CS 595: Assignment 2

Mallika Kogatam

Fall 2014

Contents

1 Problem 1

1. 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/using-twitter-rest-api-v1-dot-1-with-python/

2. 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.

3. 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. For example,

   http://cnn.it/1cTNZ3V
   http://t.co/BiYdsGotTd

4. Both ultimately redirect to cnn.com, so they count as only 1 unique URI. Also note the second URI redirects twice – don't stop at the first redirect.

   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.

1.1 Solution

Extracting 1000 unique links from Twitter was more complex endeavor than expected.

The following steps were taken in order to get the links

1. First the key and secret keys for OATH and API are collected by registering an Application in Twitter with an account.

2. TwitterSearch package is used to get the URI's from tweets for a specific keyword.

3. All the URI's that are present in the tweets got from the search for a particular key word are put in an array set. By keeping it in a set eliminates the redundant entries.

4. Then checked whether the Url's are redirected to same website or not and also checked whether the extracted url is working or not.

5. To get the URLs form tweets it take so long and apart from that i faced the 429 error which says "too many requests", so i will have to waiti for 15 minutes to run the next time.

6. When I searched for 5000 urls form the tweets I got only 480 unique links out of them. so I kept incrasing the count to get 1000 unique links

7. When I increased the count to 17000 I got 1500 unique links and this script took almost 40 minutes to run.

8. I could not even split the program into two different script as there might be repetition in the output got from both the scripts so I will have to run it in a single script.

9. This way collected 1000 URI's for keyword "twitter".

1.2 Code Listing

Here is the python code that is used to collect Thousand URL’s from twitter.

```
#! /usr/bin/env python

from TwitterSearch import *
import unittest
import urllib2
import re
import sys
import socket
import ssl
CONSUMER_KEY = ”1QjOMB7IZ00zaTvPmZ6tUrR2R”
CONSUMER_SECRET = ”4aolvqxZYw3XwyrzWjiHb6aFAlg7iNodyxmyiIyX8NowefLL53”
OAUTH_TOKEN = ”2820592280-iVQDNegTG1CtQLuQ14N9kwIefiVPSTA75zAKeye”
OAUTH_TOKEN_SECRET = ”oaUFcXLImxaL0XGUvy3nT1XfBXkVkLNWDmRHkWvvDCozi”

def main():
    tso = TwitterSearchOrder()           # create a TwitterSearchOrder object
    tso.setKeywords(['movies'])          # let's define all words we would like to have a look for
    tso.setLanguage('en')                # we want to see English tweets only
    tso.setCount(100)                    # only give us 100 results per page
    tso.setIncludeEntities(False)        # and don't give us all those entity information

    ts = TwitterSearch(
        consumer_key        = CONSUMER_KEY,
        consumer_secret     = CONSUMER_SECRET,
        access_token        = OAUTH_TOKEN,
        access_token_secret = OAUTH_TOKEN_SECRET
    )
    results = 0
    url_list = set()
    counter = 1
    for tweet in ts.searchTweetsIterable(tso):

        try:
            #link = tweet['retweeted_status']['user']['entities']['url']['urls']
            for link in tweet['retweeted_status']['user']['entities']['url']['urls']:
                # TwitterSearch package had a great advantage of using json text.
                # Found out the json structure of each search got
                #from the searchTweetIterable function and then retrieved only the urls.
                new_url = link[”expanded_url”]
                # Extracted the expanded url instead or url(shortened url) from the json structure

                url_list.add( new_url )
            counter= counter+1
            if counter > 17200:#getting 17200 urls from the json (not unique )
                break
        except KeyError:
            pass
    processed_urls = []
    for url in set(url_list):#by using set function it eliminates the redundancies from the url_list
        try:
            response = urllib2.urlopen(url, timeout = 2)
            first = response.geturl()# tried to accept only the urls which get the 200 ok but some how i ended with too many errors
            #second = urllib2.urlopen(first)
            #print second.geturl()
            #print 'RESPONSE:', response.info()[”status”]
            #if ”200 OK” in response.info()[”status”]:
            #    print response.geturl()

            while (url != first):#checked if the url got from the tweet and the response for the url is not same
                url = first
                first = (urllib2.urlopen(first, timeout = 2)).geturl()
            processed_urls.append( first )# checks till it is same and then appended to proceed_urls
        except urllib2.HTTPError as e:# if we get an http error then dump that url
```

```
            pass
        except urllib2.URLError:#dump the url if got these error
            pass
        except socket.timeout:
            pass
        except ssl.SSLError:
            pass
        except socket.error:
            pass
        except AttributeError:
            pass
    processed_urls = set( processed_urls )


    for url in processed_urls:
        print url

    print len( processed_urls )

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

Listing 1: Python program for acquiring 1000 unique links for a given keyword

1.3 Results

