

Assignment 2

Tarek Fouda

2016-02-10

1 Introduction

This report explains how I managed to solve Assignment 2 in Web Science class which is due 02/11/2016. It is mainly consisted of three Questions. Will show my approaches and implementation in each of the three Questions.

2 Problem 1- Extracting 1000 unique URLs from Twitter newsfeed

2.1 My approach

Baiscally I had to create a twitter account which I already have, also I had to register for a new twitter application to get the Consumer key, Consumer secret key, Key token and secret key token.

These variables allow you to use a Twitter python API, by just inserting these variables in your code. You would be able to, for example, Post a status from python code. But in this part of the assignment we are required to 1000 unique links from the newsfeed or even from someone's profile on Twitter.

I downloaded python-Twitter API from Github <https://github.com/ideoforms/python-twitter-examples>. It has a lot of python files which do specific tasks, for example twitter-list-retweets.py which lists all the retweets received on a specific tweet. What we only care about is extracting links. Creating a twitter application was the first thing I should do, following the steps on <https://github.com/ideoforms/python-twitter-examples> made me capable of creating a twitter app and now I can go to the config.py file and Type in my accesscode and secret keys Tokens as follows:

```
1 consumer_key = "KEtYeXDYJwgdX0IHuwUf1Hsw"  
2 consumer_secret = "  
    xkErPgMDfParcniEkbhRLgf8T6EXGOfhNP1bAbKSbog5rqCKxi"  
3 access_key = "308651543-bWKmgqe2AP3xTx85jyHPBUrovjdMtNej2SOqOjZd"  
4 access_secret = "PgKsQJxjvqaocAZmNG2D5t2Q7ZkAcDoPTHvLfOEe2ghj9"
```

Listing 1: Config.py

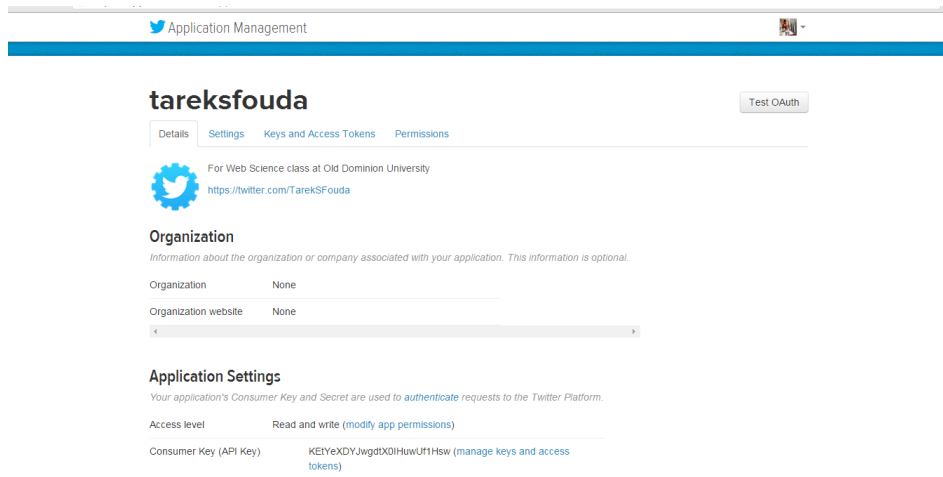


Figure 1: Twitter app

The actual code that should be implemented lies in twitter-stream-extract-links.py.

```

1 from twitter import *
2 # Tarek Fouda
3 # This program prints 1000 URLs in my twitter news feed
4 # using twitter API package downloaded from GITHUB.
5 config = {}
6 execfile("config.py", config)
7 auth = OAuth(config["access_key"], config["access_secret"], config[
8     "consumer_key"], config["consumer_secret"])
9 stream = TwitterStream(auth = auth, secure = True)
10 tweet_iter = stream.statuses.filter(track = "social")
11 listCount = 1
12 textCount = 1
13 l = []
14 for tweet in tweet_iter:
15     if (textCount <= 1000):
16         for url in tweet["entities"]["urls"]:
17             try:
18                 l.insert(listCount, url["
19                     expanded_url"])
20                 if (l.count(url["expanded_url"]) <=
21                     1):
22                     print "Found URL: %s" % url
23                     ["expanded_url"]
24                     f = open('urls.txt', 'a')
25                     f.write(url["expanded_url"]
26                         + '\n')
27                     textCount = textCount + 1
28                     f.close()
29             else:
30                 pass
31         except:

