CS532 Web Science: Assignment 2

Finished on February 11, 2016

Dr. Michael L. Nelson

Naina Sai Tipparti
ntippart@cs.odu.edu

Contents

Listings

Problem 1

Question

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/

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.).

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

Using the python module requests made this task a breeze as well as the initial code provided by Thomas Sileo's blog post.

```
# -*- encoding: utf-8 -*-
import requests
from requests_oauthlib import OAuth1
from urllib import quote

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"

CONSUMER_KEY = "c7VFfCTtUqFDg69MRHvGnpSwt"
CONSUMER_SECRET = "4r8yeBQmziO8HuXY3UQN3qGigz0hgrbQxXpD4w2nR7fBRIRLqU"

OAUTH_TOKEN = "798668178-bH8DbMpNuWkfhAHxuODgWSHwQE65B1WZnc4Ahtej"
OAUTH_TOKEN_SECRET = "FhykPKnQcgKQBE43os2bDZ31ugH9RVSG3HYoOL7QG7RNC"

SEARCH_URI = "https://api.twitter.com/1.1/search/tweets.json?q="

SEARCH_ITEMS = map(quote, [ 'allu arjun',
                            'vijayawada',
                            'Cristiano Ronaldo',
                            'Adolf Hitler',
                            'Emma Watson',
                            'Anushka Shetty',
                            'Rajini Kanth',
                            'Lionel Messi',
                            'Ricardo Kaka',
                            'Wayne Rooney',
                            'Gareth Bale',
                            'Neymar',
```

```
                    'Malaika Arora Khan',
31                  'Isco',
                    'Adam Gilchrist',
                    'Kevin de bruyne',
                    'Kevin Pietersen',
                    'Julia Roberts',
36                  'Scarlett Johansson',
                    'Kate Winslate',
                    'Kamal Hassan',
                    'Kumar sangakkara'
                    'Mahela jayawardene'
41                  'Roger Federer'
                    'Maria Sharapova'
                    'Serena Williams'
                    'Venus Williams'
                    'Justin Henin'
46                  'Justin Langer'
                    'Sania Mirza'
                    'Shoaib Malik'
                    'Shoaib Aktar'
                    'Sourav Ganguly'
51                  'Rafeal Nadal'
                    'Real Madrid CF'
                    'Rahul Dravid'
                    'Manchester United'
                    'FC Barcelona '
56                  'Manchester City'
                    'Chelsea FC'
                    'Liverpool FC'
                    'Aston Villa FC'
                    'Aston Martin'
61                  'lamborghini'
                    'FC Bayern Munich'
                    'Borussia Dortmund'
                    'Philipp Lahm'
                    'Sergio Ramos'
66                  'Sergio Aguero'
                    'Marcelo Vieira'
                    'Luis Suarez'
                    'Luis Enrique'
                    'Jose Mourinho'
71                  'Alex Ferguson'
                    'Pep Guardiola'
                    'Shane Watson'
                    'Shane Warne'
                    'Bill Gates'
76                  'Rothschild'
                    'Mark Zuckerberg'
                    'Chetan Bhagat'
                    'Ishant Sharma'
                    'Virat Kohli'
81                  'Sachin Tendulkar'
                    'Ricky Ponting'
                    'Matthew Hayden'
                    'Rohit Sharma'
                    'Irfan Pathan'
86                  'yusuf Pathan'
                    'Katrina Kaif'
                    'Anushka Sharma'
                    'Salman Khan'
                    'Ranbir Kapoor'
91                  'Puneeth Rajkumar'
                    'Rajkumar'
                    'Ganesh'
                    'Upendra'
                    'Darshan'
96                  'Shivrajkumar'
                    'Rakhul Preeth Singh'
```

```
                            'Soundarya'
                            'Savitramma'
                            'Anusuya'
101                         'Roja'
                            'Ram Gopal Varma'
                            'Pawn Kalyan'
                            'Ram Charan'
                            'Allu Arvind'
106                         'Allu Ramalinga'
                            'Kotta Srinivas Rao'
                            'Surya'
                            'Joythika'
                            'Nagarjun'
111                         'Akhil Akkeneni'
                            'Naga Chaitanya Akkeneni'
                            'Amala Akkeneni'
                            'Barack Obama'
                            'Abdul Kalam'
116                         'Subhash Chandra Bose'
                            'Shreya Ghoshal'
                            'Shreya Sharan'
                            'Sidharth'
                            'Mahesh Babu'
121                         'Jenelia '
                            'Salam Khan'
                            'Sharukh Khan'
                            'Hrithik Roshan'
                            'Deepika Padukone'
126                         'Pooja Ghandhi'
                            'Ravichandranan'
                            'Arjun Sarja'
                            'Shankar Nag'
                            'Prakash Raj'
131                         'Tennis Krishna'
                            'Nagesh'
                            'Lokanath'
                            'Meena'
                            'Arundathi Nag'
136                         'Nagma'
                            'Laila Mehdin'
                            'Sindhu Menon'
                            'Jaylalithaa'
                            'Vishnuvardhan'
141                         'Suman Nagarkar'])

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

def find_uris(uris):
    with open('output', 'a') as outfile:
151     for search_item in SEARCH_ITEMS:
            result = requests.get(SEARCH_URI + search_item + '&filter%3Alinks&count=10000',
                auth=oauth)
            for status in result.json()['statuses']:
                for url in status['entities']['urls']:
                    if len(uris) == 10000:
156                     return
                    if 'expanded_url' in url:
                        try:
                            result = requests.get(url['expanded_url'], timeout=4)
                            # only add expanded uris if they aren't in the list already
161                         if result.status_code == 200 and result.url not in uris:
                                add_uri(uris, result.url)
                                outfile.write('%s\n' % result.url)
                        except Exception as e:
```

```
                            print e
166                         continue

def add_uri(uris, uri):
    uris.add(uri)
    print 'added uri #%d: %s' % (len(uris), uri)
171
if __name__ == "__main__":
    oauth = get_oauth()
    uris = set()
    # read in previous set of uris
176 try:
        with open('output', 'r') as infile:
            for line in infile.readlines():
                add_uri(uris, line.strip())
    except IOError:
181     pass
    find_uris(uris)
```

