Assignment 2 Report

Professor Nelson

By Tyler Medina

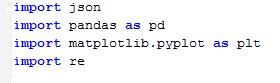
2/09/17

**1. Part 1**

**1.1 Libraries**



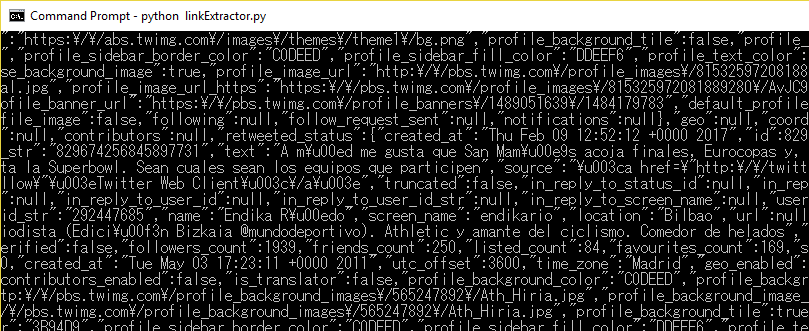
The tweepy library is used to connect to the Twitter API to allow for streaming and downloading data. StreamListener prints the tweets that are streamed and the OAuthHandler/Stream libraries handles the authentication of access tokens and consumer keys to connect to the Twitter streaming API.



The pandas library take the data and puts it into a dataframe to allow for manipulation. The data streamed from the twitter API comes in json format. To get the data we need from the Twitter stream, we need to use the json library to parse the data.

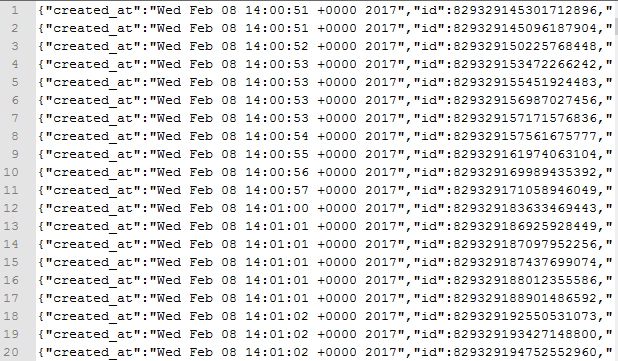
**1.2 Getting data from Twitter API**

In the linkExtractor.py file, the on\_data function prints the data from the Twitter API. The data is printed in an if statement that is finished when the count reaches 999(allowing for 1000 tweets). The keyword to search was “superbowl” with the stream.filter(track=[‘superbowl’]). There should be relative ease gathering the data since the superbowl happened under a week ago. When run, the program begins gathering data that looks like this:

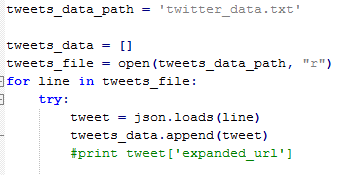


This is a large volume of data. It’s very hard to find the information we need with our eyes. The next step in the process is to parse this data to get the urls mentioned in tweets.

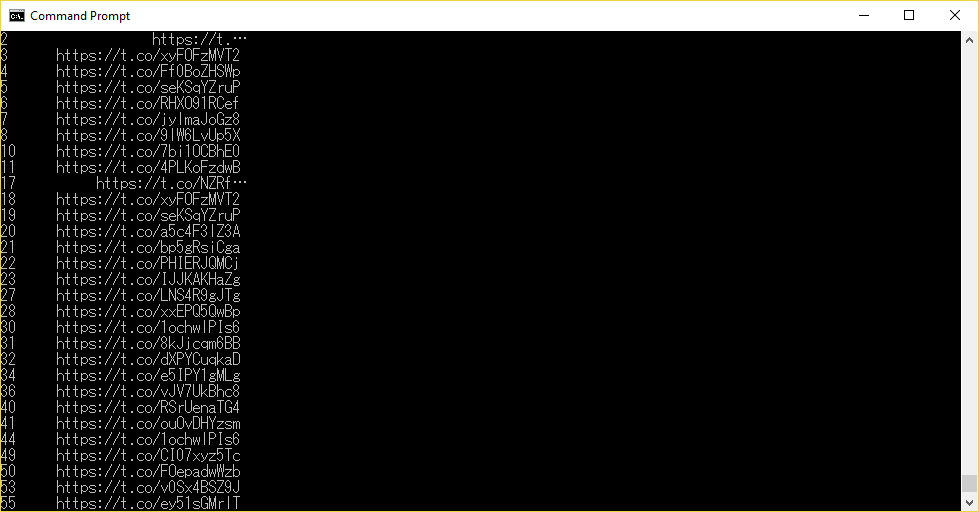
**1.3 Parsing with json**



The data in the above figure is output in json format. To parse this data and extract the bits we need we use the json library. The json output was put in a file called twitter\_data.txt. The file is opened and then read line by line. The data is put into an array called tweets.



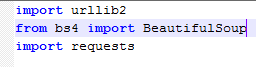
We then use regular expressions to find links that start with ”https://”. After the search finds the uri, in then returns it in the function. Link is pushed into the dataframe and can now be printed out, yielding:



**Part 2**

**1.1 Libraries**

The requests and urllib2 libraries are imported to get data from the html. BeautifulSoup is to search for mementos in the html.



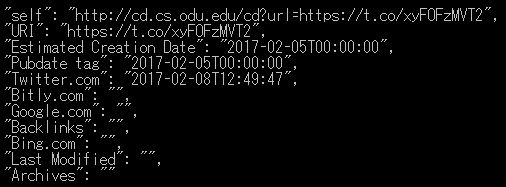
**1.2 Downloading the timemaps**

To get the timemaps of the target uris, the file containing the uris is opened. From here, the uris are read in one line at a time. Urllib2 is used to concatenate the ODU Memento Aggregator and the uri. Beautiful soup scans the html and finds instances or the “rel” tag. From here I couldn’t figure out how to get the number of mementos. I would try counting the number “rel” tags to get the number or mementos, but kept running into errors.

**Part 3**

**1.1 Estimating creation date**

Part 3 is very similar to part two in getting the data needed for the creation date. By concatenating the uri with the carbon dating tool, the desired data is outputted.



This output of one of the links seems accurate because the estimated creation date was on Sunday February 5th. Since the keywords used to extrapolate data was superbowl, the estimated date is on the same day the superbowl took place.