```

```
C:\Windows\system32\cmd.exe - python twitter-stream-extract-links.py
Found URL: http://www.wfmz.com/37945270?utm_medium=social&utm_source=twitter_69n
ews_-_Twitter
Found URL: http://allchristiannews.com/apple-facebook-nike-go-to-court-to-fight-f
or-gay-marriage/?utm_source=ReviveOldPost&utm_medium=social&utm_campaign=ReviveO
ldPost
Found URL: http://www.megaupload.us/download-video-bokep-indonesia/chubby/?utm_s
ource=ReviveOldPost&utm_medium=social&utm_campaign=ReviveOldPost
Found URL: http://allchristiannews.com/new-ager-radio-talk-show-host-used-to-han
g-up-on-christians-now-a-believer-who-shares-the-gospel/?utm_source=ReviveOldPos
t&utm_medium=social&utm_campaign=ReviveOldPost
Found URL: https://es-us.noticias.yahoo.com/video/la-respuesta-la-dj-m%C3%A1s-18
3012803.html?soc_src=social-sh&soc_trk=tw
Found URL: http://mynaturalreality.com/?p=6160&utm_source=ReviveOldPost&utm_medi
um=social&utm_campaign=ReviveOldPost
Found URL: http://answeringlegal.com/2016/02/10/things-to-avoid-while-blogging-a
nd-posting-on-social-media/
Found URL: http://gshow.globo.com/realities/bbb/BBB-16/noticia/2016/02/01/ha-ela-
ana-paula-incendeia-internet-com-seu-retorno-ao-jogo.html?utm_source=twitter&utm
_medium=social&utm_content=bbb&utm_campaign=bbb
Found URL: http://www.albanyherald.com/news/local/public_safety/albany-dougherty
-police-fire-ems-reports---feb/article_6c4804f5-f539-5f92-8da1-3742c9d680a7.html
?utm_medium=social&utm_source=twitter&utm_campaign=user-share
Found URL: http://www.theguardian.com/commentisfree/2016/feb/11/300-million-tory
-councils-favouritism-cuts-government-adult-social-care
Found URL: http://social.os.uk/kiwsh
Found URL: http://buddy18.com
Found URL: http://ow.ly/Ye11v
Found URL: http://bit.ly/SocialMediaElection
Found URL: http://dw.com/p/1Hu4m
Found URL: http://paper.li/laurentbrouat?edition_id=7dc34be0-d0e4-11e5-a5cc-0cc4
7a0d1609
Found URL: http://gamerwife.com/2015/10/12/teeblox-october-2015-unboxing-giveawa
y/?utm_source=ReviveOldPost&utm_medium=social&utm_campaign=ReviveOldPost
Found URL: http://xn----7sbihdquefksjr1d.xn--p1ai/%D0%BE%D0%B1%D1%81%D0%BB%D0%B
5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B9
/?utm_source=social&utm_medium=twitter&utm_campaign=1003197
Found URL: http://www.globalresearch.ca/decoding-the-us-empire-of-chaos-the-glob
al-reversal-of-the-social-evolution-of-humanity/5507118
Found URL: http://ift.tt/1WgAnpU
Found URL: http://oplace.ru/stati/avto/poyetapnaja-pokraska-avtomobilja.html?utm
_source=social&utm_medium=twitter&utm_campaign=1004726
Found URL: http://www.volkskrant.nl/wetenschap/overgebleven-neanderthaler-dna-ve
rgroot-kans-op-depressie~a4242976/?utm_source=twitter&utm_medium=social&utm_cont
ent=free&utm_campaign=shared%20content&hash=b38d67859680c3c759677a12fc886411c0b2
4f03
Found URL: https://disabledperson.wordpress.com/2016/02/11/social-security-disab
ility-benefits-tr/
Found URL: http://neuvoo.co.uk/job.php?id=80dz456hi3&source=twitter&lang=en&clie
nt_id=512&l=%28Unspecified+City%29%2C+England%2C+GB&k=Senior+Account+Manager+%2F
+Junior+Account+Director+%28Social+Media%29
Found URL: http://ow.ly/Ye17h
Found URL: http://socialautopilot.co
Found URL: https://twitter.com/felixsalmon/status/697843874841100292
Found URL: http://maisfutebol.iol.pt/internacional/alemanha/st-pauli-com-camisol
a-especial-nao-ha-futebol-para-fascistas?utm_campaign=auto-tw&utm_source=twitter
&utm_medium=social
```

Figure 2: Printing the unique URLs

```

27         print "except"
28         listCount = listCount+1
29     else:
30         break

```

Listing 2: Python program for extracting links from my Twitter newsfeed

as shown above the code loops to find all Tweets found in my newsfeed, we basically put all the expanded URLs that are found in Twitter feed in a list. How can we assure that the link is Unique and not repetitive? This check happens in line 18. The if condition statement does not put the URL in the list unless it is unique. I kept all the list in a text file by writing every unique URL in a text file called urls.txt as shown in line 20.

