Final Project: Twitter Geolocations

Team Members: Edward Zabrensky, Siddharth Menon, Joshua Beto

Description

Our final project uses the Twitter Stream API to collect Geolocated Tweets. We create an index on these Tweets using Lucene to generate results for search queries. We plot these results on a map using the Google Maps API.

Collaboration Details

We decided to split up the 3 parts per person. Our collaboration details are as follows:

Part 1: Twitter Stream - Joshua Beto
Part 2: Lucene Index / JSON Parsing - Siddharth Menon
Part 3: Geolocation Extension - Edward Zabrensky

Part 1: Twitter Stream

Assigned: Joshua Beto

Architecture:

The architecture is split into a few basic modules. Their functionalities are listed below:

streams module: Handles any logic that deals with the Twitter Stream API (Connection, Reconnecting, Tweets, etc.). This module also adds the 'url_title' field to the Tweet JSON through the map_urls function

listener module: Callback that the streams module calls on when the data is collected. This module allows us to record the number of Tweets streamed and writes the data to its corresponding file.

config module: Handles any configuration data including the Twitter API keys, endpoints, and output file names.

oauth module: Handles any oauth authentication. This module is used by the streams module to connect to the Twitter endpoints.

app module: Handles our main logic and calls our streams module. This module uses user input to determine when to stop streaming.

Data Collection:

Threads / Processes:

The call to the Twitter Streaming API uses a single thread. Collecting roughly 1 GB of data takes ~3 hours on an AMD Ryzen 5 2600 CPU when using a single thread. Adding the url_title field to that JSON data takes another ~3 hours, and uses up to 16 processes in a producer-consumer concurrency model.

The producer is the intermediate data file called tweets.json while the consumer processes the urls found in each JSON of the tweets.json file, visits that page, and parses the 'title' tag of the html received from the get request. It then appends the 'url_title' field with that parsed title to the json data, which it then writes to the file 'tweets_url.data'

Duplicate Tweets:

To account for duplicates, we have a check.py script which adds each Tweet from the 'tweets_url.data' file into a set to filter out any duplicates. The script then writes every element of the set back into the file 'tweets_url_no_dup.data'

Data Structures:

The streaming process is as follows:

- Stream Tweet JSON Data -> Write JSON data to tweets.data
- For each JSON in tweets.data, construct a new JSON object with the 'url_title' field
- Write that new JSON object to tweets_url.data

Our project uses files to store each intermediate step. This is so no data needs to be re-streamed in case of an error when the 'url_title' field is added. For simplicity, we convert back and forth between JSON strings and Python dictionaries to make the implementation easier when adding the 'url_title' field.

One issue we came across is a *MemoryError* when distributing the jobs to the 16 processes to add the 'url_title' field. To counteract this, we split the total JSON into chunks, and passed in a chunk of jobs at a time instead of all at once.

Our producer-consumer concurrency model implicitly uses a queue to administer jobs. To check for duplicate Tweets in our check.py script, we use a set that stores each of the JSON data.

Limitations:

Not Automated: The main drawback to this project is the lack of automation. The application is contingent on the user entering input to determine when Tweet Streaming should end. Another drawback is the lack of automated duplication checking since duplication checking was more of an afterthought than something planned during architecture.

Lack of Testing: While the application works in the basic scenario, events that are not accounted for include missing out on stream data during disconnects and more robust disconnection handling.

Deployment:

Requirements: Python 3.6, Pip installed on Python 3.6

To deploy, run the following steps:

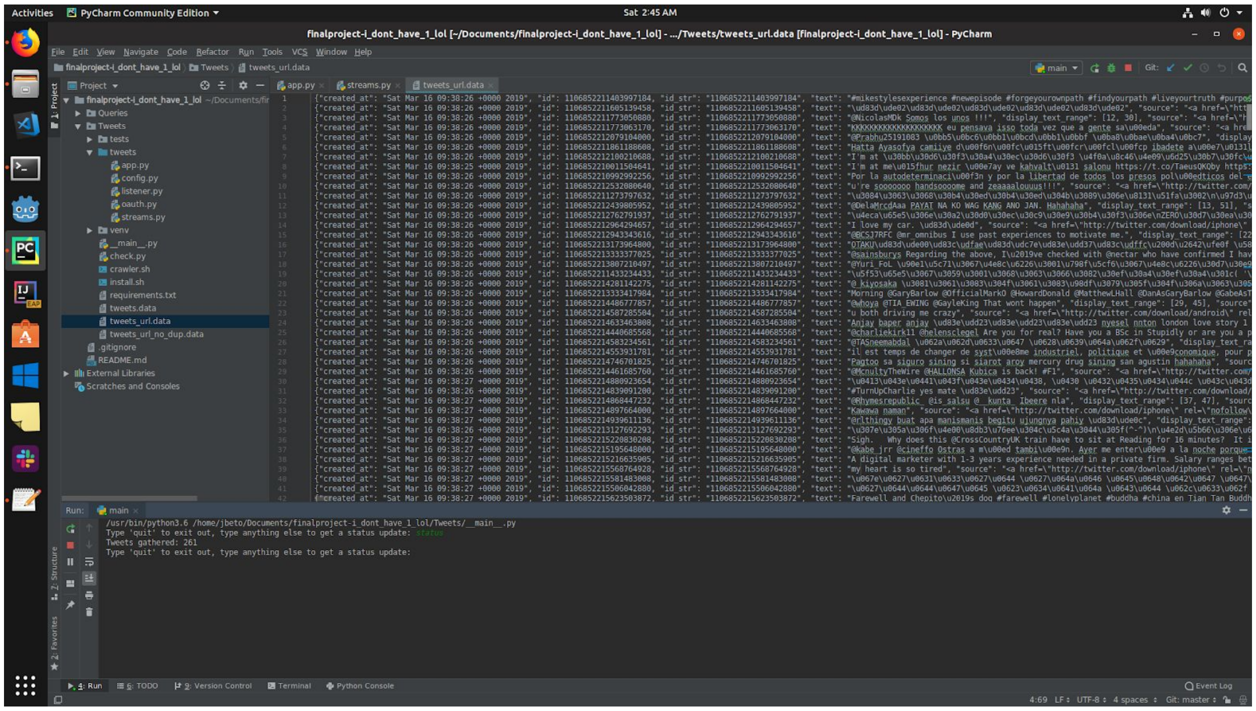
- Make sure you are on the Tweets directory
- Run ./install.sh - This will install the Python virtual environment with all the required packages
- Alternatively, you can manually create the virtual environment through the virtualenv package and use the command: pip install -r requirements.txt
- Run ./crawler.sh <CONSUMER_KEY> <CONSUMER_SECRET> <ACCESS_TOKEN> <ACCESS_SECRET>

CONSUMER_KEY = Twitter API Consumer Key
CONSUMER_SECRET = Twitter API Consumer Secret
ACCESS_TOKEN = Twitter API Access Token
ACCESS_SECRET = Twitter API Access Token Secret

You can retrieve these keys by first making a Twitter development account and creating an app.

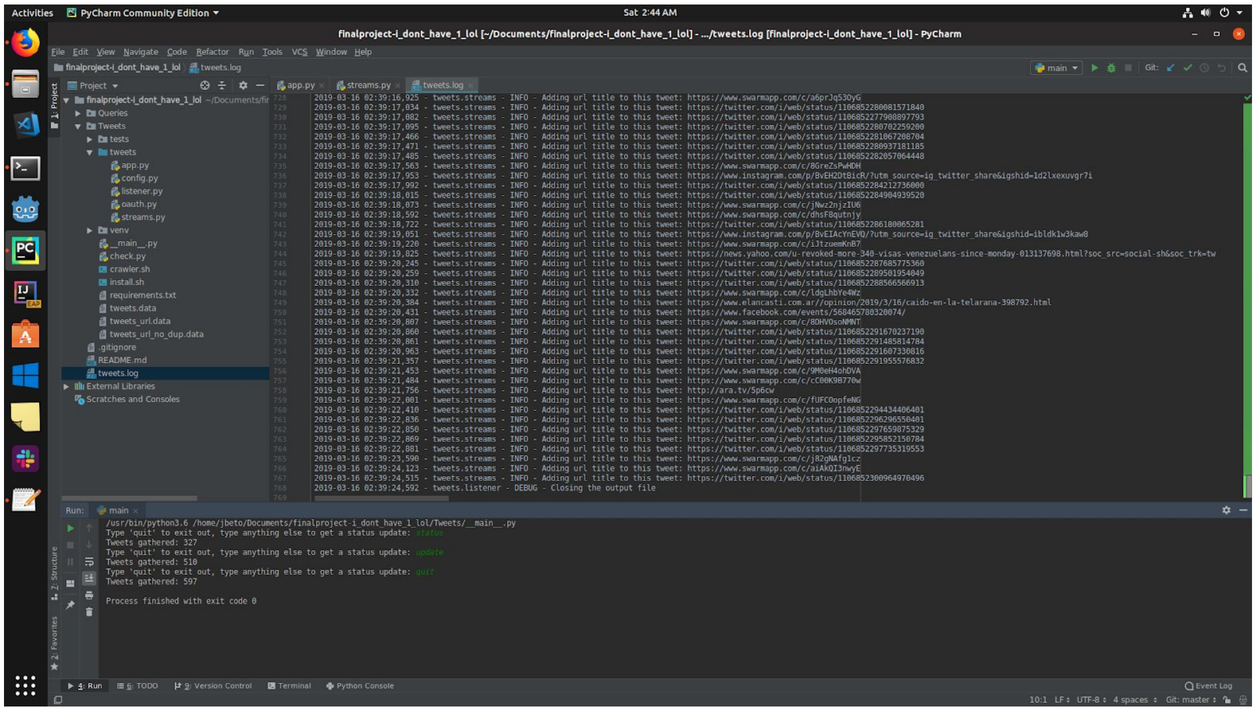
The process is documented here:
https://developer.twitter.com/en/docs/basics/developer-portal/overview

