CS 595: Assignment #2

Due on Thursday, September 25, 2014

Dr. Nelson 4:20pm

Holly Harkins

Contents

Problem 1

Write a Python program that extracts 1000 unique links from Twitter. Also note that you need to verify that the final target URI (i.e., the one that responds with a 200) is unique.

Listing 1: Python 1000 Links

```
# -*- encoding: utf-8 -*-
from __future__ import unicode_literals
import requests
from requests_oauthlib import OAuth1
from urlparse import parse_qs
import httplib2
from urllib import quote
from pprint import pprint
import twitter

REQUEST_TOKEN_URL = "https://api.twitter.com/oauth/request_token"
AUTHORIZE_URL = "https://api.twitter.com/oauth/authorize?oauth_token="
ACCESS_TOKEN_URL = "https://api.twitter.com/oauth/access_token"

##My Keys
CONSUMER_KEY = "Er2mm4vGncBo9nX49Esw"
CONSUMER_SECRET = "p2jmyo7OmfgeeeJiqtoQzowZftMucjnUyeMOMRZw82Y"

OAUTH_TOKEN = "251457467-tF4mgbwex37kkkbYzzjXGRgtQd29FhFMKgXa1mdM"
OAUTH_TOKEN_SECRET = "mcRcLNMdPtlSeldfbpWdnzupdXXh70VU3cNhmxxemDk"

def setup_oauth():

    ##Request Token
    oauth = OAuth1(CONSUMER_KEY, client_secret=CONSUMER_SECRET)
    r = requests.post(url=REQUEST_TOKEN_URL, auth=oauth)
    credentials = parse_qs(r.content)

    resource_owner_key = credentials.get(’oauth_token’)[0]
    resource_owner_secret = credentials.get(’oauth_token_secret’)[0]

    ##Authorize
    authorize_url = AUTHORIZE_URL + resource_owner_key
    print ’Please go here and authorize: ’ + authorize_url

    verifier = raw_input(’Please input the verifier: ’)
    oauth = OAuth1(CONSUMER_KEY,
                   client_secret=CONSUMER_SECRET,
                   resource_owner_key=resource_owner_key,
                   resource_owner_secret=resource_owner_secret,
                   verifier=verifier)
    ##Obtain the Access Token
    r = requests.post(url=ACCESS_TOKEN_URL, auth=oauth)
    credentials = parse_qs(r.content)
    token = credentials.get(’oauth_token’)[0]
    secret = credentials.get(’oauth_token_secret’)[0]
    return token, secret
```

Problem 1 continued on next page...

```
def get_oauth():
    oauth = OAuth1(CONSUMER_KEY,
                   client_secret=CONSUMER_SECRET,
                   resource_owner_key=OAUTH_TOKEN,
                   resource_owner_secret=OAUTH_TOKEN_SECRET)
    return oauth

def getContentLocation(link):
    h = httplib2.Http(".cache_httplib")
    h.follow_all_redirects = True
    resp = h.request(link, "GET")[0]
    contentLocation = resp[’content-location’]
    return contentLocation

if __name__ == "__main__":
    if not OAUTH_TOKEN:
      token, secret = setup_oauth()
      print "OAUTH_TOKEN: " + token
      print "OAUTH_TOKEN_SECRET: " + secret
      print
    else:

      ##Twitter Names to Follow
      TweetName = {’Enderle’,’thurrott’,’Jeremiah Owyang’,’LanceUlanoff’,’charleneli’,
          ’jsnell’,’Rafe’,’davezatz’,’Padmasree’,’harrymccracken’, ’timoreilly’,’
          leolaporte’,’inafried’,’abbielundberg’,’mattcutts’,’mattcutts’,’saschasegan’,
          ’comp_science’,’timberners_lee’,’geminodreal’,’SebastianThrun’,’BobMetcalfe’,
          ’lemire’,’fortnow’,’geomblog’,’DrQz’,’TheScienceGuy’,’carlzimmer’,’edyong209’
          ,’Jorge_Salazar’,’Bill_Romanos’,’QuantumDottie’,’Happy_Scientist’}

      ##Open File
      file1 = open(’FinalList.txt’,’w’)

      oauth = get_oauth()
      ##List for Final Links
      Final_List=[]

      for n in TweetName:
         r = requests.get(url="https://api.twitter.com/1.1/statuses/user_timeline.json
             ?screen_name="+ n +"&count="+"200", auth=oauth)
         p = r.json()
         for tweet in p:
            try:
                  u=tweet[’entities’][’urls’][0][’expanded_url’]

                  ##Finding the Final Link
                  e=getContentLocation(u)
                  print "Final link="
                  print e
                  ##Getting Response Code
                  r = requests.head(e)
                  print "response code="
                  print r.status_code
            ##Writing to file
            file1.write(e)
            file1.write('\n')
      except:
          pass

##Checking Number of Links
print "Number of Final Links="+ str(len (Final_List))

##Close File
file1.close()
```

