CS172 Final Project

By NoticeMeSen.py

For our group project, for Parts 1 and 2, we will be doing option 2. We will be working with the Twitter API and parsing Twitter JSON objects with ElasticSearch in Python. For part 3, we will doing extension 4, creating an interface and working with Twitter geolocations.

Part 1: Crawling with Twitter API (Liam Han)

Stream twitter data and collect tweets by geo-location based in California and New York. Use 'tweepy' library to access twitter API. Collected >1gb worth of data (approx. 4 days). Used temporary json file with <100 tweets to test code.

Started by loading the json file into 'geo_located_tweets' list(). Looped through each item in the collection and grabbed each 'tweet' under ['text'] key then appending to a tweets list(). The Twitter API truncates tweets that have more than 280 characters. To alleviate this, checked to see if tweet['truncated'] was True, which means the tweet value was stored under ['extended_tweet'][full_text] key.

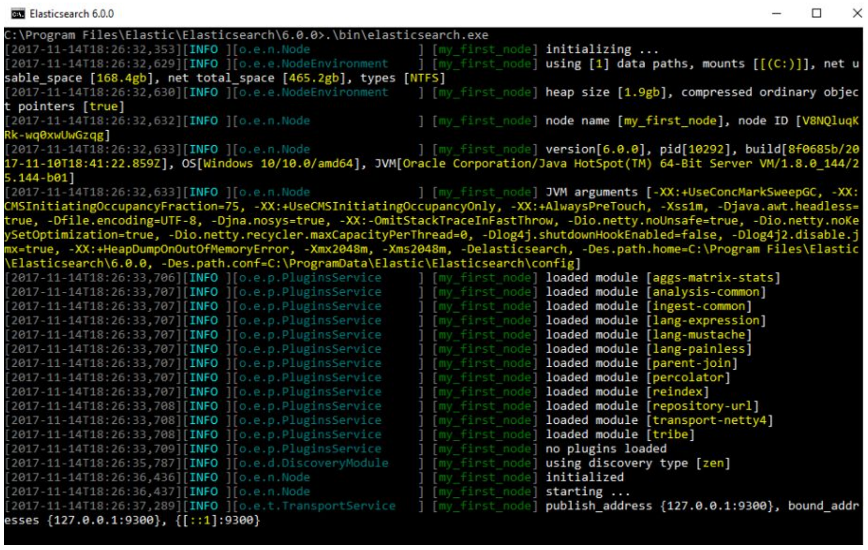
Used re.search to extract URLs inside a tweet text and input the url into BeautifulSoup to collect the webpage title of given url. Created a new ['has_url'] key for each item in the collection set to 'False' as default, set to True if the tweet has a url. If ['has_url'] key is True, create ['webpage_title'] key and set equal to title collected by BeautifulSoup. Once finished, write to new Json file.

Part 2: Index building with ElasticSearch (Antonius Panggabean)

1. Description of system:
Created function that takes all tweets in .json file from part 1 and place them iteratively (bad/slow) into the new Lucene index. Created function that takes user input of tweet search to search based on tweet text. Using the geolocation and text as a query search, use es.search() from elastic to return ranked matching tweets. Additionally created function that takes the 4 rectangular coordinates of a tweet and averages to a center point of the tweet. Elasticsearch handles the indexing similarly to a dictionary, using the different JSON fields as keys. Elasticsearch also handles the searching algorithm that ranks documents based on similarity with the queue. My retrieval function creates the query by matching the JSON tweet "{text:}" and "place{coordinates:}" and selecting tweets only through those fields.

2. Limitations of system:
At the time of demonstration to Professor Salloum, the .json tweets were iteratively placed into the elasticsearch index node one at a time. This is highly inefficient which creates a long setup time when first deploying the program. It makes it almost impossible to wait for all the tweets in the system to be processed into the index. I was not aware of the “Bulk load” function that I should have used to place all the data in the index significantly faster.