```
http://taracastle.tumblr.com
http://WELOVECHAERIN.COM
http://fuck-thatsdelicious.tumblr.com/
http://www.hallmarkbaby.com
http://artindigenous.wordpress.com/
http://iswimwithissues.net/
https://www.facebook.com/AbhayRamNandamurifanz
http://www.redsrugby.com.au/
http://ask.fm/zayncutestgifs
http://robbydonoho.weebly.com
http://rachelinlondon.wordpress.com/
http://LILDEBBIE.ORG
https://www.facebook.com/SneakerShouts
http://www.kk-allen.com
http://www.youtube.com/user/TheGabbieShow
http://instagram.com/yoteensig
https://soundcloud.com/travisscott-2/04-drugs-you-should-try-it
https://www.facebook.com/PupsPorn
http://goosepancake.tumblr.com
http://fox59.com/
http://www.abc.net.au/mediawatch/
http://greatsss.com
http://ultimate-dickhead.tumblr.com/
http://tweetsta.com/index.php
http://itssongoku.tumblr.com
https://twitter.com/zilliamson
http://socialistworker.co.uk
http://www.enzozelocchi.com
http://www.youtube.com/watch?v=VAPIOciXs6A&feature=youtu.be
http://www.rageon.com/collections/pokemon/?rageon.com
http://www.youtube.com/user/FreeFallingHQ
http://www.youtube.com/watch?v=frNsJvV5zAs&feature=youtu.be
http://cespedesfamilybarbecue.com/about/
http://www.natethehitmaker.com/
http://ImMackenson.com
http://anhonestyear.com/
http://www.romeolacoste.com
http://smarturl.it/RFLT
http://proguitarshop.com/
http://ohholybutt.tumblr.com
http://Instagram.com/lawsofsex
http://www.photografyum.com
http://www.hrtv.com
http://sensualgif.tumblr.com/
http://kushoverboys.tumblr.com
http://www.ew.com/ew/
https://31.media.tumblr.com/d0c4823d0bb16aa36fb99fba8b366140/tumblr_n7bi03tXo51t7eg8go3_250.gif
http://english.alarabiya.net/
https://twitter.com/nationofbiebs/status/457342650874474496
http://www.womenshealthmag.com
```

2 Problem 2

Download the TimeMaps for each of the target URIs. We'll use the mementoweb.org Aggregator, so for example:

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

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

You could use the cs.odu.edu aggregator:

URI-T = http://mementoproxy.cs.odu.edu/aggr/timemap/link/1/http://www.cs.odu.edu/

But be sure to say which aggregator you use -- they are likely to give different answers.

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.

See: http://en.wikipedia.org/wiki/Histogram

Note that the TimeMaps can span multiple pages. Look for links like:

```
<http://mementoweb.org/timemap/link/1000/http://www.cnn.com/>;rel="timemap";
type="application/link-format"; from ="Sun, 08 Jul 2001 21:30:54 GMT"
```

This indicates another page of the TimeMap is available. There can be many pages to a TimeMap.

2.1 Solution

1. In order to compute the mementos for a link we need the TimeMap for each link from the file containing many links which is an out from the first question.

2. I used "http://mementoweb.org/timemap/link/" because it worked better than http://mementoproxy.cs.odu.edu/aggr/t the mementoproxy provided cs department was very slow.

3. After finding the Time Map for each link parse them and count the mementos.

4. When the request that generate a 404 are recorded as having 0 mementos.and these mementos are counted using a regular expression.

5. I have written Two functions named getTimeMap*url* which get the timemaps for each link provided and countMementos*mem_url* which counts the mementos for each timemap generated by the other function.

6. There can be many pages to a TimeMap.so written a while loop which checks for any timemaps which directs to an other page and found the total count of mementos

7. Finally extracting the mementos and the url name.

2.2 Code Listing

mementocount.py

```
#!/usr/local/bin/python3
import re
import sys
import urllib2
import collections
#Regular expression that is used to count the memento from the
#TimeMap for each link
mementostructure = re.compile(r'rel.*?=.*?"memento".*?')
#Regular expression to find another timemap in the one timemap page.
speciallink = re.compile(r'<.+>;rel=.*?"timemap"')
def getTimeMap(url, prepend=True):
    if prepend:
        mem_url = "http://mementoweb.org/timemap/link/" + url
        #appending the url fro the mementoweb.org in order to get the TimeMap
    else:
        mem_url = url
    try:
        response = urllib2.urlopen(mem_url)
        timemap = response.read()#finding time map for thr url provided
    except urllib2.HTTPError:#if the link gives 404 then make the timemap none
        timemap= None #this way we are not missing the mementos for the links which doesnot have timemap
    return timemap

def countMementos(mem_url):
    time_map = getTimeMap(mem_url)

    if not time_map:#gives the memento as 0 if the timemap is none
        count = 0
    else:

        count = len(mementostructure.findall(str(time_map)))
        special_url = speciallink.findall( str(time_map) )

        while len( special_url ) == 1:#checks if there is an another timemap in the existing timemap
            getlink=re.findall("(<.*?>)",special_url[0].strip())#get the next link by a regular expression
            mem_url = getlink[1][1:-1]#discards the angular tags
            time_map = getTimeMap(mem_url, False)#pass the url to getTimeMap function and do not append to mementoweb
            count += len( mementostructure.findall(str(time_map )) )#appending the new count it to existing count
            special_url = speciallink.findall( str(time_map) )
            #assigning this link to special_url so that loop continues till the timemaps does not exits
    return count

if __name__ == "__main__":

    f = open ('listUrlCdate.txt','r')
    memlist = []
    for line in f.readlines():
        mementoCount = countMementos(line.strip())
        memlist.append(mementoCount)
        i = (str(mementoCount), line.strip())
        numbers = "\t".join(str(x) for x in i)
        print numbers
        sys.stdout.flush()

    f.close()
```

Listing 2: Python program for counting mementos

histogram.R

```r
#!/usr/bin/Rscript

d = read.csv( "mementos.txt", stringsAsFactors=F, header = FALSE, sep = "\t" )

Mementos = d[,1]

brk <- seq(0, 48375, 1)

png("q2-histogram1.png")
hist(Mementos, col=heat.colors(48375), main = "URIs vs. Number of Mementos", breaks=brk, freq = T, xlab="Mementos", ylab="URIs")

dev.off()

Mementos = Mementos[which(Mementos<1000)]


brk <- seq(0, 1000, 1)

png("q2-histogram2.png")
hist(Mementos, col=heat.colors(1000), main = "URIs vs. Number of Mementos", breaks=brk, freq = T, xlab="Mementos", ylab="URIs")

dev.off()

Mementos = Mementos[which(Mementos<200)]

brk <- seq(0, 200, 3)

png("q2-histogram3.png")
hist(Mementos, col=heat.colors(200), main = "URIs vs. Number of Mementos", breaks=brk, freq = T, xlab="Mementos", ylab="URIs")

dev.off()
```

Listing 3: R program for generating the histograms for Question 2

1. Initial I created a histogram with the output i got from the countmementos.py but the graph looked like in the Figure 1 in page 15

2. There is

3. There are 500 links with 0 mementos and 90 links with more than 1000 mementos and 59 links with 1 memento and 30 links with 2 and other links have mementos mostly under 500. which means majority of the links fall under 1000 mementos.

4. When plotted the histogram for the above data we get Figure 1 which does not give any proper visualization

5. I am not satisfied with the figure 1, so i made few changes in order get a decent histogram.

6. So i stripped out the mementos which are grater than 1000 as there are only 90 so this data gives Figure 2 in page 10 which seems little better than the previous one

7. If we focus more on the highest number of records,mementos less than 200 then the plot actually begins to look more like a histogram in Figure in page 11

2.3 Results

Output for the mementocount.py