Listing 2: Sample of 1000 Links

```
...
http://www.amazon.com/Founders-Less-Than-Three-ebook/dp/B00EIV13H6/ref=sr_1_1?ie=UTF8&qid=1377619910&sr=8-1&keywords=founders+less+than+three
https://www.facebook.com/photo.php?pid=287022984&l=e01ad468fb&id=75686763398
http://fertilityforecast.com
http://WordPress.com
http://ibmsmartcamp.com/2013/09/12/six-finalists-for-the-ibm-smartcamp-regional-finals-in-silicon-valley-announced/
http://www.kldenergy.com/
http://www.mitathletics.com/sports/m-footbl/2013-14/rostertm
http://www.beehiveid.com/
http://austin.3daystartup.org/apply/
https://plus.google.com/+PCMag/posts
http://www.pcmag.com/slideshow/story/310772/the-best-samsung-galaxy-s-4-cases
http://www.pcmag.com/article2/0,2817,2424438,00.asp
http://www.theawl.com/2013/09/techcrunch-journalists-or-startup-shills-you-decide
http://ctupowerofin.eventbrite.com/
http://www.briansolis.com/2013/08/the-disconnect-between-aging-management-and-the-younger-workforce/
https://cc.readytalk.com/cc/s/meetingArchive?eventId=2dkmxv13ci7f
https://www.surveymonkey.com/s/V3P7Q26
http://5by5.tv/afterdark/403
http://blog.lexfriedman.com/post/2856721100/ios-7
....
```

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/
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.

Listing 3: Python Memento

```
# -*- encoding: utf-8 -*-
import urllib2
import requests

file1=open("uniqLinks.txt","r")
file2=open("histogram.txt","w")
file2.write("URL"+","+" Memento"+"\n")
uri = "http://mementoproxy.cs.odu.edu/aggr/timemap/link/"

for line in file1:

    link=line.rstrip("\n")
    n = uri + link

    try:
        ##Opening New Link
        r = urllib2.Request(n)
        l = urllib2.urlopen(r)
        print "link=" + n

        ##Checking for 200 Response
        if l.code==200:
            timeMap = l.read()
            TOKENIZER_RE = re.compile('(<[^>]+>|[a-zA-Z]+="[^"]*"|[;,])\\s*')
            URI_DATETIME_RE = re.compile('/([12][90][0-9][0-9][01][0-9][0123][0-9]'
                                         '[012][0-9][0-5][0-9][0-5][0-9])/',
                                         re.IGNORECASE)
            URI_DATETIME_FORMAT = '%Y%m%d%H%M%S'

            ##Memento Count
            mc=0
            tokens=TOKENIZER_RE.findall(timeMap)

            ##Loop in Tokens
            for word in tokens:
                if word[:4] == "rel=":
                    rel=word[5:-1]

                    ## If Memento Found Add to Count
                    if "memento" in rel:
                        mc=mc+1
```