Result Screenshots:



```
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852211403997184, "id_str": "1106852211403997184", "text": "#mikestylesexperience #newepisode #forgeyourownpath #findyourpath #liveyourtruth #purpo
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852211605139458, "id_str": "1106852211605139458", "text": "\ud83d\ude02\ud83d\ude02\ud83d\ude02\ud83d\ude02\ud83d\ude02", "source": "<a href=\"htt
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852211773050880, "id_str": "1106852211773050880", "text": "@NicolasMDk Somos los unos !!!", "display_text_range": [12, 30], "source": "<a href=\"h
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852211773063170, "id_str": "1106852211773063170", "text": "KKKKKKKKKKKKKKKKKKK eu pensava isso toda vez que a gente sa\u00eda", "source": "<a hre
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852212079104000, "id_str": "1106852212079104000", "text": "@Prabhu25191083 \u0bb5\u0bc6\u0bb1\u0bcd\u0bb1\u0bbf \u0ba8\u0bae\u0ba4\u0bc7", "display
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852211861188608, "id_str": "1106852211861188608", "text": "Hatta Ayasofya camiiye d\u00f6n\u00fc\u015ft\u00fcr\u00fcl\u00fcp ibadete a\u00e7\u0131l
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852212100210688, "id_str": "1106852212100210688", "text": "I'm at \u30bb\u30d6\u30f3\u30a4\u30ec\u30d6\u30f3 \u4f0a\u8c46\u4e09\u6d25\u30b7\u30fc
{"created_at": "Sat Mar 16 09:38:25 +0000 2019", "id": 1106852210011504641, "id_str": "1106852210011504641", "text": "I'm at me\u015fhur nezir \u00e7ay ve kahvalt\u0131 salonu https://t.co/TaeusQKQby https
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852210992992256, "id_str": "1106852210992992256", "text": "Por la autodeterminaci\u00f3n y por la libertad de todos los presos pol\u00edticos del
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852212532080640, "id_str": "1106852212532080640", "text": "u're sooooooo handsoooome and zeaaaalouuus!!!", "source": "<a href=\"http://twitter.com/
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852211273797632, "id_str": "1106852211273797632", "text": "\u3084\u3063\u3068\u30b4\u30ed\u30b4\u30ed\u304b\u3089\u306e\u8131\u51fa\u3002\n\u97d3\u
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852212439805952, "id_str": "1106852212439805952", "text": "@DelaMrcdAaa PAYAT NA KO WAG KANG ANO JAN. Hahahaha", "display_text_range": [13, 51], "s
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852212762791937, "id_str": "1106852212762791937", "text": "\u4eca\u65e5\u306e\u30a2\u30d0\u30ec\u30c9\u30e9\u30b4\u30f3\u306e\nZERO\u30d7\u30ea\u30
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852212964294657, "id_str": "1106852212964294657", "text": "I love my car. \ud83d\ude0d", "source": "<a href=\"http://twitter.com/download/iphone\"
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852212943343616, "id_str": "1106852212943343616", "text": "@BCSJ7RFC @mr_omnibus I use past experiences to motivate me.", "display_text_range": [22
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852213173964800, "id_str": "1106852213173964800", "text": "OTAKU\ud83d\ude00\ud83c\udfae\ud83d\udc7e\ud83e\udd37\ud83c\udffc\u200d\u2642\ufe0f \u58
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852213333377025, "id_str": "1106852213333377025", "text": "@sainsburys Regarding the above, I\u2019ve checked with @nectar who have confirmed I hav
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852213807210497, "id_str": "1106852213807210497", "text": "@Yuri_FoL \u90e1\u5c71\u3067\u4e8c\u6226\u3001\u798f\u5cf6\u3067\u4e8c\u6226\u30d7\u30e9
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852211433234433, "id_str": "1106852211433234433", "text": "\u5f53\u65e5\u3067\u3059\u3001\u3068\u3063\u3066\u3082\u30ef\u30a4\u30ef\u30a4\u301c( '\
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852214281142275, "id_str": "1106852214281142275", "text": "@_kiyosaka \u3081\u3061\u3083\u304f\u3061\u3083\u98df\u3079\u305f\u304f\u306a\u3063\u305
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852213333417984, "id_str": "1106852213333417984", "text": "Morning @GaryBarlow @OfficialMarkO @HowardDonald @MatthewLHall @DanAsGaryBarlow @GabeAsT
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852214486777857, "id_str": "1106852214486777857", "text": "@whoya @TIA_EWING @GayleKing That wont happen", "display_text_range": [29, 45], "source"
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852214587285504, "id_str": "1106852214587285504", "text": "u both driving me crazy", "source": "<a href=\"http://twitter.com/download/android\" rel