```
4 http://taracastle.tumblr.com
211 http://WELOVECHAERIN.COM
1 http://fuck-thatsdelicious.tumblr.com/
10 http://www.hallmarkbaby.com
3 http://artindigenous.wordpress.com/
0 http://iswimwithissues.net/
0 https://www.facebook.com/AbhayRamNandamurifanz
218 http://www.redsrugby.com.au/
0 http://ask.fm/zayncutestgifs
1 http://robbydonoho.weebly.com
30 http://rachelinlondon.wordpress.com/
39 http://LILDEBBIE.ORG
0 https://www.facebook.com/SneakerShouts
0 http://www.kk-allen.com
0 http://www.youtube.com/user/TheGabbieShow
0 http://instagram.com/yoteensig
0 https://soundcloud.com/travisscott-2/04-drugs-you-should-try-it
0 https://www.facebook.com/PupsPorn
1 http://goosepancake.tumblr.com
1657 http://fox59.com/
729 http://www.abc.net.au/mediawatch/
0 http://greatsss.com
1 http://ultimate-dickhead.tumblr.com/
0 http://tweetsta.com/index.php
0 http://itssongoku.tumblr.com
2 https://twitter.com/zilliamson
1314 http://socialistworker.co.uk
21 http://www.enzozelocchi.com
0 http://www.youtube.com/watch?v=VAPI0ciXs6A&feature=youtu.be
0 http://www.rageon.com/collections/pokemon/?rageon.com
0 http://www.youtube.com/user/FreeFallingHQ
0 http://www.youtube.com/watch?v=frNsJvV5zAs&feature=youtu.be
8 http://cespedesfamilybarbecue.com/about/
0 http://www.natethehitmaker.com/
14 http://ImMackenson.com
3 http://anhonestyear.com/
8 http://www.romeolacoste.com
0 http://smarturl.it/RFLT
473 http://proguitarshop.com/
0 http://ohholybutt.tumblr.com
1 http://Instagram.com/lawsofsex
0 http://www.photografyum.com
378 http://www.hrtv.com
0 http://sensualgif.tumblr.com/
0 http://kushoverboys.tumblr.com
7329 http://www.ew.com/ew/
0 https://31.media.tumblr.com/d0c4823d0bb16aa36fb99fba8b366140/tumblr_n7bi03tXo51t7eg8go3_250.gif
2786 http://english.alarabiya.net/
0 https://twitter.com/nationofbiebs/status/457342650874474496
1620 http://www.womenshealthmag.com
```

Histograms

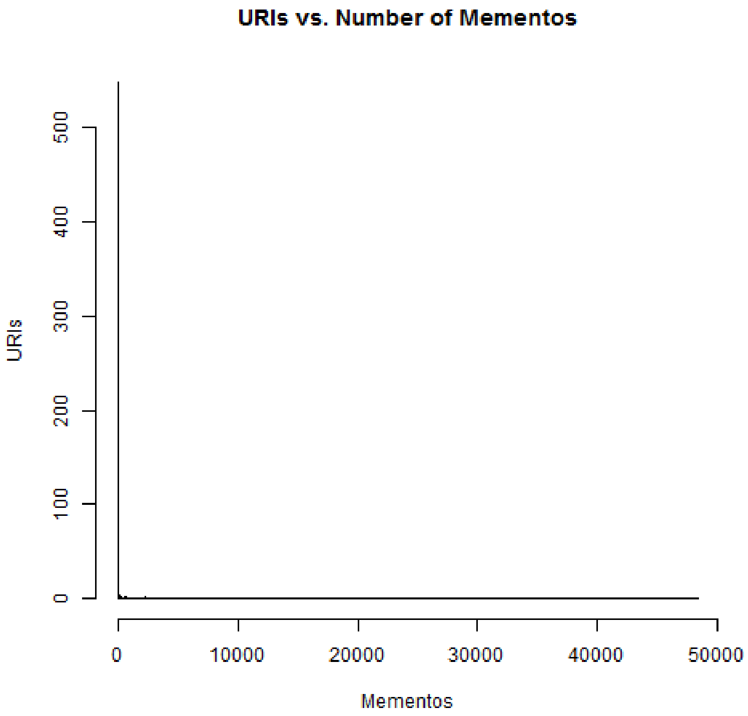

Figure 1: Histogram 1

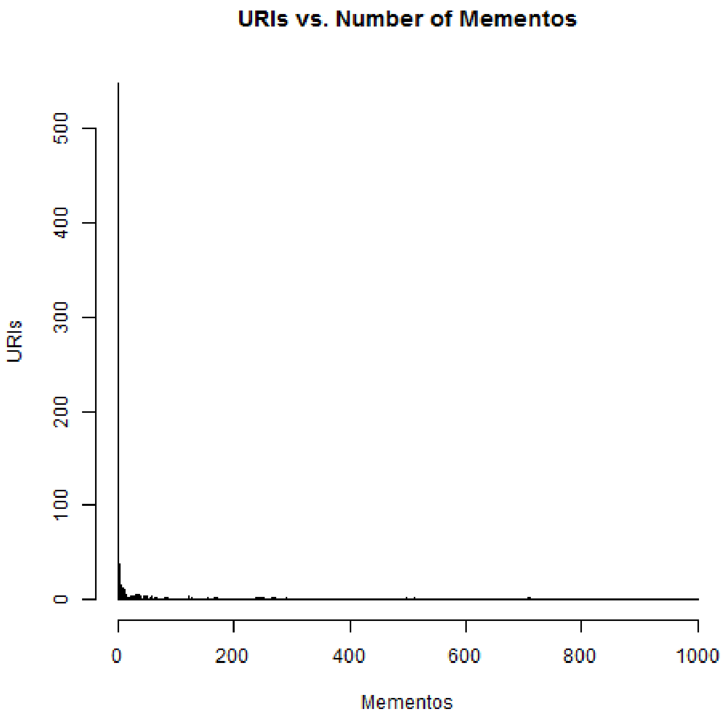

Figure 2: Histogram 2

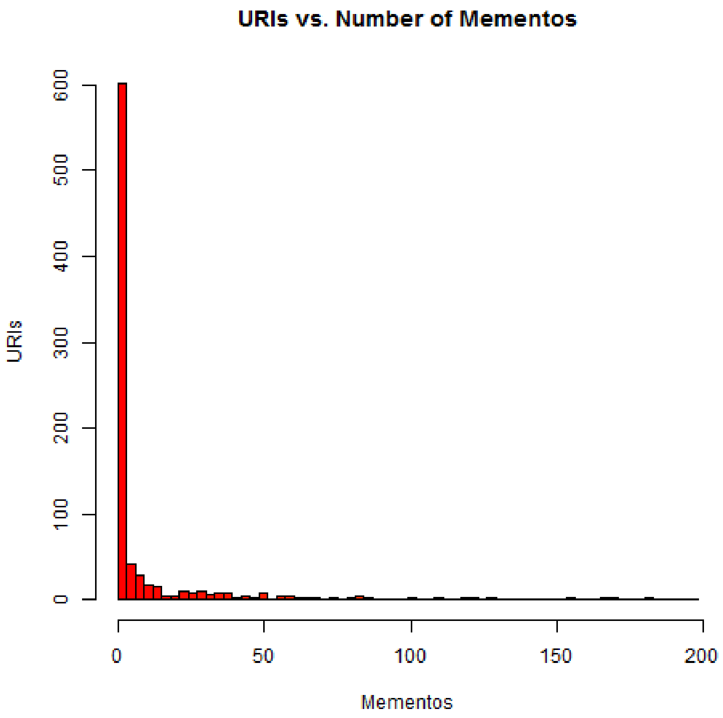

Figure 3: Histogram 3

3 Problem 3