Problem 2 continued on next page...

```
                    elif "first memento" in rel:
                        mc=mc+1
                    elif "last memento" in rel:
                        mc=mc+1
                    elif "memento first" in rel:
                        mc=mc+1
                    elif "memento last" in rel:
                        mc=mc+1
                    elif "first last memento" in rel:
                        mc=mc+1

            ##Write to File
            file2.write(link + "," + str(mc)+"\n")

        mc=0
        ##Write Zero Mementos to File
        file2.write(link + "," + str(mc)+"\n")
        continue

file1.close()
file2.close()
```

Listing 4: Sample of Memento Links

```
URL, Memento
...
http://allthingsd.com/20130912/samsung-of-course-our-next-smartphones-will-be-64-bit/?
    mod=tweet,2
http://blog.computationalcomplexity.org/2013/04/computer-assisted-proofs-still.html,0
http://instagram.com/p/eN140wmxK4/,0
http://keelingcurve.ucsd.edu/,1
http://mashable.com/2013/09/09/google-embedded-posts/,3
http://news.discovery.com/space/100yss-former-president-bill-clinton-backs-100-year-
    starship-120905.htm,1
http://omnomnomify.com,2
http://oxforddictionaries.com/words/what-do-you-call-a-group-of,29
http://radar.oreilly.com/2011/07/google-plus-social-backbone.html,43
http://stores.ebay.com/Auction-Cause-Charity-Auctions/B612-Foundation.html,0
...
```

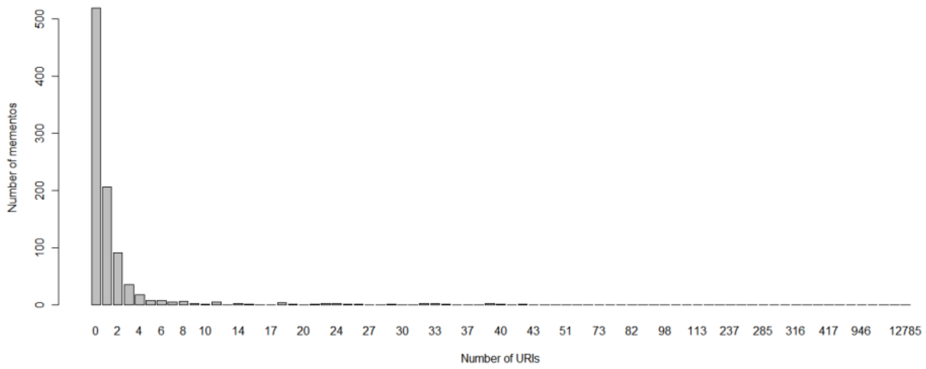

Figure 1: Number of URIs vs Number of Mementos

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

Listing 5: Python Carbon Date

```
# -*- encoding: utf-8 -*-
import urllib2
import requests
from server import *
import time
from datetime import date
from datetime import datetime

file1=open("histogram.txt","r")
##File for Carbon Date Results
file2=open("CarbonDate.txt","w")

##File for New Extracted Results
file3=open("histogram2.txt","w")

##Writing Results of First Line
file3.write("URL"+","+" Memento"+","+" Estimated Creation Date"+"\n")

lineCount=0
today=date.today()

for line in file1:
    lineCount=lineCount+1

    if lineCount!=1:

      Oneline=line.rstrip("\n")
      splitLine={}
      splitLine=Oneline.split(",")
      link=splitLine[0]
      memento=splitLine[1]

      ##Send Link to Carbon Tool
      CDate=carbonDate(link)

      file2=open("CarbonDate.txt","w")

      #Write to File
      file2.write("link= " + link + "\n" + CDate + "\n")
      file2.close()

      file2=open("CarbonDate.txt","r")

      #Read Data
      for line in file2:
          l=line.rstrip("\n")
          if l.find("Estimated Creation Date") != -1:
            if l[37:38] != "":
              ##Create Date Format
              d1=l[38:40]
              m1=l[35:37]
              y1=l[30:34]
              edate=date(int(y1), int(m1), int(d1))

              ##Calculate Age
              Age=abs(today-edate)

              ##Write to File
              file3=open("histogram2.txt","w")
              file3.write(link + "," + memento + "," + str(Age.days) + "\n")
              file3.flush()

file1.close()
file2.close()
file3.close()
```