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852214633463808, "id_str": "1106852214633463808", "text": "Anjay baper anjay \ud83e\udd23\ud83e\udd23\ud83e\udd23 nyesel nnton london love story 1
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852214440685568, "id_str": "1106852214440685568", "text": "@charliekirk11 @helensclegel Are you for real? Have you a BSc in Stupidly or are you a p
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852214583234561, "id_str": "1106852214583234561", "text": "@TASneemabdal \u062a\u062d\u0633\u0647 \u0628\u0639\u064a\u062f\u0629", "display_text_ra
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852214553931781, "id_str": "1106852214553931781", "text": "il est temps de changer de syst\u00e8me industriel, politique et \u00e9conomique, pour p
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852214746701825, "id_str": "1106852214746701825", "text": "Pagtoo sa siguro sining si siarot aroy mercury drug sining san agustin hahahaha", "sourc
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852214461685760, "id_str": "1106852214461685760", "text": "@McnultyTheWire @HALLONSA Kubica is back! #F1", "source": "<a href=\"http://twitter.com/
{"created_at": "Sat Mar 16 09:38:27 +0000 2019", "id": 1106852214880923654, "id_str": "1106852214880923654", "text": "\u0413\u043e\u0441\u043f\u043e\u0434\u0438, \u0430 \u0432\u0435\u0434\u044c \u043c\u043d
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852214839091200, "id_str": "1106852214839091200", "text": "#TurnUpCharlie yes mate \ud83e\udd23", "source": "<a href=\"http://twitter.com/download/
{"created_at": "Sat Mar 16 09:38:27 +0000 2019", "id": 1106852214868447232, "id_str": "1106852214868447232", "text": "@Rhymesrepublic_ @is_salsu @__kunta_ Ibeere nla", "display_text_range": [37, 47], "sourc
{"created_at": "Sat Mar 16 09:38:27 +0000 2019", "id": 1106852214897664000, "id_str": "1106852214897664000", "text": "Kawawa naman", "source": "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollow\
{"created_at": "Sat Mar 16 09:38:27 +0000 2019", "id": 1106852214939611136, "id_str": "1106852214939611136", "text": "@rlthingy buat apa manismanis begitu ujungnya pahiy \ud83d\ude0c", "display_text_range":
{"created_at": "Sat Mar 16 09:38:26 +0000 2019", "id": 1106852213127692293, "id_str": "1106852213127692293", "text": "\u307e\u305a\u306f\u4e00\u8db3\u76ee\u304c\u5c4a\u3044\u305f(^-^)\n\u4e2d\u5b66\u306e\u6
{"created_at": "Sat Mar 16 09:38:27 +0000 2019", "id": 1106852215220830208, "id_str": "1106852215220830208", "text": "Sigh.   Why does this @CrossCountryUK train have to sit at Reading for 16 minutes?  It i
{"created_at": "Sat Mar 16 09:38:27 +0000 2019", "id": 1106852215195648000, "id_str": "1106852215195648000", "text": "@kabe_jrr @cineffo Ostras a m\u00ed tambi\u00e9n. Ayer me enter\u00e9 a la noche porque
{"created_at": "Sat Mar 16 09:38:27 +0000 2019", "id": 1106852215216635905, "id_str": "1106852215216635905", "text": "A digital marketer with 1-3 years experience needed in a private firm. Salary ranges bet
{"created_at": "Sat Mar 16 09:38:27 +0000 2019", "id": 1106852215568764928, "id_str": "1106852215568764928", "text": "my heart is so tired", "source": "<a href=\"http://twitter.com/download/iphone\" rel=\"n
{"created_at": "Sat Mar 16 09:38:27 +0000 2019", "id": 1106852215581483008, "id_str": "1106852215581483008", "text": "\u067e\u0627\u0631\u0633\u0627\u0644 \u0627\u064a\u0646 \u0645\u0648\u0642\u0647 \u0647\
{"created_at": "Sat Mar 16 09:38:27 +0000 2019", "id": 1106852215506042880, "id_str": "1106852215506042880", "text": "\u0627\u0644\u0644\u0647\u0645 \u0623\u0634\u0641\u064a \u0643\u0644 \u062c\u0633\u062f
{"created_at": "Sat Mar 16 09:38:27 +0000 2019", "id": 1106852215623503872, "id_str": "1106852215623503872", "text": "Farewell and Chepito\u2019s dog #farewell #lonelyplanet #buddha #china en Tian Tan Buddh
```

```
/usr/bin/python3.6 /home/jbeto/Documents/finalproject-i_dont_have_1_lol/Tweets/__main__.py
Type 'quit' to exit out, type anything else to get a status update: status
Tweets gathered: 261
Type 'quit' to exit out, type anything else to get a status update:
```



```
2019-03-16 02:39:16,925 - tweets.streams - INFO - Adding url title to this tweet: https://www.swarmapp.com/c/a6prJq53OyG
2019-03-16 02:39:17,034 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852280081571840
2019-03-16 02:39:17,082 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852277908897793
2019-03-16 02:39:17,095 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852280702259200