```
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 better luck downloading and installing the tool
rather than using the web service (which will run slowly and likely
be unreliable).

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.
```

3.1 Solution

1. Downloaded the carbon tool provided and installed as instructed in the readme file.
2. From some test runs, it became apparent that the Carbon Date tool takes between 1 and 7 minutes to query all of its services for a given URI.
3. So getting the dates for all the 1000 URI's from a single script will take more than 36 hours so i executed 5 scripts in 5 different Linux servers which let me extract all the links in 12 hours.
4. Made changes to local.py from the carbon tool in order to read 1000 URI's from a text file and write back the result(time,URI) into an other text file
5. Getting the dates was not enough, because we need the age of each link.
6. I have written a small python program(caldays.py) which will take the date got from the Carbon Date tool and calculated the days
7. We will have to create a graph for the links which have ¿0 mementos and the days for that link
8. So inorder to filter the links which have mementos¿0 and get the days, i have written an other simple python program memvsdays.py.
9. memvsdays.py will collect the data which we need to build a graph for URIs that have ¿ 0 Mementos and the days .

3.2 Code Listing

local.py

```
import json
from ordereddict import OrderedDict
import json as simplejson
import re
from getBitly import getBitlyCreationDate
from getArchives import getArchivesCreationDate
from getGoogle import getGoogleCreationDate
from getBacklinks import *
from getLowest import getLowest
from getLastModified import getLastModifiedDate
from getTopsyScrapper import getTopsyCreationDate
from htmlMessages import *
from pprint import pprint
from threading import Thread
import Queue
import datetime

import sys, traceback


fh = open("lin_uniqUrl1.txt",'r')#opens the file which have 1000 unique URI's

for line in fh:
        url=line
        url=url.replace('\n','')


        def cd(url):
            print 'Getting Creation dates for: ' + url

            threads = []
            outputArray =['','','','','','']
            now0 = datetime.datetime.now()


            lastmodifiedThread = Thread(target=getLastModifiedDate, args=(url, outputArray, 0))
            bitlyThread = Thread(target=getBitlyCreationDate, args=(url, outputArray, 1))
            googleThread = Thread(target=getGoogleCreationDate, args=(url, outputArray, 2))
            archivesThread = Thread(target=getArchivesCreationDate, args=(url, outputArray, 3))
            backlinkThread = Thread(target=getBacklinksFirstAppearanceDates, args=(url, outputArray, 4))
            topsyThread = Thread(target=getTopsyCreationDate, args=(url, outputArray, 5))


            # Add threads to thread list
            threads.append(lastmodifiedThread)
            threads.append(bitlyThread)
            threads.append(googleThread)
            threads.append(archivesThread)
            threads.append(backlinkThread)
            threads.append(topsyThread)


            # Start new Threads
            lastmodifiedThread.start()
            bitlyThread.start()
            googleThread.start()
            archivesThread.start()
            backlinkThread.start()
            topsyThread.start()


            # Wait for all threads to complete
            for t in threads:
                t.join()

            # For threads
            lastmodified = outputArray[0]
            bitly = outputArray[1]
            google = outputArray[2]
            archives = outputArray[3]
            backlink = outputArray[4]
            topsy = outputArray[5]