Listing 1: urifinder.py

The script was run multiple times to get the desired 1000 unique URIs. It would end prematurely at times, so the data set was initialized with the data of the previous run and then passed on to the `find_uris` function to preserve work performed.

```
https://www.youtube.com/watch?v=fEhXgOKpE-o&feature=youtu.be&a
https://twitter.com/FutNaRedonda/status/696095799013478402
3 https://twitter.com/dr_javi/status/693995331164463104/photo/1
https://www.youtube.com/watch?v=Ig-kBwOdL_g&feature=youtu.be&a
https://www.youtube.com/watch?v=yypauzQV8d0&feature=youtu.be&a
https://twitter.com/asksparati/status/696367298525978624
https://twitter.com/History_Futbol/status/694720215087652864/photo/1
8 http://www.dainikbhaskar.info/sports/kids-should-learn-from-soaring-leicester-not-superstars
    -lionel-messi-and-cristiano-ronaldo-they-show-the-value-of-hard-graft/
https://twitter.com/Juezcentral/status/696515201743659008
http://www.oldpicsarchive.com/selected-photos-part-4-33-rare-pics/4/?utm_content=buffer840ce
    &utm_medium=social&utm_source=twitter.com&utm_campaign=buffer
http://www.georgewalkerbush.net/bushfamilyfundedhitler.htm
http://odia.ig.com.br/noticia/rio-de-janeiro/2016-02-03/justica-proibe-venda-e-divulgacao-de
    -livro-escrito-por-adolf-hitler.html
13 http://www.telegraph.co.uk/news/worldnews/donald-trump/12038640/Who-said-it-Donald-Trump-or-
    Adolf-Hitler.html
https://www.youtube.com/watch?v=d3r70E6Dvfs&feature=youtu.be&a
https://www.youtube.com/watch?v=sI1E4Vs7cbk&feature=youtu.be&a
http://www.newsweek.com/adolf-hitler-black-holocaust-dark-secrets-423735?rx=us
http://www.flimper.com/events/4
18 https://www.facebook.com/robcaiafa/posts/10208639359731659
http://linkis.com/www.youtube.com/lrpOF
https://www.youtube.com/watch?v=QnpBN-ltVtE&feature=youtu.be
```

Listing 2: Sample of 1000 Links

Problem 2

Question

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/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.