2019-03-16 02:39:17,466 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852281067208704
2019-03-16 02:39:17,471 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852280937181185
2019-03-16 02:39:17,485 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852282057064448
2019-03-16 02:39:17,563 - tweets.streams - INFO - Adding url title to this tweet: https://www.swarmapp.com/c/8GreZsPwHDH
2019-03-16 02:39:17,953 - tweets.streams - INFO - Adding url title to this tweet: https://www.instagram.com/p/BvEH2DtBicR/?utm_source=ig_twitter_share&igshid=1d2lxexuvgr7i
2019-03-16 02:39:17,992 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852284212736000
2019-03-16 02:39:18,015 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852284904939520
2019-03-16 02:39:18,073 - tweets.streams - INFO - Adding url title to this tweet: https://www.swarmapp.com/c/jNwz2njzIU6
2019-03-16 02:39:18,592 - tweets.streams - INFO - Adding url title to this tweet: https://www.swarmapp.com/c/dhsF8qutnjy
2019-03-16 02:39:18,722 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852286180065281
2019-03-16 02:39:19,051 - tweets.streams - INFO - Adding url title to this tweet: https://www.instagram.com/p/BvEIAcYnEVQ/?utm_source=ig_twitter_share&igshid=ibldk1w3kaw8
2019-03-16 02:39:19,220 - tweets.streams - INFO - Adding url title to this tweet: https://www.swarmapp.com/c/iJtzuemKnB7
2019-03-16 02:39:19,825 - tweets.streams - INFO - Adding url title to this tweet: https://news.yahoo.com/u-revoked-more-340-visas-venezuelans-since-monday-013137698.html?soc_src=social-sh&soc_trk=tw
2019-03-16 02:39:20,245 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852287685775360
2019-03-16 02:39:20,259 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852289501954049
2019-03-16 02:39:20,310 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852288566566913
2019-03-16 02:39:20,332 - tweets.streams - INFO - Adding url title to this tweet: https://www.swarmapp.com/c/ldgLhbYe4Wz
2019-03-16 02:39:20,384 - tweets.streams - INFO - Adding url title to this tweet: https://www.elancasti.com.ar//opinion/2019/3/16/caido-en-la-telarana-398792.html
2019-03-16 02:39:20,431 - tweets.streams - INFO - Adding url title to this tweet: https://www.facebook.com/events/568465780320074/
2019-03-16 02:39:20,807 - tweets.streams - INFO - Adding url title to this tweet: https://www.swarmapp.com/c/8DHVOsoNMNT
2019-03-16 02:39:20,860 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852291670237190
2019-03-16 02:39:20,861 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852291485814784
2019-03-16 02:39:20,963 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852291607330816
2019-03-16 02:39:21,357 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852291955576832
2019-03-16 02:39:21,453 - tweets.streams - INFO - Adding url title to this tweet: https://www.swarmapp.com/c/9M0eH4ohDVA
2019-03-16 02:39:21,484 - tweets.streams - INFO - Adding url title to this tweet: https://www.swarmapp.com/c/cC00K9B770w
2019-03-16 02:39:21,756 - tweets.streams - INFO - Adding url title to this tweet: http://ara.tv/5p6cw
2019-03-16 02:39:22,001 - tweets.streams - INFO - Adding url title to this tweet: https://www.swarmapp.com/c/fUFCOopfeNG
2019-03-16 02:39:22,410 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852294434406401
2019-03-16 02:39:22,836 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852296296550401
2019-03-16 02:39:22,850 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852297659875329
2019-03-16 02:39:22,869 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852295852150784
2019-03-16 02:39:22,881 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852297735319553
2019-03-16 02:39:23,590 - tweets.streams - INFO - Adding url title to this tweet: https://www.swarmapp.com/c/j82gNAfg1cz
2019-03-16 02:39:24,123 - tweets.streams - INFO - Adding url title to this tweet: https://www.swarmapp.com/c/aiAkQI3nwyE
2019-03-16 02:39:24,515 - tweets.streams - INFO - Adding url title to this tweet: https://twitter.com/i/web/status/1106852300964970496
2019-03-16 02:39:24,592 - tweets.listener - DEBUG - Closing the output file
```

```
/usr/bin/python3.6 /home/jbeto/Documents/finalproject-i_dont_have_1_lol/Tweets/__main__.py
Type 'quit' to exit out, type anything else to get a status update: status
Tweets gathered: 327
Type 'quit' to exit out, type anything else to get a status update: update
Tweets gathered: 510
Type 'quit' to exit out, type anything else to get a status update: quit
Tweets gathered: 597

Process finished with exit code 0
```