            #note that archives["Earliest"] = archives[0][1]
            try:
                lowest = getLowest([lastmodified, bitly, google, archives[0][1], backlink, topsy]) #for thread

            except:
               print sys.exc_type, sys.exc_value , sys.exc_traceback
            result = []

            result.append(("URI", url))
            result.append(("Estimated Creation Date", lowest))
            result.append(("Last Modified", lastmodified))
            result.append(("Bitly.com", bitly))
            result.append(("Topsy.com", topsy))
            result.append(("Backlinks", backlink))
            result.append(("Google.com", google))
            result.append(("Archives", archives))
            values = OrderedDict(result)
            r = json.dumps(values, sort_keys=False, indent=2, separators=(',', ': '))

            now1 = datetime.datetime.now() - now0


            #print "runtime in seconds: "
            #print now1.seconds
            #print r
            #$print 'runtime in seconds:  ' +  str(now1.seconds) + '\n' + r + '\n'
            k = str(now1.seconds) + '\n' + r
            i = lowest
            print url + "" +i
            #print i
            saveFile = open("li_cdate1.txt",'a')#saving the date and URI to a new file
            saveFile.write( "{:<20} {}\n".format(lowest, url) )
            #saveFile.write('\n')
            saveFile.close()
            return r
        cd(url)

if len(sys.argv) == 1:
    print "Usage: ", sys.argv[0] + " url (e.g: " + sys.argv[0] + " http://www.cs.odu.edu)"
elif len(sys.argv) == 2:
    #fix for none-thread safe strptime
    #If time.strptime is used before starting the threads, then no exception is raised (the issue may thus come from #strptime.py not being imported in a thread safe manner). -- http://bugs.python.org/issue7980
    time.strptime("1995-01-01T12:00:00", '%Y-%m-%dT%H:%M:%S')
    cd(sys.argv[1])
```

Listing 4: Python program for getting creation date for URI's

cladays.py

```
#!/usr/local/bin/python3

import sys
import time
import datetime

#carbonDate1 = sys.argv[1]

file = open(”cDate.txt”,’r’)

for line in [l.strip() for l in file if l.split()]:

    try:
        #print line.split()
        data = line.split()
        if len(data) != 2:
            print 0, data[0]
            #data = (datetime.now(),data)
        else:
            (cdate, uri) = data
            ct = time.strptime(cdate, ”%Y–%m–%dT%H:%M:%S”)

            cdt = datetime.datetime.fromtimestamp(time.mktime(ct))
            now = datetime.datetime.now()
            days = (now − cdt).days
            print(str(days) + ’ ’ + uri)
    except ValueError:
        # skip over those items without carbon dates
        print (’0’+ ’ ’ + uri)

file.close()
```

Listing 5: Python program for calculating the age of URI’s

memvsdays.py

```python
import sys

def main():
    mementoData = {}
    ageData = {}

    mem_urls = open('memUrlCdate.txt','r')

    for line in mem_urls:
        line = line.strip()
        (mementoCount, uri) = line.split('\t')

        if int(mementoCount) > 0:
            mementoData[uri] = mementoCount

    mem_urls.close()

    carbon_date = open('cDays.txt','r')

    for line in carbon_date:
        line = line.strip()
        (age, uri) = line.split(' ')

        ageData[uri] = age

    carbon_date.close()
    for key in mementoData:
        print(key + ' ' + mementoData[key] + ' ' + ageData[key])

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

Listing 6: Python program for collecting Mementos versus days

q3graph.R

```r
#!/usr/bin/Rscript

mems_vs_days <- read.csv("dayvsmem.txt", stringsAsFactors = F, header = FALSE, sep = " ")

data = mems_vs_days[,c(2,3)]

png("q3-scatterplot.png")

plot(data, col=c("blue"), ylab="Age (in days)", xlab="Number of Mementos", main = "Number of Mementos vs. Age of URI")

dev.off()
```

Listing 7: R program for generating the histograms for Question 2

3.3 Results

carbondates.txt

```
http://taracastle.tumblr.com
http://WELOVECHAERIN.COM
http://fuck-thatsdelicious.tumblr.com/
http://www.hallmarkbaby.com
http://artindigenous.wordpress.com/
http://iswimwithissues.net/
https://www.facebook.com/AbhayRamNandamurifanz
http://www.redsrugby.com.au/
http://ask.fm/zayncutestgifs
http://robbydonoho.weebly.com
http://rachelinlondon.wordpress.com/
http://LILDEBBIE.ORG
https://www.facebook.com/SneakerShouts
http://www.kk-allen.com
http://www.youtube.com/user/TheGabbieShow
http://instagram.com/yoteensig
https://soundcloud.com/travisscott-2/04-drugs-you-should-try-it
https://www.facebook.com/PupsPorn
http://goosepancake.tumblr.com
http://fox59.com/
http://www.abc.net.au/mediawatch/
http://greatsss.com
http://ultimate-dickhead.tumblr.com/
http://tweetsta.com/index.php
http://itssongoku.tumblr.com
https://twitter.com/zilliamson
http://socialistworker.co.uk
http://www.enzozelocchi.com
http://www.youtube.com/watch?v=VAPIOciXs6A&feature=youtu.be
http://www.rageon.com/collections/pokemon/?rageon.com
http://www.youtube.com/user/FreeFallingHQ
http://www.youtube.com/watch?v=frNsJvV5zAs&feature=youtu.be
http://cespedesfamilybarbecue.com/about/
http://www.natethehitmaker.com/
http://ImMackenson.com
http://anhonestyear.com/
http://www.romeolacoste.com
http://smarturl.it/RFLT
http://proguitarshop.com/
http://ohholybutt.tumblr.com
http://Instagram.com/lawsofsex
http://www.photografyum.com
http://www.hrtv.com
http://sensualgif.tumblr.com/
http://kushoverboys.tumblr.com
http://www.ew.com/ew/
https://31.media.tumblr.com/d0c4823d0bb16aa36fb99fba8b366140/tumblr_n7bi03tXo51t7eg8go3_250.gif
http://english.alarabiya.net/
https://twitter.com/nationofbiebs/status/457342650874474496
http://www.womenshealthmag.com
```