Listing 6: Sample of Carbon Date Links

```
URL, Memento, Estimated Creation Age
http://BillMoyers.com,285,4957
http://abcnews.go.com/Entertainment/t/story/van-gogh-museum-van-gogh-identified
    -20197249,0,15
http://biocareers.com/bio-careers-blog/business-academic-science,4,606
http://code.google.com/p/msinttypes/,27,2398
http://codekeyboards.com/,11,29
http://en.wikipedia.org/wiki/Continuum_%28TV_series%29,0,495
http://instagram.com/p/aHT9afsMXF/,0,114
http://lilly.tumblr.com/post/43088488614/a-few-folks-have-asked-me-what-i-think-of-the
    -news,3,223
http://mashable.com/2013/09/09/google-embedded-posts/,3,16
http://pioneers.io/blog/hardware-is-the-new-software-chris-anderson,1,64
http://radar.oreilly.com/2011/07/google-plus-social-backbone.html,43,799
```

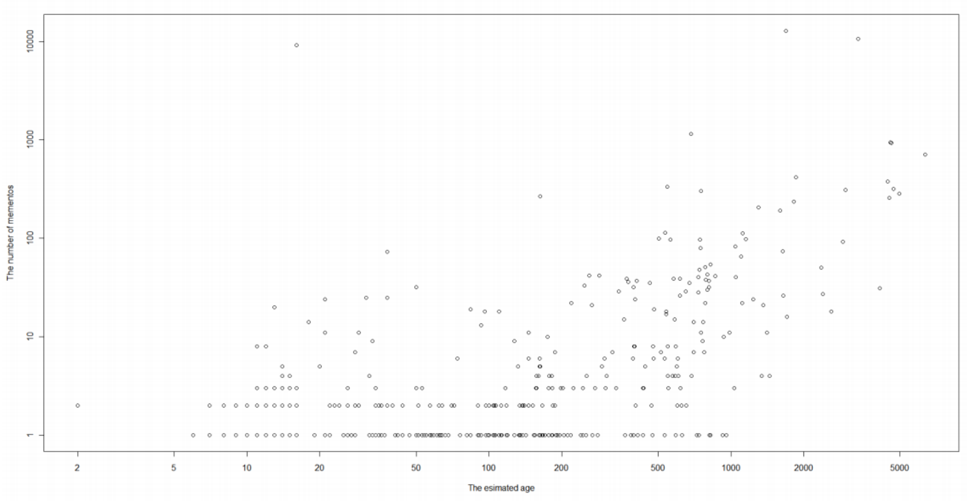

Figure 2: Number of Memento vs The Estimate Age

References

Using Twitter REST API with Python
Author: Thomas Sileo
http://thomassileo.com/blog/2013/01/25/using-twitter-rest-api-v1-dot-1-with-python/

Obtain Redirected Links to Ultimate Ones
http://stackoverflow.com/questions/6158895/httplib-is-not-getting-all-the-redirect-codes/1161781711617817/

Twitter API- GET statuses user timeline
https://dev.twitter.com/docs/api/1.1/get/statuses/usertimeline/

Python - Get HTTP response code from a URL
http://stackoverflow.com/questions/1140661/python-get-http-response-code-from-a-url/

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

Carbon Dating
http://ws-dl.blogspot.com/2013/04/2013-04-19-carbon-dating-web.html