Part 2: Lucene Index

Assigned: Siddharth Menon

Architecture:

We implemented the indexing of the input twitter data using Java Lucene Core. This section of the project consists of:

The Indexer: this module takes care of taking in the input and building an inverted index, which will make searching based on aggregated points/ranking easier.

The Searcher: this module uses the index built by the indexer to search based on a query and return the ranked results ready to be viewed by the user.

Index Structures:

The input twitter data is of the json format. The indexer first parses this json data using the Java's org.json library. The parsed data gives us access to different fields in the input data, such as "Time created", "Username", "Tweet text", "Location Coordinates", and "URL".

Our Indexer takes each of these fields and builds a unique Lucene document object. This document object is then added into the associated lucene index.

Once all the data is parsed and added into the index, it is ready to be used by the Searcher to return results.

Search Algorithm:

The search algorithm uses the previously created inverted lucene index. Using the parsed twitter text data as a way to traverse the index. It is able to take in a user's input and then search the index for relevant data.

The relevant data is returned in terms of the lucene documents (as stored in the indexer structure). The relevant information can be taken out of the documents to display to the user.

Limitations:

Currently the searcher is only able to return relevant results based on the inherent Lucene ranking. However, it should be possible to gather more information during the indexing process to do some more custom ranking and give more sophisticated relevancy information.

Deployment:

To deploy this section of the project, it is required to install Java Lucene Core and download the org.json library as well. To test it in your own environment, the paths to essential Lucene .jar files and the org.json .jar file needs to be included in the environment variables. This also requires access to a data file that holds the input data.

Part 3: Geolocations

Assigned: Edward Zabrensky

Description:

For the front end portion of this project we used Flask to run our web server. The web server will wait for a user to put in a query. Once a query is requested our Flask web server will call our Lucene Indexer search function that was made in java. This call will search the index that was created and return the top 20 results. These results are then put into a list and sent back to the web server. Once the web server receives the results it will center the map around the user's location based on the browser. Then it will create google maps markers from the locations received from the indexer. The user can click on these markers and it will show the username, tweet, and the title of the link if there is a link.

We also included functions to calculate the tweet center and to calculate the distance of a tweet from the user's location. The calculated the tweet center was created because tweets are given in a bounding box, so to estimate the center of the tweet we created this function. The calculation of the distance from a tweet to the user's location was created so that we can limit the amount of tweets shown. However, since the number of tweets is only 1GB we decided to show all the tweets anywhere in the world.

Limitations:

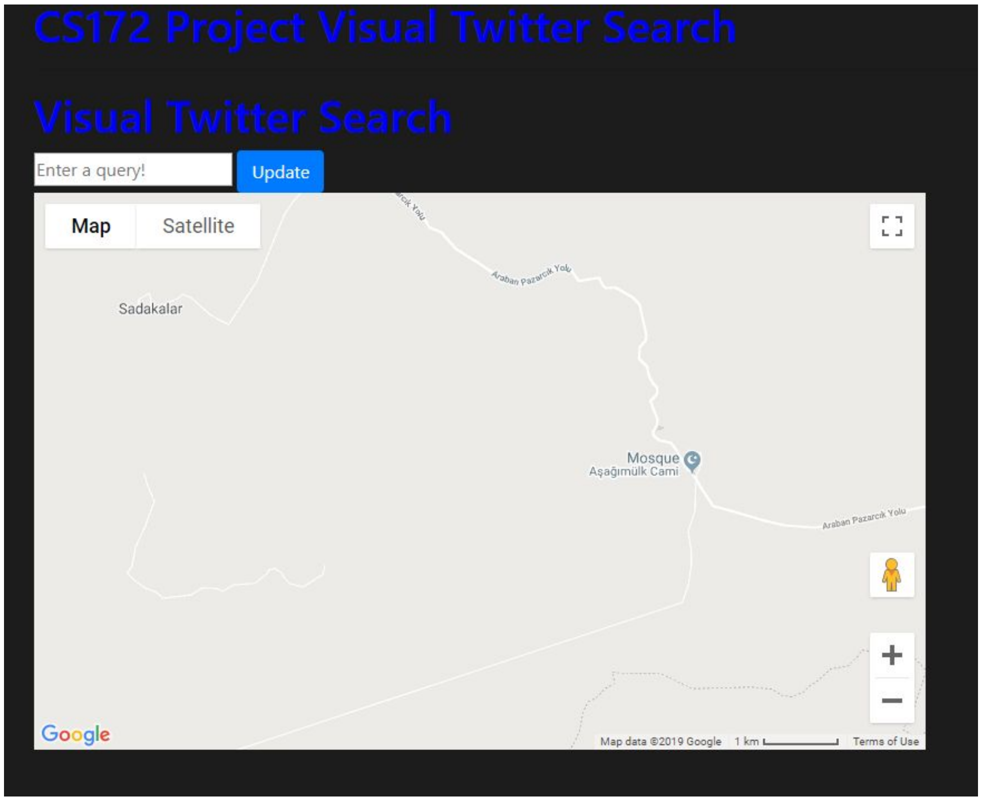
One limitation was that our indexer was created in java while our web server was created in python. To resolve this we used jython which allowed our python web server to call methods in our Java. Also to make this work we needed to create methods in the Lucene Indexer file that returned python objects instead of java ones so that we could call these methods from our python script. We also had to update the jython file itself to include the classpaths of our java dependencies because when our web server would call our Lucene Indexer more than once it would only include the classpaths for the first call to Lucene Indexer.