3. Instructions on how to deploy the system:
Must have ElasticSearch and Java installed on system. To utilize ElasticSearch in Program, must run elasticsearch.exe (or elasticsearch.bat) for example /.../bin/elasticsearch.exe. This creates an Elasticsearch node that utilizes Lucene and allows the functions in the program to PUT/ and GET/ data from the node.



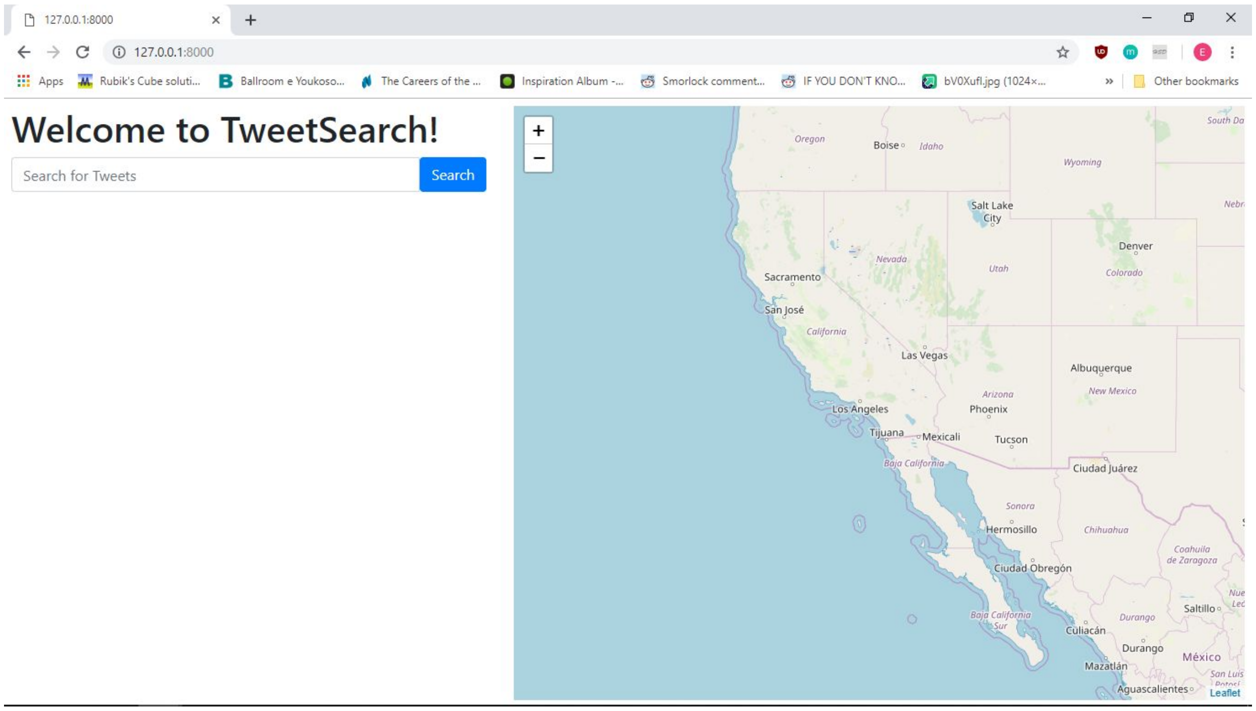
```
C:\Program Files\Elastic\Elasticsearch\6.0.0>.\bin\elasticsearch.exe
[2017-11-14T18:26:32,353][INFO ][o.e.n.Node               ] [my_first_node] initializing ...
[2017-11-14T18:26:32,629][INFO ][o.e.e.NodeEnvironment    ] [my_first_node] using [1] data paths, mounts [[(C:)]], net u
sable_space [168.4gb], net total_space [465.2gb], types [NTFS]
[2017-11-14T18:26:32,630][INFO ][o.e.e.NodeEnvironment    ] [my_first_node] heap size [1.9gb], compressed ordinary objec
t pointers [true]
[2017-11-14T18:26:32,632][INFO ][o.e.n.Node               ] [my_first_node] node name [my_first_node], node ID [V8NQluqK
Rk-wq0xwUwGzqg]
[2017-11-14T18:26:32,633][INFO ][o.e.n.Node               ] [my_first_node] version[6.0.0], pid[10292], build[8f0685b/20
17-11-10T18:41:22.859Z], OS[Windows 10/10.0/amd64], JVM[Oracle Corporation/Java HotSpot(TM) 64-Bit Server VM/1.8.0_144/2
5.144-b01]
[2017-11-14T18:26:32,633][INFO ][o.e.n.Node               ] [my_first_node] JVM arguments [-XX:+UseConcMarkSweepGC, -XX:
CMSInitiatingOccupancyFraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -XX:+AlwaysPreTouch, -Xss1m, -Djava.awt.headless=
true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -XX:-OmitStackTraceInFastThrow, -Dio.netty.noUnsafe=true, -Dio.netty.noKe
ySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.disable.j
mx=true, -XX:+HeapDumpOnOutOfMemoryError, -Xmx2048m, -Xms2048m, -Delasticsearch, -Des.path.home=C:\Program Files\Elastic