2.2 The list

Listing the 1000 links I got will make this report unofficial so I referred to the text file I created that has all the URLs which is named urls.txt

3 Problem number 2 - Finding TimeMaps and Histogram

In this problem, it was required to download the TimeMaps for each link I got in the first problem. I basically downloaded a python code from Github github.com/joc11. The file is called timemap.py and it simply takes a url as an argument and print out the TimeMaps and mementos for this URL. But in my situation, I need to find the TimeMaps for all URLs I have, so I had to amend the code to be able to read the URLs from the urls.txt and loop to perform getting Time Maps for all the URLs I have. the code was as follows:

```

1  from datetime import datetime
2  import dateutil.parser
3  import re
4  import StringIO
5  import urllib
6  import urllib2
7  import re
8  from sys import argv
9
10 #=====
11 # KEYWORDS
12 #=====
13
14 #=====
15 # REGULAR EXPRESSIONS
16 #=====
17 i = 0
18 tokenizer = re.compile('(<[^>]+>|[a-zA-Z]+="[^"]*"|[""]+|;|,|\\s*)')
19
20 #=====
21 class TimeMap(object):

```

```

22 #
23
24 def __init__(self, uri=None, data=None):
25     self.original = None
26     self.timebundle = None
27     self.timegate = None
28     self.timemap = None
29     self.first_memento = None
30     self.last_memento = None
31     self.mementos = {}
32     self.__tokens = TimeMapTokenizer(uri, data)
33     link = self.get_next_link()
34     while link != None:
35         if link[0] == 'memento':
36             self.mementos[link[1]] = link[2]
37         elif link[0] == 'original':
38             self.original = link[2] if link != None else None
39         elif link[0] == 'timebundle':
40             self.timebundle = link[2] if link != None else None
41         elif link[0] == 'timegate':
42             self.timegate = link[2] if link != None else None
43         elif link[0] == 'timemap':
44             self.timemap = link[2] if link != None else None
45         elif link[0] == 'first memento':
46             self.mementos[link[1]] = link[2]
47             self.first_memento = link[1] if link != None else
48             None
49         elif link[0] == 'last memento':
50             self.mementos[link[1]] = link[2]
51             self.last_memento = link[1] if link != None else
52             None
53         link = self.get_next_link()
54
55 def get_next_link(self):
56     uri = None
57     datetime = None
58     rel = None
59     resource_type = None
60     for token in self.__tokens:
61         if token[0] == '<':
62             uri = token[1:-1]
63         elif token[:9] == 'datetime=':
64             datetime = token[10:-1]
65         elif token[:4] == 'rel=':
66             rel = token[5:-1]
67         elif token[:5] == 'type=':
68             resource_type = token[6:-1]
69         elif token[:6] == 'until=':
70             datetime = token[7:-1]
71         elif token == ';':
72             None
73         elif token == ',':
74             return ( rel, dateutil.parser.parse(datetime)
75                     if datetime != None else None,
76                     uri, resource_type )
77     else:
78         raise Exception('Unexpected timemap token', token)

```

```

77         if uri == None:
78             return None
79         else:
80             return ( rel , dateutil.parser.parse(datetime)
81                     if datetime != None else None,
82                     uri , resource_type )
83
84     def __getitem__(self, key):
85         return self.mementos[key]
86
87     #=====
88     class TimeMapTokenizer(object):
89     #=====
90
91     def __init__(self, uri=None, data=None):
92         if uri is not None:
93             self._tmfile = urllib2.urlopen(uri)
94         elif data is not None:
95             self._tmfile = StringIO.StringIO(data)
96         self._tokens = []
97
98     def __iter__(self):
99         return self
100
101     def next(self):
102         if len(self._tokens) == 0:
103             line = self._tmfile.readline()
104             if len(line) == 0:
105                 raise StopIteration
106             self._tokens = tokenizer.findall(line)
107         return self._tokens.pop(0)
108
109     #=====
110
111 # MAIN FOR TESTING
112 #=====
113
114 if __name__ == "__main__":
115     import logging
116     logging.basicConfig()
117     with open('urls.txt') as fil:
118         countOfzeromementos = 0
119         for line in fil:
120             urls=line
121             try:
122                 timemap_uri = "http://mementoproxy.cs.odu.edu/aggr/
123                             timemap/link/1/" + urls
124                 tm = TimeMap(uri=timemap_uri);
125                 print "Original:      ", tm.original
126                 print "Time Bundle:   ", tm.timebundle
127                 print "Time Gate:    ", tm.timegate
128                 print "Time Map:     ", tm.timemap
129                 print "First Memento:", tm.first_memento
130                 print "Last Memento: ", tm.last_memento

