Gustavo Blanco

Michael De La Torre

CS 172: Intro to Information Retrieval

Prof: Dr. Vagelis Hristidis

TA: Shiwen Cheng

Spring 2014

**Team**

Gustavo Blanco ([gblan002@ucr.edu](mailto:gblan002@ucr.edu))

I worked on getting access to the Twitter Public Stream and parsing the JSON structures that Twitter provided in order to only save the required fields (text, timestamp, geoJSON, tweet user, link, link title). I also worked on managing all of the data and storing the required 5GB of tweets into 10MB files.

Michael De La Torre ([mdela011@ucr.edu](mailto:mdela011@ucr.edu))

I worked on parsing the stored data for possible links. If the tweet contained a link, then we evaluated the URL and retrieved the data to record the title. Afterwards, the JSON object was updated if a title was found in our files, else it was left alone.

[http://www.cs.ucr.edu/~vagelis/classes/CS172/Course%20Project.htm](http://www.cs.ucr.edu/~vagelis/classes/CS172/Course%20Project.htm" \t "_blank)

Tweet Structure: [https://dev.twitter.com/docs/platform-objects/tweets](https://dev.twitter.com/docs/platform-objects/tweets" \t "_blank)

Twitter API: [https://github.com/sixohsix/twitter](https://github.com/sixohsix/twitter" \t "_blank)

Python JSON Spec: [https://docs.python.org/2/library/json.html#module-json](https://docs.python.org/2/library/json.html" \l "module-json" \t "_blank)

SETUP (from terminal) using -Python 2.7.5

\*\* make sure 'easy\_install' is available in computer

**SYSTEM**

Development Environment:

Python 2.7.5

OS X 10.9.2 & Windows 7

Terminal

Sublime Text Editor 2

Our program simple opens up the Twitter Public Stream and uses an iterator to go through all of the tweets.

Once we’ve received a tweet we check to see if we have reached the number of desired files, in which we are done, or if the file we’re writing too has reached 10MB, in which we create a new file.

Once we know we are ready to write to a file, we parse the JSON object for the required 5 fields (text, timestamp, geoJSON, tweet user, link) in order to create a new JSON object that we will write to our file.

Occasionally our Stream times out because we reach Twitter Streaming Limit, to remedy this, we simply re-initialize the stream if we have yet to reach our 5GB data requirement.

At this point we are done collection all of our 5GB of tweets. We then parse through all of our stored tweets to see if they have a URL in their link field, if they do we evaluate the link to retrieve its data and then save the title in the linktitle field.

**LIMITATIONS**

One of our programs limitations is that it needs to finish collecting all 5GB of Tweets before it can even being to evaluate possible link’s titles. This can be remedied at a later date by using multiple threads; one to collect the tweets and another to get the link title is the tweets has a URL.

**INSTRUCTIONS**

In order to run our program, simple give permission to our crawler.sh file and then run it:

chmod +x crawler.sh & crawler.sh

Our crawler.sh will simply install three necessary dependencies:

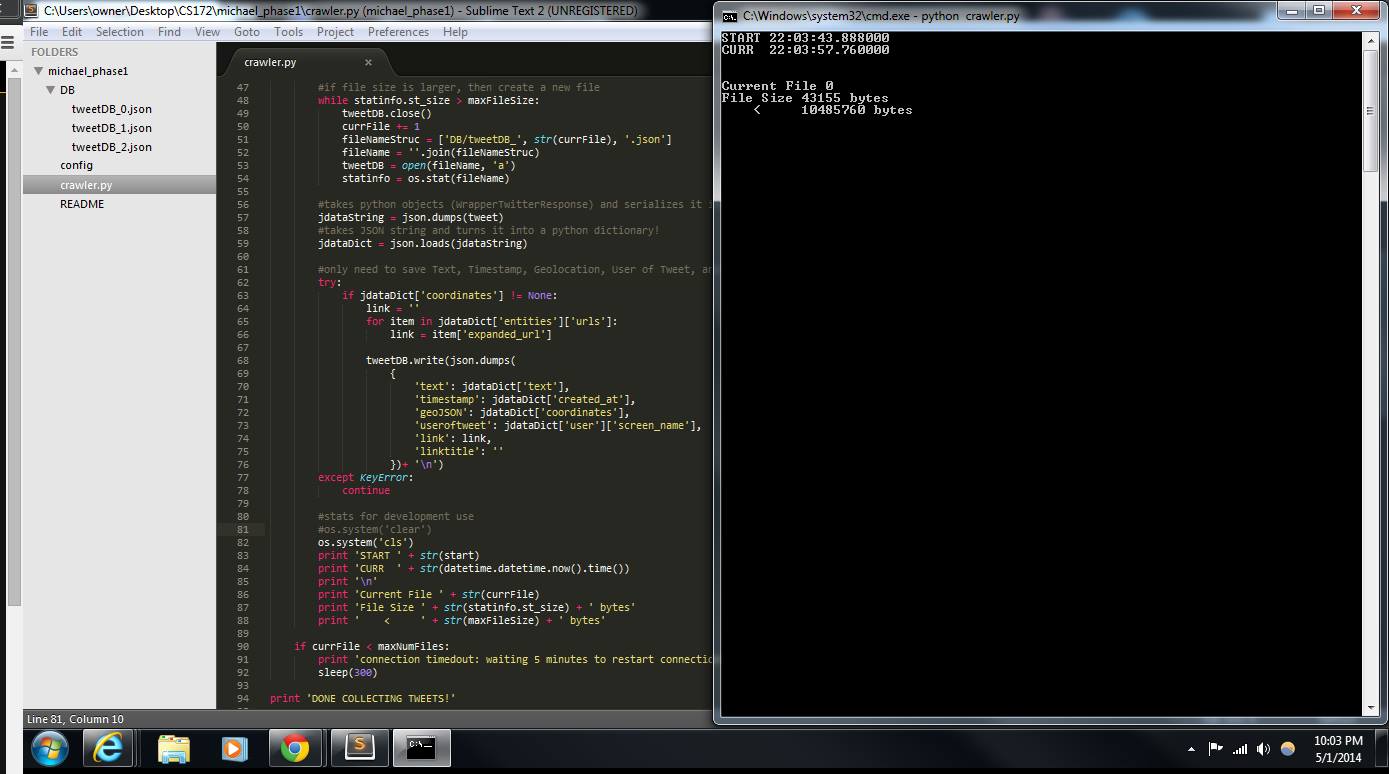
Twitter API (http://mike.verdone.ca/twitter/)

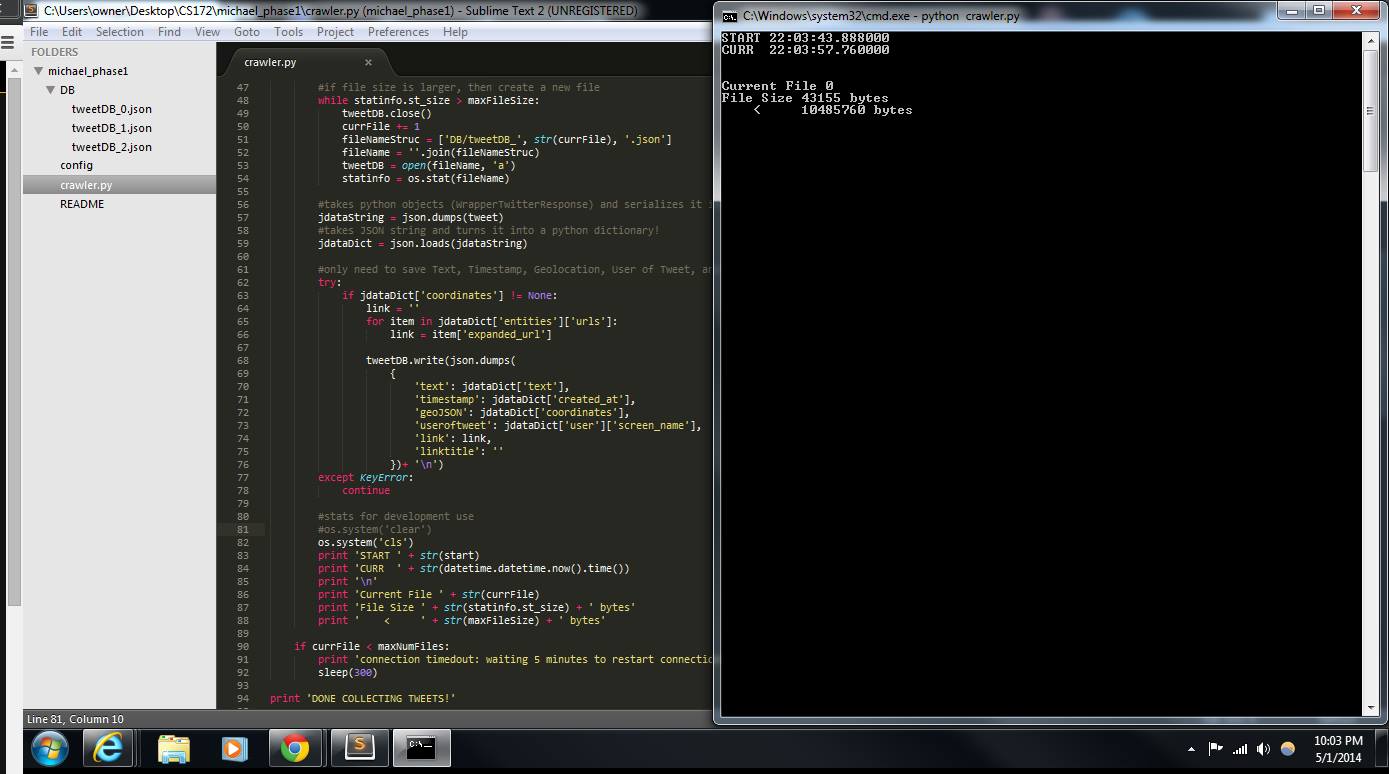
lxml (http://lxml.de/)

