Name : Riyaz Rafi Ahmed

Email : *[riyazahm3d@gmail.com](mailto:riyazahm3d@gmail.com)*

Contact : 9629375211

**Collecting tweets from Twitter**

* We use the tweepy library to gather tweets from twitter and pymongo to load it into the mongodb instance
* The consumer\_key, consumer\_secret, access\_token and the access\_token\_secret are fetched from the twitter app created and referenced here
* Pymongo is initialized with the database name and collection name
* A class is defined to fetch the tweet and print it’s contents and count. It also inserts the fetched details of the tweet into the mongodb collection created.
* A filter is further defined to narrow down tweet results based on location ( which is set to the entire earth ) and language as English.
* The words tracked are { "a", "the", "i", "you", "u", "lol", "love" } as these are the most frequently occurring words in tweets ( based on statistics from <http://techland.time.com/2009/06/08/the-500-most-frequently-used-words-on-twitter/> )

**Extracting the text only from the collected tweets**

* We use pymongo and json libraries to work on the above mentioned function.
* We initialize the mongodb client and set the name of the database and the collection we intend to work with.
* A file is opened in write mode to insert the text collected to. ( FILE A )
* For each line present in the tweet, we iterate over each entry and filter only the text part of it and dump it into the file using json.dump ( to preserve the JSON format )
* The file is closed

**Identifying the named entities from the extracted text in the file**

* We use the ntlk { Natural Language Tool Kit } to find the named entities present in the text extracted and stored in a file in the previous code.
* A function is defined to extract only the named entities whose label reads “ NE ” and consider only those.
* One file is opened to read the text from ( FILE A ) and another file is opened to write the named entities extracted to ( FILE B )
* We read each line from file A and perform word wise tokenizing in it. Further, we speech tag the words and then chunk it to gather the named entities.
* For every word present in the extracted set of words, we print it and insert it into file B and then append a new line for convenience purposes.
* We also print a new line for the same reason stated above.

**Retrieve news based on the named entities present in file B**

* We use pymongo and newsapi libraries to connect to the mongodb database and collect news articles based on the named entities present in file B.
* The name of the database and collection is defined.
* An authorization is done for the newsapi ( key gathered from an account created )
* Two files are opened. One is file B to read the named entities from and the other is file C to append the news articles to. The former is opened in read mode and the latter is opened in write mode for rather obvious reasons.
* For every word in file B, we gather news articles of language as English, from date as the date whichever you wish to set it to { please update the date as per your requirements } and we sort the news articles based on relevancy for accurate results. Only one news article is extracted and hence page size is 1.
* The news data is dumped into file C using json.dump to preserve the JSON format
* The news article is also inserted into the database defined above.
* Please note : This method works only for the first 1000 words. For any query done above 1000 words, a premium paid plan is necessary.

**Extract news text only from the news JSON file**

* We use pymongo to connect to the mongodb database.
* A query parameter is defined to extract only the title and description of the news article and rule out all the other unnecessary fields.
* File D is opened to store only the text extracted from the news in write mode.
* A loop is setup to iterate through each news article and print the title and description. Further, we dump the description of the news article into file D using json.dump to preserve the JSON format. A new line is appended for convenience reasons.
* As the “text:” is also present in file D, we replace it with blank spaces to further improve the quality of the document and also increase convenience to access this later on.
* The file D is closed.

**Sentiment Analysis**

* Here, we use VADER sentiment to achieve sentiment analysis of the extracted contents above.
* A function is defined with two parameters ( the input file containing the sentences of which we need to find the sentiment, and another file to which we import the calculated sentiment values ) which calculates the VADER sentiment of the parameters given as input.
* The sentiment values are printed too.
* The texts extracted from the tweets is passed as input and sentiment analysis is done and stored in a separate file.
* The texts extracted from the news is passed as input to the function and sentiment analysis is done and the result is stored in another file.
* All the files used above are closed.

**Graph plotting ( 7 & 8 )**

* Matplotlib is used to plot the sentiments as a pie chart. To be specific, we use the pyplot function present in the matplotlib library.
* PLEASE NOTE : Run 7VaderGraphForText and 8VaderGraphForNews simultaneously to compare the pie chart of the tweet and the respective news article. Close each one to load the next tweet and its respective news. Keep note of which graph represents the tweet and which represents the news article. Drag as per requirements.
* The colors used are : Green for positive, Red for negative and Blue for neutral.
* For each sentiment analysis present in the file which is opened beforehand, we plot a pie chart.
* Pie chart characteristics :
  + - Starting angle is 90 degrees
    - Radius of the pie chart is 1
    - The values are rounded off to x.y format
    - Shadows are enabled
* The above pie chart is plotted for the texts in tweets and the texts in news.

Refer to ‘Databases Used’ folder for the databases used.   
The databases were exported using the mongoexport function as follows :

*Mongoexport --db dbase --collection tcollec --out TweetDbase,json*

*Mongoexport --db dbasenews --collection ncollec --out NewsDbase.json*

Python editor used is PyCharm Community edition 2017.3

Interpreter is Python 3.6

Mongodb server set up locally under hostname 127.0.0.1 and port number 27017

Mongodb version : 3.6.3 Community version   
mongodb compass used to view databases ( recommended )

Various libraries used are mentioned above. Here’s a summary :   
tweepy, matplotlib, newsapi, json, sys, os, vadersentiment, pymongo and ntlk