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

Answer

The python script in Listing 3 was used to retrieve the timemaps and then parse the returned html, traveling down the rabbit hole if the target URI has more than 1000 mementos.

```
# -*- encoding: utf-8 -*-
#! /usr/bin/python

import requests
import re

MW_URI = "http://mementoproxy.cs.odu.edu/aggr/timemap/link/1/"

if __name__ == '__main__':
    with open('output', 'r') as f:
        output = open('site_mementos', 'w')
        mementos = {}
        for uri in f.read().split('\n'):
            if uri is '':
                continue
            count = 0
            target_uri = MW_URI + uri
            while True:
                result = requests.get(target_uri)
                if result.ok:
                    count = count + result.text.count('rel="memento"')
                last_line = result.text.split('\n')[-1]
                if 'rel="timemap"' not in last_line:
                    break
                sites = re.findall(r'<([^<|^>]+)>', last_line)
                target_uri = sites[1]
            mementos[uri] = count
            print 'found %d mementos for uri: %s' % (count, uri)
            output.write('%s %d\n' % (uri, count))
    output.close()
```

Listing 3: mementofinder.py

The dataset created Listing 3. A log scale was used along the y-axis to show more detail among the results. The script in Listing 4 was used to create the histogram in Figure 1, which shows the distribution of mementos per site from the dataset of Question 1.

```
#! /usr/bin/Rscript

data <- read.table("D:/cs532/q2/results", header=TRUE, comment.char="")
counts <- table(data$Mementos)
pdf("histogram.pdf")
barplot(counts, log="y", ylim=c(.75, nrow(data)), ylab="Sites", xlab="Memento Count", main="
    Memento Count per Site")
dev.off()
```

Listing 4: build_histogram.r

```
https://twitter.com/asksparati/status/696367298525978624 0
https://twitter.com/History_Futbol/status/694720215087652864/photo/1 0
http://www.dainikbhaskar.info/sports/kids-should-learn-from-soaring-leicester-not-superstars
    -lionel-messi-and-cristiano-ronaldo-they-show-the-value-of-hard-graft/ 0
https://twitter.com/Juezcentral/status/696515201743659008 0
http://www.oldpicsarchive.com/selected-photos-part-4-33-rare-pics/4/?utm_content=buffer840ce
    &utm_medium=social&utm_source=twitter.com&utm_campaign=buffer 0
http://www.georgewalkerbush.net/bushfamilyfundedhitler.htm 79
http://odia.ig.com.br/noticia/rio-de-janeiro/2016-02-03/justica-proibe-venda-e-divulgacao-de
    -livro-escrito-por-adolf-hitler.html 0
http://www.telegraph.co.uk/news/worldnews/donald-trump/12038640/Who-said-it-Donald-Trump-or-
    Adolf-Hitler.html 7
https://www.youtube.com/watch?v=d3r70E6Dvfs&feature=youtu.be&a 16
https://www.youtube.com/watch?v=sI1E4Vs7cbk&feature=youtu.be&a 16
http://www.newsweek.com/adolf-hitler-black-holocaust-dark-secrets-423735?rx=us 2
http://www.flimper.com/events/4 0
https://www.facebook.com/robcaiafa/posts/10208639359731659 0
http://linkis.com/www.youtube.com/lrpOF 0
https://www.youtube.com/watch?v=QnpBN-ltVtE&feature=youtu.be 16
http://daveschlueteronline.com/the-penguin-updates-and-seo/ 3
https://twitter.com/MrJohnQZombie/status/664324668149493760/photo/1 0
https://twitter.com/MrJohnQZombie/status/664324668149493760 0
http://www.nzherald.co.nz/entertainment/news/article.cfm?c_id=1501119&objectid=11586082&ref=
    rss&utm_source=dlvr.it&utm_medium=twitter 0
http://frtyb.com/go/Og1_bnwHp_j4D2/DEFAULT 0
https://www.youtube.com/watch?v=cZKeqenZbJk&feature=youtu.be&a 16
```

