

Assignment 2 Report

Professor Nelson

By Tyler Medina

2/09/17

1. Part 1

1.1 Libraries

```
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream
```

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.

```
import json
import pandas as pd
import matplotlib.pyplot as plt
import re
```

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:

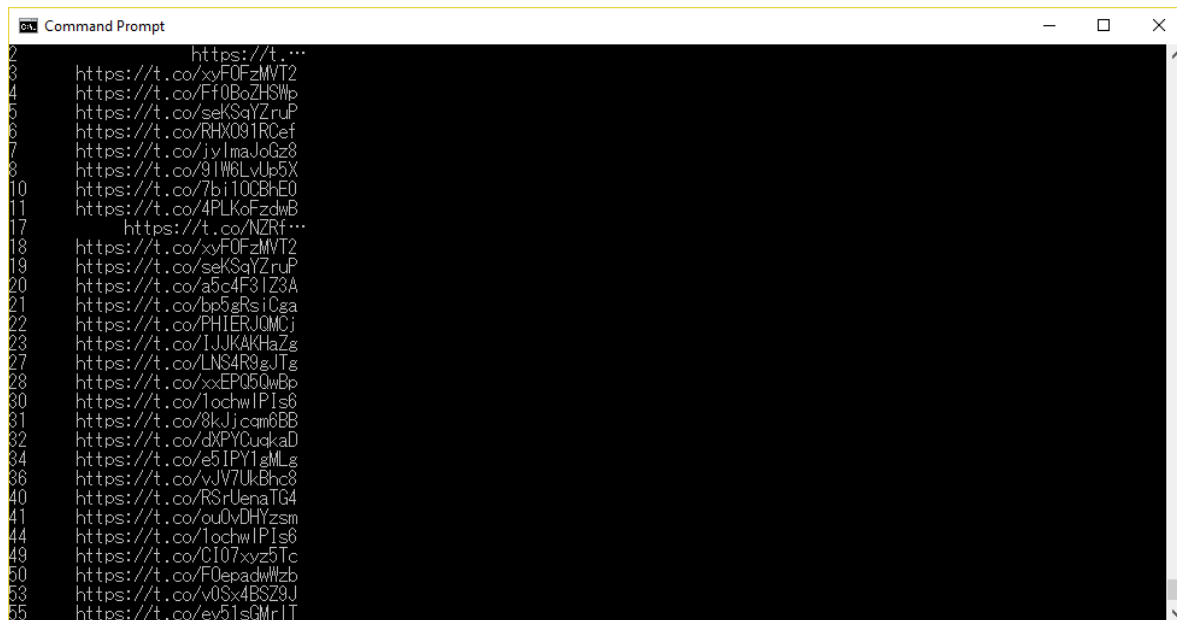

```

tweets_data_path = 'twitter_data.txt'

tweets_data = []
tweets_file = open(tweets_data_path, "r")
for line in tweets_file:
    try:
        tweet = json.loads(line)
        tweets_data.append(tweet)
        #print tweet['expanded_url']

```

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



```

2      https://t...
3      https://t.co/xyF0FzMT2
4      https://t.co/Ff0BoZHSWp
5      https://t.co/sekSqYZruP
6      https://t.co/RHX091RCef
7      https://t.co/jvImaJoGz8
8      https://t.co/9tW6LvUp5X
9      https://t.co/7bi10CBhE0
10     https://t.co/4PLKoFzdWB
11     https://t.co/NZRf...
12     https://t.co/xyF0FzMT2
13     https://t.co/sekSqYZruP
14     https://t.co/a5c4F3IZ3A
15     https://t.co/bp5gRsiCga
16     https://t.co/PHIERJOMCj
17     https://t.co/IJJKAKHaZg
18     https://t.co/LNS4R9gJ1g
19     https://t.co/xxEPQ5QwBp
20     https://t.co/1ochwIP1s6
21     https://t.co/8kJjcm6BB
22     https://t.co/dXPYCuKaD
23     https://t.co/e51PY1gMLg
24     https://t.co/vJV7UkBhc8
25     https://t.co/RSrUenaTG4
26     https://t.co/ou0vDHYzsm
27     https://t.co/1ochwIP1s6
28     https://t.co/C107xyZ5Tc
29     https://t.co/F0epadwWzb
30     https://t.co/v0Sx4BSZ9J
31     https://t.co/ev51sGMrIT

```

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.

```
import urllib2
from bs4 import BeautifulSoup
import requests
```

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 of 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 of 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.

```
"self": "http://cd.cs.odu.edu/cd?url=https://t.co/xyFOFzMVT2",
"URI": "https://t.co/xyFOFzMVT2",
"Estimated Creation Date": "2017-02-05T00:00:00",
"Pubdate tag": "2017-02-05T00:00:00",
"Twitter.com": "2017-02-08T12:49:47",
"Bitly.com": "",
"Google.com": "",
"Backlinks": "",
"Bing.com": "",
"Last Modified": "",
"Archives": ""
```

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.