carbonday.txt

```
2012-05-29T00:00:00  http://taracastle.tumblr.com
2013-05-02T14:02:10  http://WELOVECHAERIN.COM
2012-10-31T00:00:00  http://fuck-thatsdelicious.tumblr.com/
2009-08-01T00:00:00  http://www.hallmarkbaby.com
2010-04-14T00:00:00  http://artindigenous.wordpress.com/
2012-04-21T00:00:00  http://iswimwithissues.net/
2014-06-05T00:00:00  https://www.facebook.com/AbhayRamNandamurifanz
2014-09-24T03:50:51  http://www.redsrugby.com.au/
2013-12-25T00:00:00  http://ask.fm/zayncutestgifs
2010-10-22T00:00:00  http://robbydonoho.weebly.com
2011-01-31T00:00:00  http://rachelinlondon.wordpress.com/
2011-07-05T00:00:00  http://LILDEBBIE.ORG
2011-05-23T00:00:00  https://www.facebook.com/SneakerShouts
                     http://www.kk-allen.com
2014-02-12T00:00:00  http://www.youtube.com/user/TheGabbieShow
2013-01-29T00:00:00  http://instagram.com/yoteensig
2013-12-09T00:00:00  https://soundcloud.com/travisscott-2/04-drugs-you-should-try-it
2014-04-06T00:00:00  https://www.facebook.com/PupsPorn
                     http://goosepancake.tumblr.com
                     http://fox59.com/
2009-01-19T00:00:00  http://www.abc.net.au/mediawatch/
2014-05-30T00:00:00  http://greatsss.com
                     http://ultimate-dickhead.tumblr.com/
2009-05-17T00:00:00  http://tweetsta.com/index.php
2014-04-19T00:00:00  http://itssongoku.tumblr.com
2012-10-01T00:00:00  https://twitter.com/zilliamson
2011-10-26T08:26:02  http://socialistworker.co.uk
2008-04-15T00:00:00  http://www.enzozelocchi.com
2012-04-27T00:00:00  http://www.youtube.com/watch?v=VAPIOciXs6A&feature=youtu.be
2014-04-02T00:00:00  http://www.rageon.com/collections/pokemon/?rageon.com
2008-01-02T00:00:00  http://www.youtube.com/user/FreeFallingHQ
                     http://www.youtube.com/watch?v=frNsJvV5zAs&feature=youtu.be
2014-02-06T00:00:00  http://cespedesfamilybarbecue.com/about/
2012-11-09T00:00:00  http://www.natethehitmaker.com/
2012-06-29T00:00:00  http://ImMackenson.com
2012-09-11T00:00:00  http://anhonestyear.com/
2012-04-06T00:00:00  http://www.romeolacoste.com
2014-09-13T00:00:00  http://smarturl.it/RFLT
                     http://proguitarshop.com/
2010-01-17T00:00:00  http://ohholybutt.tumblr.com
2001-02-01T00:00:00  http://Instagram.com/lawsofsex
2009-12-05T00:00:00  http://www.photografyum.com
                     http://www.hrtv.com
2009-10-03T00:00:00  http://sensualgif.tumblr.com/
                     http://kushoverboys.tumblr.com
2000-03-03T21:00:40  http://www.ew.com/ew/
                     https://31.media.tumblr.com/d0c4823d0bb16aa36fb99fba8b366140/tumblr_n7bi03tXo51t7eg
2011-04-22T12:53:05  http://english.alarabiya.net/
2014-09-24T06:39:31  https://twitter.com/nationofbiebs/status/457342650874474496
2004-10-07T17:42:11  http://www.womenshealthmag.com
```