Listing 5: Sample of Memento Links

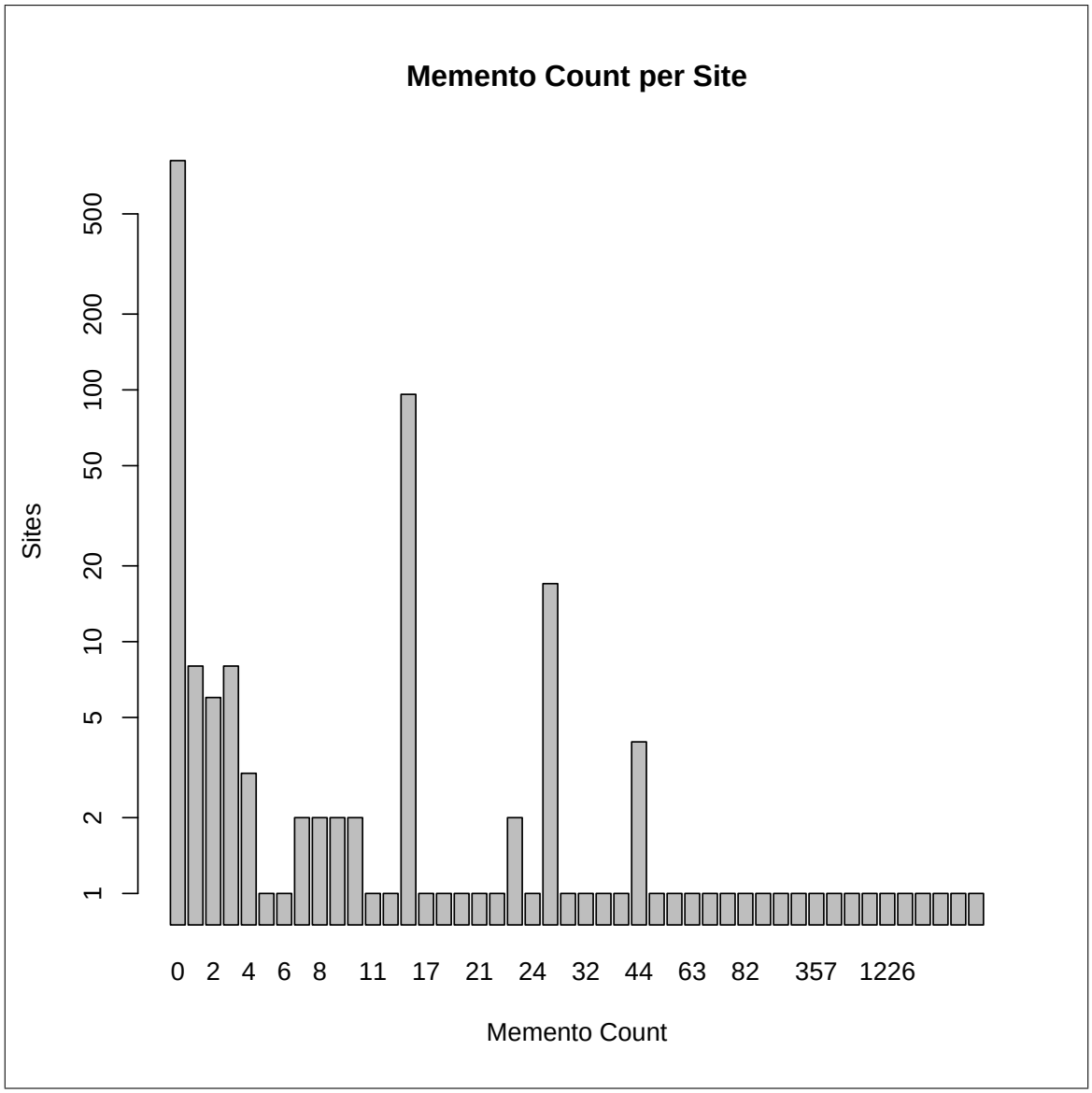

Figure 1: Histogram of Site Mementos

References