INTRO. TO WEB SCIENCE: CS 532: A6

Due on Thursday, March 23, 2017

Dr. Nelson

Udochukwu Nweke

Contents

Problem 1 **3**

Problem 2 **6**

Problem 1

Listing 1: Retrieve Dr. Nelson’s Twitter Followers and Calculate their Friendship Code

```
from util import *
import tweepy
import json
import time

consumer_key = ’01oEm0j9Y52TFtjA5wgRHt9z6’
consumer_secret = ’MWPYnHvrezvhuuFQkOQgnBMYwXZD6SJpnoI8QlE7ZajgQMUKKs’
access_token = ’154076252-uK6XnhweIkuc0qIvsNmGiiRebLqvYHbtWDgA5PBi’
access_token_secret = ’LDA5Qel3UQtIwUhvAZLffCGZ9pmmc7wkFOL5k0xx5Yt9O’

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(auth)

def writeFollowers(response, friendsDict):

    for follower in response:
        friendsDict[follower.screen_name] = follower.name

def getLink(source_screen_name, source_name, target_screen_name, target_name, outfile):
    link = api.show_friendship(source_screen_name=source_screen_name,
    target_screen_name=target_screen_name)

    if( link[0].following ):
        print(’\t\t’, source_name, ’follows’, target_name)
        #graphEdges.append( {’source’: source_name, ’target’: target_name} )
        outfile.write(’{"source": "’ + source_name + ’",
        "target": "’ + target_name + ’"},\n’)

    if( link[0].followed_by ):
        print(’\t\t’, target_name, ’follows’, source_name)

        #graphEdges.append( {’source’: target_name, ’target’: source_name} )
        outfile.write(’{"source": "’ + target_name + ’",
        "target": "’ + source_name + ’"},\n’)

def writeFirstDegreeLinks(outfile):

    followersList = readTextFromFile(’./phonedude_mln.followers.array.json’)
    followersList = json.loads(followersList)

    for follower in followersList:
        outfile.write(’{"source": "Michael L. Nelson", "target":
         "’ + follower[’name’] + ’"},\n’)


def computeFriendshipLink(outfile):

    followersList = readTextFromFile(’./phonedude_mln.followers.array.json’)
```

Problem 1 continued on next page. . .

```
    followersList = json.loads(followersList)

    counter = 0
    for i in range(9, 100):
        for j in range(i+1, len(followersList)):
            source_screen_name = followersList[i]['screen-name']
            target_screen_name = followersList[j]['screen-name']

            source_name = followersList[i]['name']
            target_name = followersList[j]['name']

            print('\t', source_screen_name, source_name )
            print('\t', target_screen_name, target_name )

            getLink(source_screen_name=source_screen_name, source_name=source_name,
            target_screen_name=target_screen_name,
            target_name=target_name, outfile=outfile)


            print('\t', i, j)
            print('\tsleeping')
            time.sleep(6)

            counter += 1
        print('\n'*4)

    print('counter:', counter)


def getFriendsDict(twitterName):

    try:
        rateLimitQuota = api.rate_limit_status()['resources']['followers']
        ['/followers/list']['remaining']
        print('\trateLimitQuota:', rateLimitQuota)


        twitterAccount = api.get_user(twitterName)
        #followersCount = int(twitterAccount._json['friends_count'])

        followersDict = {}
        response = twitterAccount.followers(count=200, cursor=-1)
        nextCursor = response[1][1]
        writeFollowers(response[0], followersDict)

        for i in range(0, 3):
            response = twitterAccount.followers(count=200, cursor=nextCursor)
            nextCursor = response[1][1]
            writeFollowers(response[0], followersDict)


        writeDictToJson('./phonedude_mln.followers.json', followersDict)
    except:
        errorMsg()

#getFriendsDict('phonedude_mln')


outfile = open('./friendshipGraph.json', 'a')

#outfile.write('[\n')
#writeFirstDegreeLinks(outfile)

computeFriendshipLink(outfile)


outfile.write(']')
outfile.close()
```

Use D3 to visualize your Twitter followers. Use my twitter account ("@phonedude_mln") if you do not have >= 50 followers. For example, @hvdsomp follows me, as does @mart1nkle1n. They also follow each other, so they would both have links to me and links to each other.

To see if two users follow each other, see:

`https://dev.twitter.com/rest/reference/get/friendships/show`

Attractiveness of the graph counts! Nodes should be labeled (avatar images are even better), and edge types (follows, following) should be marked.

Note: for getting GitHub to serve HTML (and other media types), see:

`http://stackoverflow.com/questions/6551446/can-i-run-html-files-directly-from-github-inste`
Be sure to include the URI(s) for your D3 graph in your report.

Solution 1:

1. In order to visualize Dr. Nelson's twitter followers, I used *getFriendsDict('phonedude_mln')* in Listing 1 to extract Dr. Nelson's 633 followers and saved the result into *phonedude_mln.followers.json* file.

2. I used *computeFriendshipLink()* in listing 1 to compute Dr. Nelson's first 100 followers friendship link. The result of Dr. Nelson's Frienship link is saved in *phonedude_mln.followers.array.json* .

3. I considered the first 100 followers from *phonedude_mln.followers.array.json* in generating Dr. Nelson's frienship graph and saved the result into *followers.json*

Finally, I used the D3 to create a graph for Dr. Nelson's twitter followers and his friend's followers in (*followers.json*) and hosted the graph on github. The graph can be previewd with the link below:

`http://htmlpreview.github.io/?https://github.com/UdochukwuNweke/cs532-s17/blob/master/A6/followers.html`

Problem 2

3. Using D3, create a graph of the Karate club before and after the split.
Weight the edges with the data from: `http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/zachary.dat`
Have the transition from before/after the split occur on a mouse click. This is a toggle, so the graph will go back and forth beween connected and disconnected.

Solution 2:

In order to create a graph for karate club before and after split with D3, I generated two graphs. The first graph *KarateClub.before.json* has all edges connecting Mr. Hi and John A subgraphs. The second graph *KarateClub.after.json* removes the edges connecting Mr. Hi and John A subgraphs.

The graph at the click of a node toggles back and forth between *KarateClub.before.json* (connected) and *KarateClub.after.json* (disconnected)nodes.
`http://htmlpreview.github.io/?https://github.com/UdochukwuNweke/cs532-s17/blob/master/A6/karateClub.html`

References

[1] Data-Driven Document. https://d3js.org/. Accessed: 2017-22-03.

[2] Karate Club. http://bl.ocks.org/majetisiri/316e3a1537b469154779. Accessed: 2017-21-03.

[3] Tweepy. http://www.tweepy.org/. Accessed: 2017-22-03.