```

```

129         print "Mementos:"
130         for memento in sorted(tm.mementos.keys()):
131             print memento, "=", tm.mementos[memento] + "{0}".
                format(i)
132             i = i+1
133     except:
134         countOfzeromementos = countOfzeromementos +1
135         print url+ "has zero mementos, and the number of
            urls having zero mementos are : " + "{0}".format(
                countOfzeromementos)

```

Listing 3: Time map python code

In line 116, we Opened the text file to read all the URL's in there, and we looped on each URL to count the Time Maps, and if a 404 message is printed, then the corresponding link has zero mementos. that was kept track of in the exception part, from line 133.

Keeping in mind I kept track of the numbers of the links that have zero mementos, in line 134 of the code.

This a sample of the command prompt upon running the amended code:

We were required also to draw a Histogram to show the number of URLs and the mementos, for example 100 URL's having 0 mementos and 200 URL's having 1 memento and so on.

```

C:\Users\Samy\Desktop>cd A2-WebScience

C:\Users\Samy\Desktop\A2-WebScience>python timemap.py
Traceback (most recent call last):
  File "timemap.py", line 135, in <module>
    print links+ 'has zero mementos, and the number of urls having zero mementos
are : " + "{0}".format(countOfzeromementos)
NameError: name 'links' is not defined

C:\Users\Samy\Desktop\A2-WebScience>python timemap.py
http://movimail.co/the-new-justin-bieber-emoji-has-caused-a-social-media-frenzy/
has zero mementos, and the number of urls having zero mementos are : 1
https://twitter.com/xdarkinsjde/status/697449170194558977
has zero mementos, and the number of urls having zero mementos are : 2
https://twitter.com/carolacaracola5/status/697860515700330496
has zero mementos, and the number of urls having zero mementos are : 3
http://www.bez.es/231729894/La-vulnerabilidad-social-no-tiene-freno.html
has zero mementos, and the number of urls having zero mementos are : 4
https://twitter.com/YouthsDigest/status/697865960007589888
has zero mementos, and the number of urls having zero mementos are : 5
https://twitter.com/lauramoro325/status/697071929883234304
has zero mementos, and the number of urls having zero mementos are : 6
http://bit.ly/1LjviY8
has zero mementos, and the number of urls having zero mementos are : 7
https://movietvtechgeeks.com/ben-higgins-aka-the-bland-bachelor/?utm_source=twit
ter&utm_medium=social&utm_campaign=SocialWarfare
has zero mementos, and the number of urls having zero mementos are : 8
https://twitter.com/ya_boy_crump/status/697537690015813633
has zero mementos, and the number of urls having zero mementos are : 9
http://on.fb.me/1Ut9sqF
has zero mementos, and the number of urls having zero mementos are : 10
https://twitter.com/voxdotcom/status/697832498206789635
has zero mementos, and the number of urls having zero mementos are : 11
http://www.detroitbadboys.com/2016/2/10/10958844/chauncey-billups-best-point-gua
rd-nba-2002-08?utm_campaign=sean_corp&utm_content=chorus&utm_medium=social&utm_s
ource=twitter
has zero mementos, and the number of urls having zero mementos are : 12
https://twitter.com/keepitlitttt/status/697869783115563008
has zero mementos, and the number of urls having zero mementos are : 13
https://twitter.com/water/status/697828369954541570
has zero mementos, and the number of urls having zero mementos are : 14
http://social-tipp.de/0cdch
has zero mementos, and the number of urls having zero mementos are : 15
http://www.seattletimes.com/nation-world/female-tiger-killed-by-mating-partner-a
t-sacramento-zoo/?utm_source=twitter&utm_medium=social&utm_campaign=article_left
1.1
has zero mementos, and the number of urls having zero mementos are : 16

