Assignment 2

Fall 2014
CS595 Web Science
Dr. Michael Nelson

Mathew Chaney

September 18, 2014

Contents

1 Question 1

1.1 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 different shortened URIs for www.cnn.com. For example,

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

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.

1.2 Resources

- Getting Started with Twitter API: http://thomassileo.com/blog/2013/01/25/using-twitter-rest-api-v1-dot-1-with-python/
- Twitter Search API: https://dev.twitter.com/rest/public/search
- Twitter API - Get / Search Tweets: https://dev.twitter.com/rest/reference/get/search/tweets

1.3 Answer

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

```
1  # -*- encoding: utf-8 -*-
2  import requests
3  from requests_oauthlib import OAuth1
4  from urllib import quote
5
6  REQUEST_TOKEN_URL = "https://api.twitter.com/oauth/request_token"
7  AUTHORIZE_URL = "https://api.twitter.com/oauth/authorize?oauth_token="
8  ACCESS_TOKEN_URL = "https://api.twitter.com/oauth/access_token"
9
10 CONSUMER_KEY = "PDBekXkvUto4V0XYZrrizcEub"
11 CONSUMER_SECRET = "OElKNfpNWF8Eh4iwbDFYFKDMBSouni3uRZrpsoGhbJcLZZmnBq"
12
13 OAUTH_TOKEN = "2560074793-g7ESlsQmwl3YKAfCJnIBa0lh3wHLjmPqj96XFuV"
14 OAUTH_TOKEN_SECRET = "tGYCQa9LL2i6wmApJzbGzHdVIVA65xiwVffPmbqWJwZPs"
15
16 SEARCH_URI = "https://api.twitter.com/1.1/search/tweets.json?q="
17
18 SEARCH_ITEMS = map(quote, [ 'space x',
19                             'elon musk',
```

```
                            'richard garriott',
                            'starcraft 2',
                            'ebola virus',
                            'world cup',
                            'singularity',
                            'rick and morty',
                            'iphone 6',
                            'android',
                            'robin williams',
                            'tony stewart',
                            'bitcoin',
                            'game of thrones',
                            'facebook',
                            'youtube',
                            'google',
                            'chris roberts',
                            'hyper light drifter',
                            'golang'])

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

def find_uris(uris):
    for search_item in SEARCH_ITEMS:
        result = requests.get(SEARCH_URI + search_item + '&filter%3Alinks&count=1000', auth=
            oauth)
        for status in result.json()['statuses']:
            for url in status['entities']['urls']:
                if len(uris) == 1000:
                    return
                if 'expanded_url' in url:
                    try:
                        result = requests.get(url['expanded_url'])
                        # only add expanded uris if they aren't in the list already
                        if result.status_code == 200 and result.url not in uris:
                            add_uri(uris, result.url)
                    except Exception as e:
                        print e
                        continue

def add_uri(uris, uri):
    uris.add(uri)
    print 'added uri #%d: %s' % (len(uris), uri)

if __name__ == "__main__":
    oauth = get_oauth()
    uris = set()
    # read in previous set of uris
    with open('output', 'r') as infile:
        for line in infile.readlines():
            add_uri(uris, line.strip())
    find_uris(uris)
    with open('output', 'w') as outfile:
        outfile.writelines('%s\n' % uri for uri in uris)
```

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.