\Elasticsearch\6.0.0, -Des.path.conf=C:\ProgramData\Elastic\Elasticsearch\config]
[2017-11-14T18:26:33,706][INFO ][o.e.p.PluginsService     ] [my_first_node] loaded module [aggs-matrix-stats]
[2017-11-14T18:26:33,707][INFO ][o.e.p.PluginsService     ] [my_first_node] loaded module [analysis-common]
[2017-11-14T18:26:33,707][INFO ][o.e.p.PluginsService     ] [my_first_node] loaded module [ingest-common]
[2017-11-14T18:26:33,707][INFO ][o.e.p.PluginsService     ] [my_first_node] loaded module [lang-expression]
[2017-11-14T18:26:33,707][INFO ][o.e.p.PluginsService     ] [my_first_node] loaded module [lang-mustache]
[2017-11-14T18:26:33,707][INFO ][o.e.p.PluginsService     ] [my_first_node] loaded module [lang-painless]
[2017-11-14T18:26:33,707][INFO ][o.e.p.PluginsService     ] [my_first_node] loaded module [parent-join]
[2017-11-14T18:26:33,707][INFO ][o.e.p.PluginsService     ] [my_first_node] loaded module [percolator]
[2017-11-14T18:26:33,707][INFO ][o.e.p.PluginsService     ] [my_first_node] loaded module [reindex]
[2017-11-14T18:26:33,708][INFO ][o.e.p.PluginsService     ] [my_first_node] loaded module [repository-url]
[2017-11-14T18:26:33,708][INFO ][o.e.p.PluginsService     ] [my_first_node] loaded module [transport-netty4]
[2017-11-14T18:26:33,708][INFO ][o.e.p.PluginsService     ] [my_first_node] loaded module [tribe]
[2017-11-14T18:26:33,709][INFO ][o.e.p.PluginsService     ] [my_first_node] no plugins loaded
[2017-11-14T18:26:35,787][INFO ][o.e.d.DiscoveryModule    ] [my_first_node] using discovery type [zen]
[2017-11-14T18:26:36,436][INFO ][o.e.n.Node               ] [my_first_node] initialized
[2017-11-14T18:26:36,437][INFO ][o.e.n.Node               ] [my_first_node] starting ...
[2017-11-14T18:26:37,289][INFO ][o.e.t.TransportService   ] [my_first_node] publish_address {127.0.0.1:9300}, bound_addr
esses {127.0.0.1:9300}, {[::1]:9300}
```