```

Figure 3: A sample for the URLs with their mementos

4 Problem number 3 Estimate the age of URLs

Finally the last part of the assignment was to estimate the age (Creation date) of each of the URLs, I downloaded Carbon date API and in the local.py file it was easy to pass a URL as an argument and print out its carbon date, but What is required is calculating the age of all URLs, so I implemented a function in cdate.py file to call the local.py downloaded within the Carbondate API, and pass every URL as an argument in a loop. so the final code will calculate the carbon date for all URLS. Then I copied and pasted the output in a file which to be read by a function in Python called q3.py which basically reads all the text file and disregard everything other than the URL and the corresponding Creation date.

```

1 import re
2 from sys import argv
3 txt = open("EstimatedTime.txt")
4 #print txt.read()
5 x=re.findall(r'\"(.+?)\"',txt.read())
6 lengthOfList = len(x)
7 counter=0
8 while(counter < lengthOfList):
9     check2 = x[counter-2]
10    check = x[counter]
11    if(check[0].isdigit()):
12        print check
13        if(check2[0] == "h"):
14            print check2
15    counter = counter +1

```

Listing 4: q3.py

Basically I put all the output in a text file and read it in a list, then as soon as I find a creation date, I print it out along with the corresponding URL, Keeping in mind I do not print out the URLs that do not have a creation date. Here is a figure which shows a sample of the output upon calling q3.py :

The following figure represents a graph between The Creation Dates and mementos for only the links that have both mementos and creation dates. X axis represents the age in Years and Y- axis represent the corresponding mementos. The Graph is based on the Output file shown above and Figure 3 which states the mementos for all URLs.

<http://inosmi.ru/social/20160208/235315525.html>
 2016-02-11T22:49:21
http://injo.com/2016/02/535008-two-presidential-candidates-left-the-campaign-trail-to-vote-on-this-senate-bill/?utm_source=Twitter
 2016-02-11T22:44:28
http://www.focus.de/regional/chemnitz/wer-kennt-diese-maenner-erst-klauen-sie-ein-auto-dann-werden-sie-geblitzt_id_5268569.html?utm_source=facebook
 2015-05-19T10:37:23
http://www.skyscanner.ru/news/deshevye-strany-dlia-puteshestvii-v-2016-godu?utm_medium=social+paid
 2016-02-11T22:47:54
https://blab.im/yuvraj-kewate-social-media-trends-and-tips-for-2016-http-bit-ly-23v0b0u?utm_source=twitter
 2013-10-19T21:21:08
<http://blog.hubspot.com/marketing/ultimate-guide-social-media-image-dimension-s-infographic>
 2015-07-19T11:46:55
<http://cgn.mx/es/>
 2016-02-11T08:57:01
http://www.newsminer.com/news/kris_capps/get-your-backyard-rinks-ready-for-hockey-week-contest/article_4d61ca40-d09d-11e5-9f61-63d744ea11ed.html?utm_medium=social
 2014-01-10T13:06:42
<http://smq.tc/SVkt1I>
 2016-02-09T01:43:21
http://www.unotv.com/noticias/estados/distrito-federal/detalle/legisladores-violan-reglamento-transito-815110/?utm_source=shared
 2016-02-11T22:49:39

Figure 4: A sample for the output from q3.py, URLs and their age

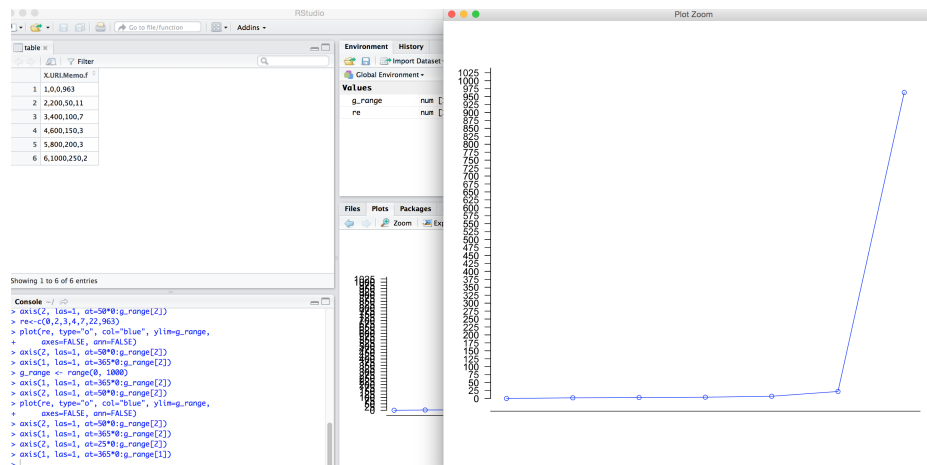


Figure 5: A sample for the output from q3.py, URLs and their age