Introduction to Web Sciences : Assignment 2
Prof. Anwala

Dontavus Riddick
Sunday, February 11, 2018

Table of Contents

Problem 1

```
Write a Python program that extracts 1000 unique links from Twitter. Omit links
from the Twitter domain (twitter.com).You might want to take a look at:

https://pythonprogramming.net/twitter-api-streaming-tweets-python-tutorial/
http://adilmoujahid.com/posts/2014/07/twitter-analytics/

see also:

http://docs.tweepy.org/en/v3.5.0/index.html
https://github.com/bear/python-twitter
https://dev.twitter.com/rest/public

But there are many other similar resources available on the web.
Note that only Twitter API 1.1 is currently available; version 1
code will no longer work.

Also note that you need to verify that the final target URI (i.e.,
the one that responds with a 200) is unique.  You could have many
different shortened URIs for www.cnn.com (t.co, bit.ly, goo.gl,
etc.).  For example:

$ curl -IL --silent https://t.co/DpO767Md1v | egrep -i "(HTTP/1.1|^location:)"
HTTP/1.1 301 Moved Permanently
location: https://goo.gl/40yQo2
HTTP/1.1 301 Moved Permanently
Location: https://soundcloud.com/roanoketimes/ep-95-talking-hokies-recruiting-one-
week-before-signing-day
HTTP/1.1 200 OK

You might want to use the streaming or search feature to find URIs. If
you find something inappropriate for any reason you see fit, just
discard it and get some more links.  We just want 1000 links that
were shared via Twitter.
```

Solution.

The solution for problem 1 is outlined by the following steps:

After following the prerequisite of this assignment by creating a Twitter API account, I was able to proceed. Only being able to extract 1000 links I was not able to omit the twitter domains from the links nor verify the final target through my python program. See Figure 1 and Figure 2 below to show what I was able to complete.

Fig. 1: JSON file

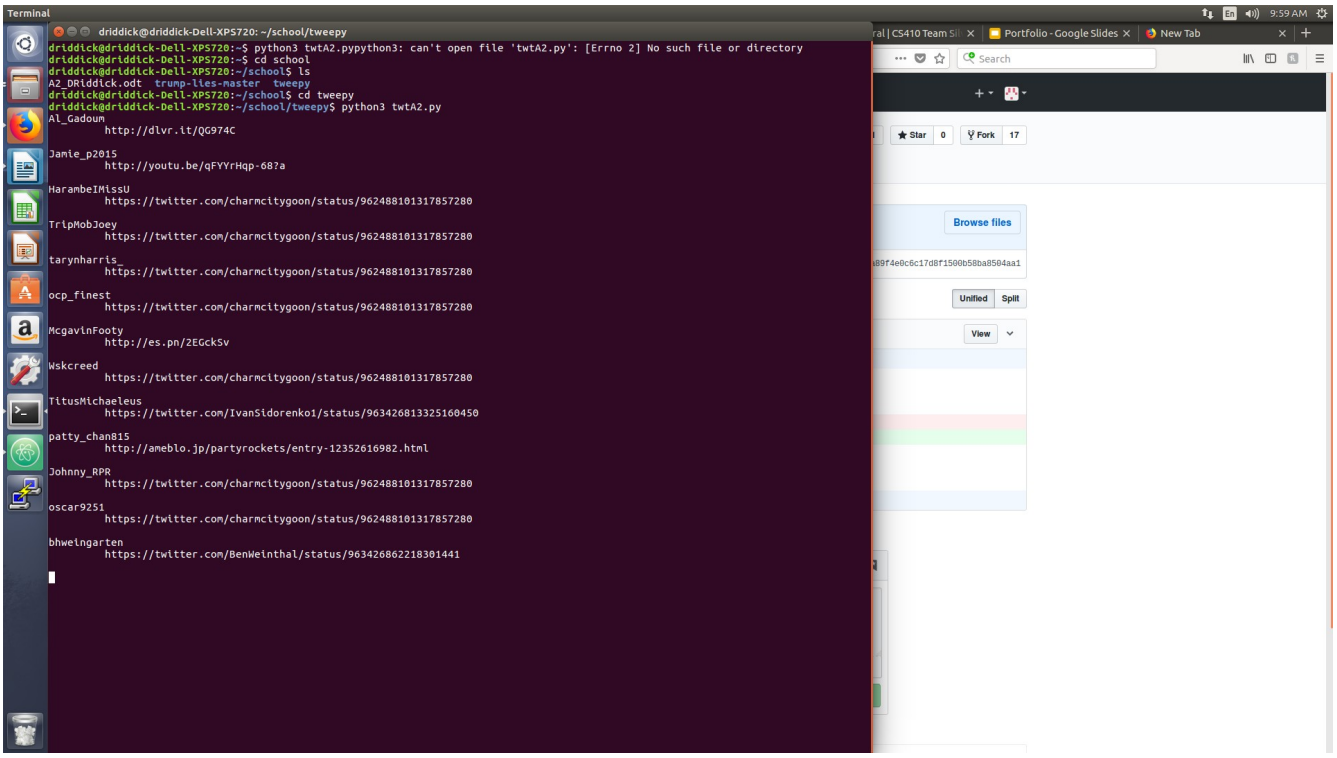

Fig. 2: Python program output of twitter links

Problem 2

```
Download the TimeMaps for each of the target URIs.  We'll use the ODU
Memento Aggregator, so for example:

URI-R = http://www.cs.odu.edu/

URI-T = http://memgator.cs.odu.edu/timemap/link/http://www.cs.odu.edu/

or:

URI-T = http://memgator.cs.odu.edu/timemap/json/http://www.cs.odu.edu/

(depending on which format you'd prefer to parse)

Create a histogram* of URIs vs. number of Mementos (as computed
from the TimeMaps).  For example, 100 URIs with 0 Mementos, 300
URIs with 1 Memento, 400 URIs with 2 Mementos, etc.  The x-axis
will have the number of mementos, and the y-axis will have the
frequency of occurence.

* = https://en.wikipedia.org/wiki/Histogram

What's a TimeMap?
See: http://www.mementoweb.org/guide/quick-intro/
And the week 4 lecture.
```

Solution

N/A

Problem 3

```
Estimate the age of each of the 1000 URIs using the "Carbon
Date" tool:

http://ws-dl.blogspot.com/2017/09/2017-09-19-carbon-dating-web-version-40.html

Note: you should use "docker" and install it locally.  You can do
it like this:

http://cd.cs.odu.edu/cd?url=http://www.cs.odu.edu/

But it will inevitably crash when everyone tries to use it at the
last minute.

For URIs that have > 0 Mementos and an estimated creation date,
create a graph with age (in days) on the x-axis and number of
mementos on the y-axis.

Not all URIs will have Mementos, and not all URIs will have an
estimated creation date.  Show how many fall into either categories.
For example,

total URIs:         1000
no mementos:        137
no date estimate:   212
```

Solution

N/A