memvsday.txt

```
http://texascruzin.blogspot.com/ 4 2448
http://www.youtube.com/swoozie 60 3149
http://dailyteenlife.com/ 2 150
http://www.desertstormradio.com 50 1351
http://duniasone.tumblr.com 2 825
http://search.espn.go.com/jemele-hill/ 83 2219
http://www.HeForShe.org 86 201
http://freebies4mom.com 185 2105
http://txsaywhat.com 13 997
http://www.hugedomains.com/domain_profile.cfm?d=juicegoddess&e=com 3 2154
http://www.theScore.com 53 4468
http://iamnamelessgem.tumblr.com 1 225
http://www.chargers.com 2664 5707
http://favstar.fm/users/mrtruthandsoul/recent 3 1879
http://zeenews.india.com/ 1135 385
http://www.ign.com 10561 5628
http://www.emc.com/index.htm 709 5791
http://Instagram.com/sammytellem 1 0
http://kenjisalk.tumblr.com/ 2 937
http://paramoreband.net 6 442
http://www.indonesia-zsolt72.blogspot.com 9 2255
https://www.facebook.com/MatthewKoma 4 756
http://www.hazymills.com 43 3678
https://twitter.com/louis_tomlinson/status/120620074301267968 2 1160
http://www.thenation.com/ 5164 0
http://www.amazon.co.uk/The-Good-Bad-Furry-Melancholy/dp/0751552399 5 835
http://www.cbsnews.com/evening-news/ 1751 858
http://writing.jan.io 31 1319
http://www.eyeconfla.com 95 2506
http://demilovato.com 328 3722
http://www.dnaindia.com 1254 3257
http://www.dodge.com/en/ 1209 2446
http://mastersgrandslam.com/en/home 4 848
http://batmanandbullwinkle.tumblr.com/ 1 0
http://www.complex.com 3991 5248
http://stlouis.cardinals.mlb.com/index.jsp?c_id=stl 861 2789
http://www.youtube.com/user/taylorcaniffofficial 8 1866
http://www.Fanatics.com 1561 5056
http://www.ingridmichaelson.com 462 4983
http://cuntsthesedays.tumblr.com 2 1152
http://www.EvaCassini.com 18 4559
http://www.girlythought.com/ 3 91
http://gyllenhaal.co.vu 4 2568
http://fy-winner.com 8 459
http://www.brainpickings.org/ 2163 2166
https://au.sports.yahoo.com/ 887 5029
http://www.alphaphiunk.com 2 117
http://www.katyperry.com 736 3711
http://abcfamily.go.com/ 1995 4618
http://www.eurogamer.net 2635 5226
```

q3-scatterplot

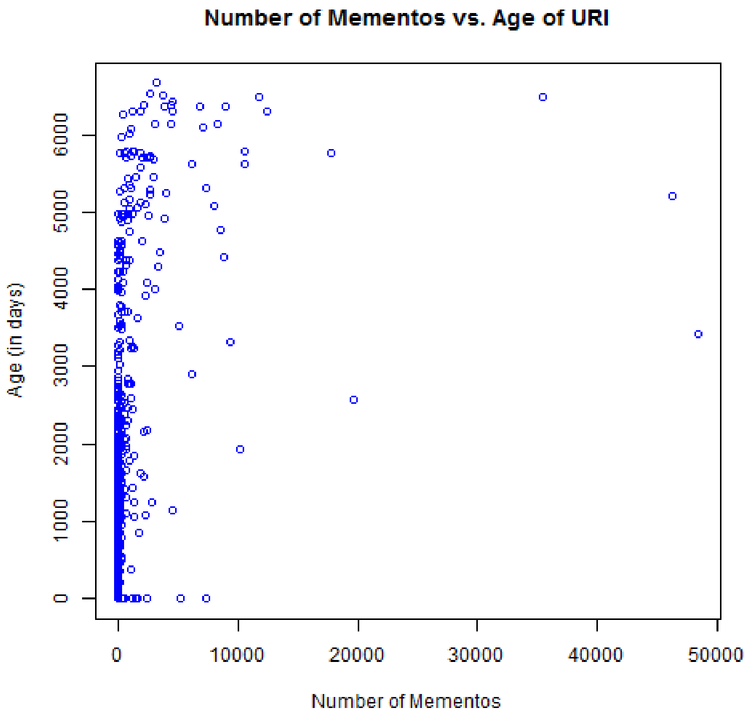

Figure 4: ScatterPlot

Bibliography

[1] Basic date and time types in python. https://docs.python.org/2/library/datetime.html.

[2] Entities in object. https://dev.twitter.com/overview/api/entities-in-twitter-objects#urls.

[3] Github for carbondate. https://github.com/HanySalahEldeen/CarbonDate.

[4] Producing simple graphs in r. http://www.harding.edu/fmccown/r/.

[5] Python twitter. https://code.google.com/p/python-twitter/.

[6] Python twitter. https://github.com/bear/python-twitter/blob/master/twitter_test.py.

[7] Twitter search for python. https://github.com/ckoepp/TwitterSearch.

[8] urllib2 dcomentation. https://docs.python.org/2/library/urllib2.html.

[9] Using twitter api keys. http://thomassileo.com/blog/2013/01/25/using-twitter-rest-api-v1-dot-1-with-python/.