Assignment 2

CS595

Introduction to Web Science

Old Dominion University

Computer Science

Due: 11:59 pm Sept 26

Lulwah Alkwai

Question One-

Write a Python program that extracts 1000 unique links from Twitter. You might want to take a look at:

http://thomassileo.com/blog/2013/01/25/usingtwitter-rest-api-v1-dot-1-with-python/

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 different shortened URIs for www.cnn.com.

You might want to use the search feature to find URIs, or you can pull them from the feed of someone famous (e.g., Tim O'Reilly).

Hold on to this collection – we'll use it later throughout the semester.

Answer One-

For this question I used the sample code on http://thomassileo.com/blog/2013/01/25/using-twitter-rest-api-v1-dot-1-with-python/ and did some alteration on it to do the required tasks.

First, I wanted to run the existing example but there were some steps that I had to follow. I had to download http://docs.python-requests.org/en/latest/ which is an Apache2 Licensed HTTP library, written in Python, for human beings. Also, I had to download a package called request https://github.com/requests/requests-oauthlib. Next, from the page https://dev.twitter.com/start I logged in using my twitter account and had to create an application. Which allowed me to create and save a consumer key and consumer secret which I added to my python code.

Then by running the following commands, on UNIX I can see my last mentions:

$ sudo pip install request request_ oauthlib

$wget https://gist.github.com/raw/4637864/9ea056ffbe5bb88705e95b786332ae4c0fd7554c/mytweets.py

python mytweets.py

And by searching the web on redirected links I found a package from this link http://stackoverflow.com/questions/6158895/httplib-is-not-getting-all-the-redirect-codes/11617817#11617817 that I included in my code that allowed me to send a raw link to the function and it would return the final redirected link. To use this package I needed to install a package

called httplib2 `http://code.google.com/p/httplib2/downloads/list` After that, I decided to use the some users time line to extract some links from. I saved some twitter screen names as a variable to take their time lines. And I specified the number of tweets for each person we extracted the tweets from.In my case I set 200 just to be on the safe side.

Next, I opened two files the first is to write the raw links to and the second to write the final redirected links to. And for testing purposes I have created two list to save the same data.

Finally, I search on tutorials on streaming data from twitter using python and I found this wonderful link on streaming dataand I used it as a guide line. Where I created a loop to go through all the different screen names and extract links from their time line, and save it.And then send the link to the function that returns the final redirected link.Also, I showed all responses so I can make sure that the final links are 200 response codes.

So by running the code I can see the raw link the response code, the ultimate link and the response code. In addition to the number of raw links and the number of final links. A file called (rawlinks.txt) and (ultimatelinks.txt) are also created.

By trying the code I have noticed that some links are redirected more than once and it takes some time to get the last redirected link. And some of them simply result to a 404 so we will not save them with the final file.

The final file (ultimatelinks.txt) saved is not unique and since the file is saved in a text file I decided to leave it as it is and redirect to another file that is sorted and unique using the command:

>sort u (ultimatelinks.txt) ¿ (uniq-ultimate-links.txt)

By doing that I will save my work in two files directories so it could be a back up. Also, we can count to see how many links are extracted and how many are filtered using the command:

>wc (uniq-ultimate-links.txt)

Finally, to get only 1000 links I used the following command on the (uniq-ultimate-links.txt) file:

1,1000w final_ links.txt

Code-

Attached are: (a2q1.py): the python code, (rawlinks.txt):the first links gathered, (ultimatelinks.txt):all links unique and sorted, (finl_ links.txt):select only 1000 links

Question Two-

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://mementoproxy.cs.odu.edu/aggr/timemap/link/http://www.cs.odu.edu/`

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.

Answer Two-

To solve this question first we need to write a python code that reads the existing file (from question one) that has all the links, and we need to create another file to write the results in with the first line set as url,memento.

Next, we need to create a new uri that contains ODU Memento Aggregator concatenated with the links in the file.

After that, we loop in the the file line by line by stripping out the new line and run the new url.Then we create a request object and pass it to urlopen() method to open url and check if its code is a successful one.And we create a tokenizer which is based on the Timestamp created by Scott `https://github.com/galsondor/Timemap.py` and we loop in tokens and search for key words:"memento","first memento","last memento","memento first","memento last", "first last memento" and add to memento counter. When exiting the loop we write it to file. However if an HTTP error is returned the memento count is zero.

Code and Histogram-

Attached are: (a2q2.py): the python code, (histogram.txt): the memento count of each url.

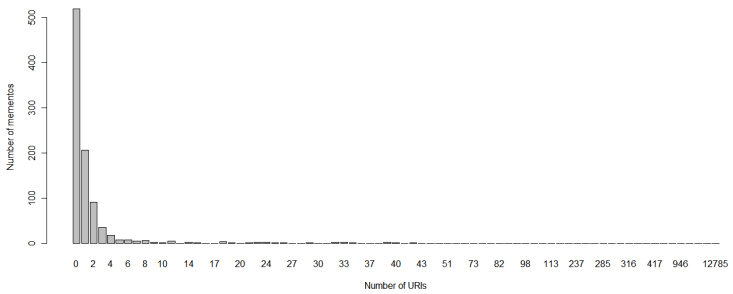

Figure 1: Number of URIs vs Number of Memento

Question Three-

Estimate the age of each of the 1000 URIs using the "Carbon Date" tool:
`http://ws-dl.blogspot.com/2013/04/2013-04-19-carbon-dating-web.html`
Note: you'll have to download the tool and install; don't try to use the web service.

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

Answer Three-

In the code I used three files, the first is the one is needed to read the existing number of memento from, the second is a temporary file that extract the carbon tool result and save it all.The third is to write final result to. The initial carbon date tool can be found at `https://github.com/LulwahAlkwai/CarbonDate`.

First, I saved all result of tool in a variable which is transferred to second file, then we read file and look for the estimated date and save it and convert it to a date format.

To calculate age I subtracted (today's date),the day I ran the code, from the (estimate). Finally, I saved the result in the file as days format.

Note: I used flush to view the results as it flows.

Code and Graph-

Attached are: (a2q3.py): the python code (histogram.txt): the memento count of each url(from answer2) (histogram2.txt):the estimate age of each url (histogram2 _ nZeroMemento.txt):the estimate age of each url that have no zero values

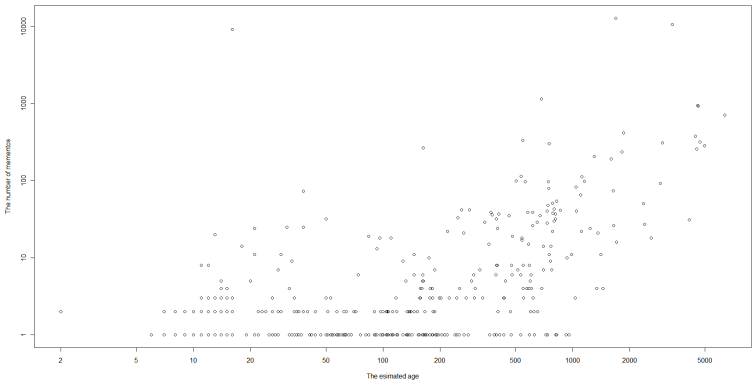

Figure 2: Number of Memento vs The Estimate Age