Part 3: Twitter Geolocation and Interface Extension (Elijah Nicasio)

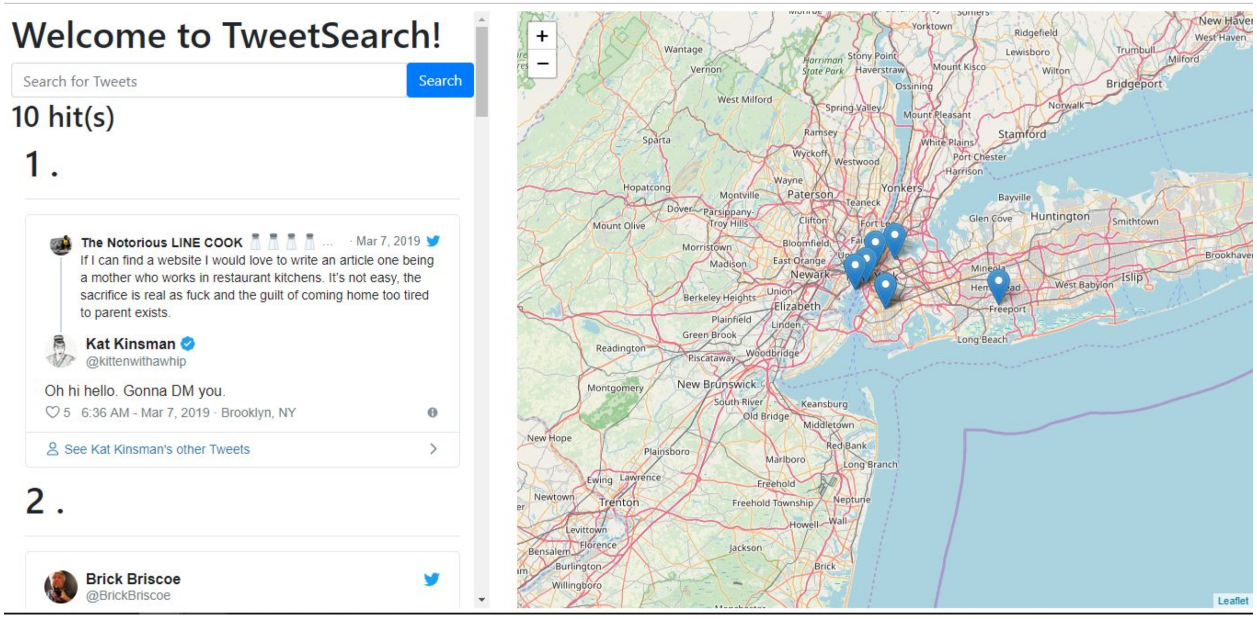
For part 3, we decided to do idea number 4, and build a UI that allows for search and display it on a map. The UI is in the form of a website. I used Django to create and run the website. In Django, I used Folium to display the map and place the tweets on the map, according to their geolocation. I used Bootstrap to design the UI. I displayed the search results as embedded tweets, unless the tweet was deleted, in which case they are shown as plain text.

The submitted program will index the data in webApp/tweetMap/tweets/t_tweets.json, as the actual file to be tested is too large to be pushed onto github.

To run everything altogether, you must have python and virtualenv. Virtualenv is not necessary, however, if you lack the libraries necessary to run our program (which you most likely will), virutalenv simply allows you to run it without installing all the required libraries. First, activate the virtual environment by running ‘source /loc/activate’. Activate is located in webApp/env/scripts. Then, go to the webapp folder and run ‘python manage.py runserver’. Connect to the website, which is locally hosted at ‘http://127.0.0.1:8000/’, and our program should be working. Enter the query you which to search for, and the program will return a list of tweets from California and New York in a ranked list. Keep in mind that this all assumes that indexing has already been done as per the part 2, in elasticsearch.


127.0.0.1:8000

Welcome to TweetSearch!


Welcome to TweetSearch!

10 hit(s)

1 .

The Notorious LINE COOK · Mar 7, 2019
If I can find a website I would love to write an article one being a mother who works in restaurant kitchens. It's not easy, the sacrifice is real as fuck and the guilt of coming home too tired to parent exists.

Kat Kinsman
@kittenwithawhip

Oh hi hello. Gonna DM you.
♡5 6:36 AM - Mar 7, 2019 · Brooklyn, NY

See Kat Kinsman's other Tweets

2 .

Brick Briscoe
@BrickBriscoe