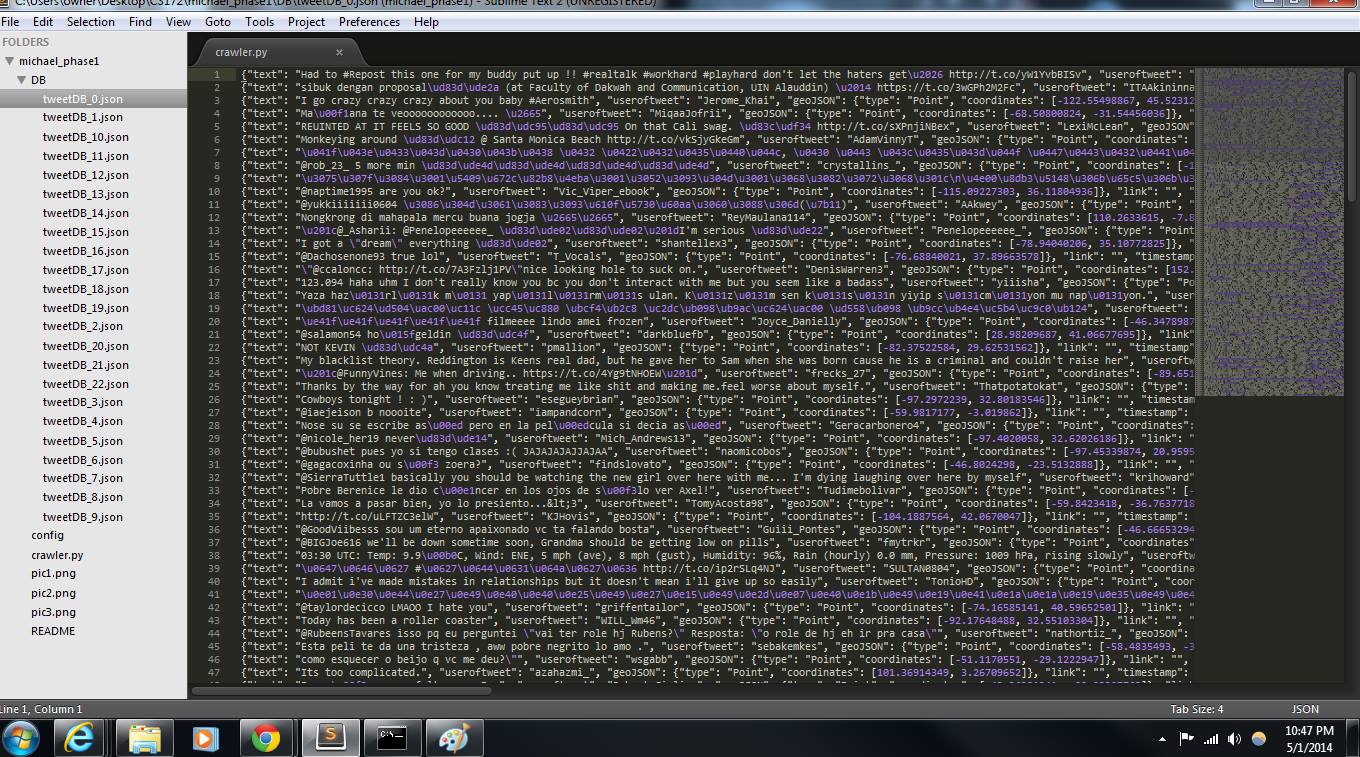
requests (<http://docs.python-requests.org/en/latest/>)

And then runs our twitter.py program

**SCREENSHOTS**



****

****