Another limitation was the emojis. When we tried sending the text of the tweet as a python string to the server it would always crash. We figured out that we needed to encode the text and url title of tweets as python unicode so that it would know how to encode these pieces of data.

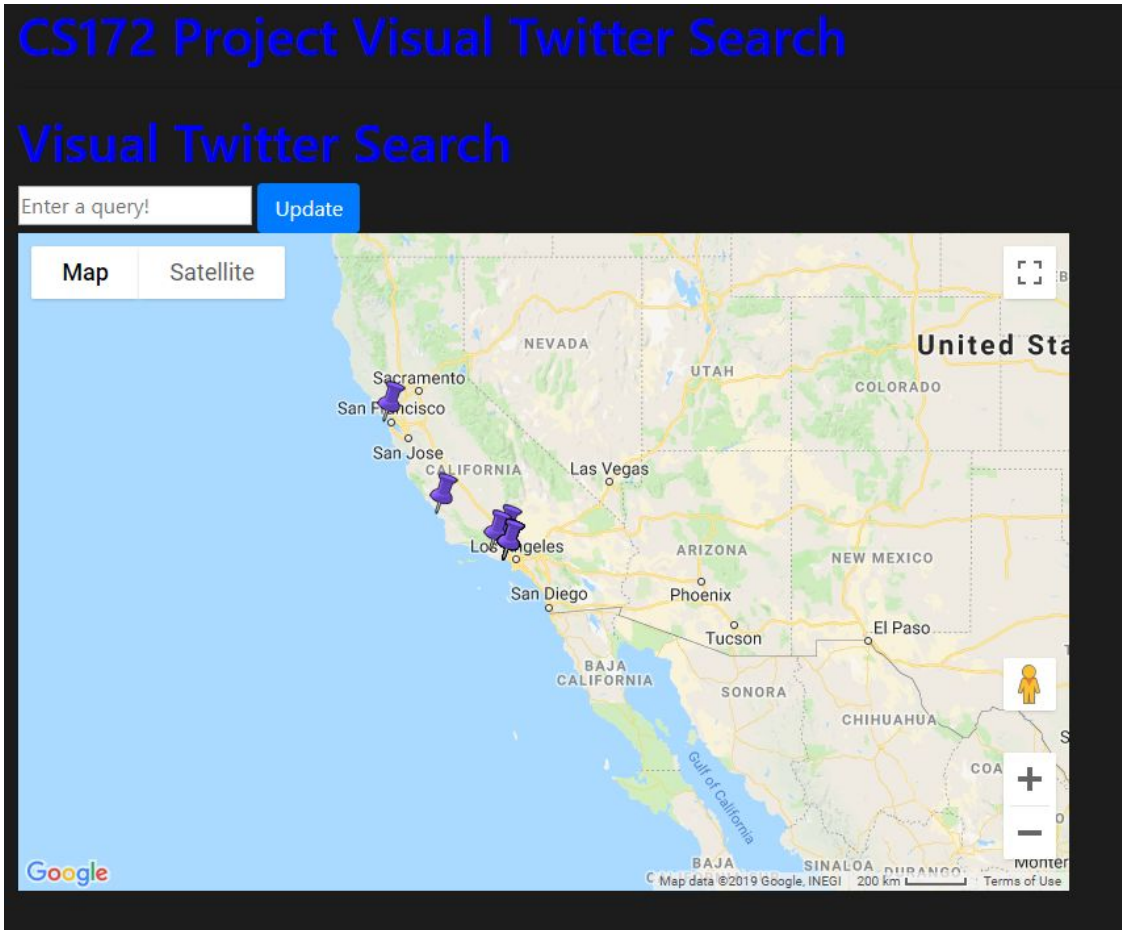
Deployment:

To deploy this front end we need google maps api key, flask, jython 2.7.1 and java's org.python.core dependency. More about deployment is on the frontend section of the readme.

