run the notebook cell for the four JSON files created in the step 11 above.

This means this notebook cell has to be run four times, each with different input JSON and output GEO JSON files (for both positive and negative sentiments).

Change the JSON file name in the variable ‘fname’.

See commented variables in the notebook cell.

Uncomment one variable and run the cell (remember to uncomment the corresponding output GEO json file names as well.)

For example, to run for Trump in Texas uncomment following lines:

fname = './map/sentiment\_trump\_CA.json'

output\_geo\_json\_positive = './map/geo\_data\_trump\_texas\_positive\_new.json'

output\_geo\_json\_negative = './map/geo\_data\_trump\_texas\_negative\_new.json'

This will generate 8 geo JSON files in the map folder.

geo\_data\_hillary\_california\_positive\_new.json

geo\_data\_hillary\_california\_negative\_new.json

geo\_data\_hillary\_texas\_positive\_new.json

geo\_data\_hillary\_texas\_negative\_new.json

geo\_data\_trump\_california\_positive\_new.json

geo\_data\_trump\_california\_negative\_new.json

geo\_data\_trump\_texas\_positive\_new.json

geo\_data\_trump\_texas\_negative\_new.json

1. **Twitter Sentiment Map:**

Double click the html file : TweetMap.html.

It should a map with the colored coordinates points showing sentiments for Trump and Clinton. Hover over to right side and check/uncheck different layers to see the positive and negative sentiments for these two candidates.

To see the map used in the presentation , go to the folder “MapUsedInPresentation” (inside “map) and click html file “TweetMap\_Presented.html”

1. **Twitter Sentiment Analysis plots:**

* To plot sentiment score for each candidate in a state, run write.py

(Please change the Mongo DB collection name in the file write.py)

python write.py

This will generate four files: scores.neg, scores.pos,

times.neg and times.pos

Copy the folder “alchemyapi-recipes-twitter” files to a machine, in which a R Studio/ R Shell is intaled and run R file “individual\_sentiment\_plot.R”.

R < individual\_sentiment\_plot.R --vanilla

This will generate following 3 image files for each collection:

twitter\_sentiment\_kernel.png, twitter\_sentiment\_raw.png and twitter\_sentiment\_volume.png.

* To plot Twitter sentiment for both the candidates, run R file “total\_sentiment\_plot.R”.

R < total\_sentiment\_plot.R

Make sure the file “results.csv” with each candidates sentiment score is present in the folder.