Result Screenshots:

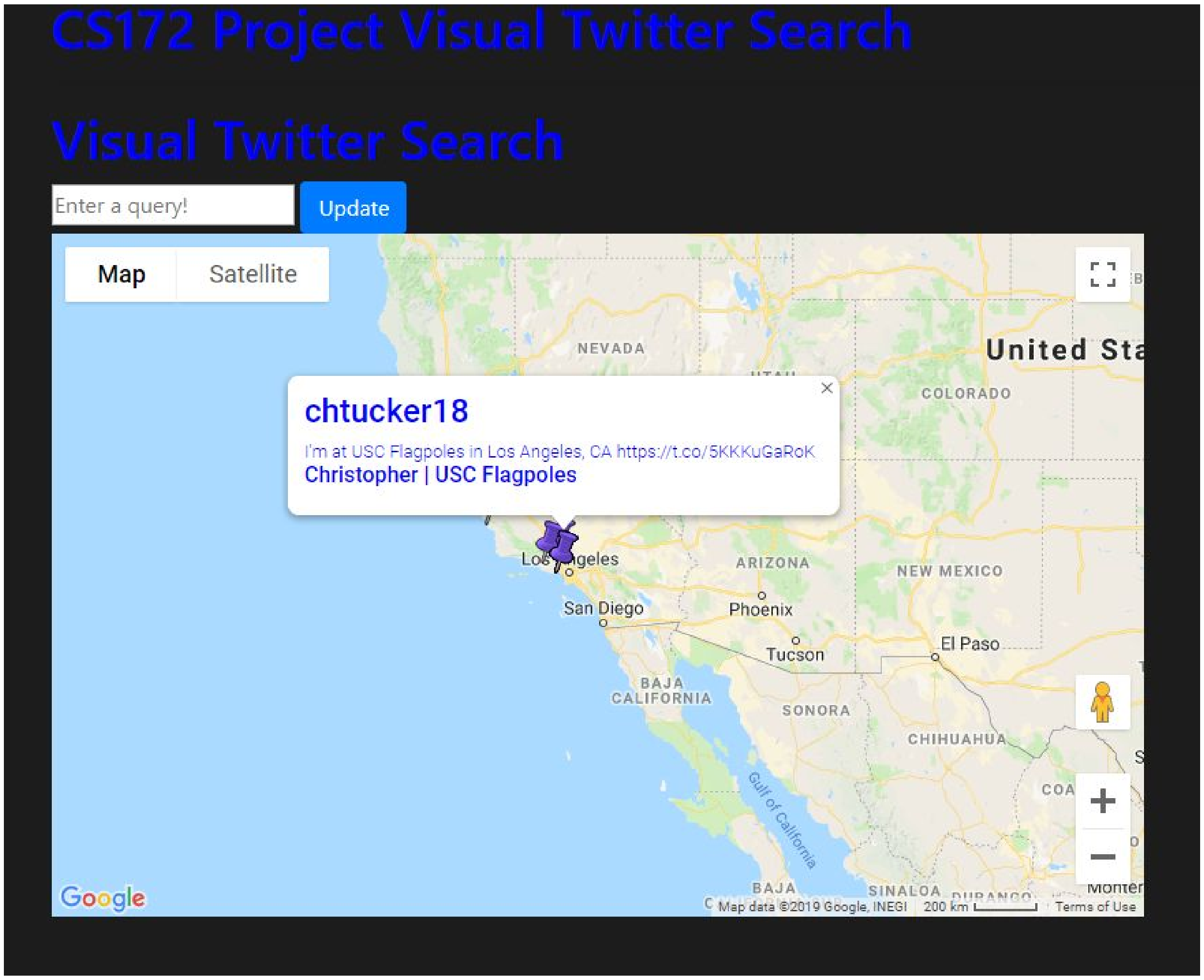


Search for ‘Angeles’


CS172 Project Visual Twitter Search

Visual Twitter Search

After clicking on a pin



CS172 Project Visual Twitter Search

Visual Twitter Search

chtucker18

I'm at USC Flagpoles in Los Angeles, CA https://t.co/5KKKuGaRoK

Christopher | USC Flagpoles

Shows the username, the tweet, and the title of the link if a link is available.