Web Science: Assignment #3

Alexander Nwala

Puneeth Bikkasandra

Wednesday, February 21, 2018

Contents

Problem 1

Download the 1000 URIs from assignment 2. "curl", "wget", or "lynx" are all good candidate programs to use. We want just the raw HTML, not the images, stylesheets, etc.

Upload both sets of files to your Github account.

SOLUTION :

1. The program requires to run the ’CURL’ command over all URIs obtained during the Assignment 1, found in "1000TwitterLinks.txt".

   1. To get the raw HTML content for every URI :

   ```
     f = open(htmlFile, "w")
   subprocess.call(curlCommand, shell=True, stdout=f)
   ```

   All the raw HTML content will be saved in independent files in the current directory

   2. To get the text content for every URI :

   ```
   extractor = Extractor(extractor=’ArticleExtractor’, url=link)
   file.write(str(extractor.getText()
   ```

   Text content of the page will be saved in independent files with names obtained through **md5 hashing**

All the raw HTML and the text contents can be found in the **Content** directory on github submission

Listing 1: rawCode1.py

```

import subprocess
from boilerpipe.extract import Extractor
import sys
import hashlib
reload(sys)
sys.setdefaultencoding(’utf8’)

linksDict = {}
linksFile = open(’1000TwitterLinks.txt’,’r’)
for link in linksFile:
    if(link == ’’):
        pass
    else:
        try:
            curlCommand = ’curl ’ + link
            hash_object = hashlib.md5(link)
            print(hash_object.hexdigest() + ’.html’)
            htmlFile = hash_object.hexdigest() + ’:htmlFile’
            textFile = hash_object.hexdigest() + ’:txt’
            f = open(htmlFile, "w")
            raw_html = subprocess.call(curlCommand, shell=True, stdout=f)
```

```
            extractor = Extractor(extractor='ArticleExtractor', url=link)
            with open(textFile, 'w') as the_file:
                the_file.write(str(extractor.getText()))
                linksDict[textFile] = link
                print str(extractor.getText())

        except KeyboardInterrupt:
            exit()
        except:
            pass


with open('textURLFile', 'w') as file:
    for key,value in linksDict.items():
        file.write('%s:%s\n' % (key, value))
```

The above code will save the Text File name and the URI in to **"textURLFile.txt"** file, as attached in github.

Problem 2

Choose a query term (e.g., "shadow") that is not a stop word (see week 5 slides) and not HTML markup from step 1 (e.g., "http") that matches at least 10 documents (hint: use "grep" on the processed files). If the term is present in more than 10 documents, choose any 10 from your list.

Compute TFIDF values for the term in each of the 10 documents and create a table with the TF, IDF, and TFIDF values, as well as the corresponding URIs. The URIs will be ranked in decreasing order by TFIDF values.

SOLUTION

The solution for this problem is outlined by the following steps:

1. The program requires to run the 'GREP' command over all the text files obtained during Problem 1.
   1. GREP will find the text **"goals"** on the processed files

```
cmd = "grep -ci 'goals' * > grepOutput.txt"
subprocess.call(cmd, shell=True)
```

The output of the above step would return filenames of the processed files whose hit count is more than 0

   2. To calculate the terms to tabulate for every URI, the code determines TF ,IDF and TFIDF values

```
Term Frequency (TF) =
             Number of occurrences of the word / Total number of words
Inverse Document Frequency (IDF) =
   log2(total URIs in corpus/total processed files with term 'goals')
TF-IDF = TF * IDF
```

Listing 2: rawCode2.py

```python
import subprocess
import math

cmd = "grep -ci 'goals' * > grepOutput.txt"
subprocess.call(cmd, shell=True)
count = 0
matchCount = round(0,4)
corpusCount = round(0,4)
docsWithTerm = round(0,4)
idf = round(0,4)
tf = round(0,4)
tfidf = round(0,4)
totalWordsInFile = round(0,4)
num_words = 0
tfDict = {}

def countWordsInFile(fileName):
    global num_words
    with open(fileName, 'r') as f:
        for line in f:
            words = line.split()
            num_words += len(words)
    return num_words

linksFile = open('grepOutput.txt','r')
for line in linksFile:
    line = line.replace('\n', '')

    if(':txt' in line):
        matchCount = round(int(line[(line.rfind(':')+1):]),4)
        corpusCount = corpusCount + 1
        if(matchCount >= 1):
            line  = line[0:line.rfind(':')]
            totalWordsInFile = countWordsInFile(line)
            print ('line',line)
            print('Total Words :',totalWordsInFile)
            print ('matchCount',matchCount)
            docsWithTerm = docsWithTerm + 1
            tf = round((matchCount / totalWordsInFile),4)
            if(tf >= 0.0003):
                tfDict[line] = tf
            print('\n')
    else:
        continue

# idf = round((corpusCount / docsWithTerm),4)

idf = math.log(round((corpusCount / docsWithTerm),4)) / math.log(2)
tfidf = round((tf * idf),4)

print('TFIDF    TF    IDF         URL')
print('-----    --    ---         ---')

data = dict()
with open('textURLFile','r') as raw_data:
    for item in raw_data:
        if ':txt' in item:
            key,value = item.split(':txt:', 1)
            data[key]=value
        else:
            pass

for key, value in tfDict.items():
      key = key[0:key.rfind(':')]
      tfIdfValue = round((float(value) * idf),4)
      tfValue = value
      url = data[key]
      print(str(tfIdfValue)+'   '+str(tfValue)+'   '+str(idf)+'   '+url)
```

The output of the code execution can be found in the **"P3_Output.txt"**

Listing 3: Extracted ratios

```

TFIDF    TF    IDF           URL
-----    ---   ---           ---
0.001   0.0003   3.3513315059   https://www.thescottishsun.co.uk/sport/football
/2216776/huddersfield-star-accidentally-flashes-xxxxx-to-millions-worldwide-on-
live-tv-during-premier-league-clash-with-bournemouth/

0.0013   0.0004   3.3513315059   https://www.chelseafc.com/news/latest-news
/2018/02/home-and-away--eidur-gudjohnsen.html

0.0017   0.0005   3.3513315059   http://www.independent.co.uk/sport/football/
premier-league/manchester-united-munich-air-disaster-wembley-sport-greatest
-ever-stories-a8195736.html

0.001   0.0003   3.3513315059   http://www.gulf-times.com/story/581358/Nepal
-plays-2-2-against-Lebanon-in-Asian-Football-

0.001   0.0003   3.3513315059   http://www.und.com/sports/m-baskbl/spec-rel/
021118aaa.html

0.001   0.0003   3.3513315059   https://pechalbata.com/super-vip-football-
predictions-2-02-2018/

0.001   0.0003   3.3513315059   https://www.birminghammail.co.uk/sport/football/
football-news/john-terry-villa-birmingham-city-14276580

0.001   0.0003   3.3513315059   https://www.mirror.co.uk/sport/football/
match-reports/electric-egyptian-mo-salah-just-12009750

0.0013   0.0004   3.3513315059   https://www.onefootdown.com/2018/2
/11/17000964/womens-lacrosse-notre-dame-loses-boston-college-maddie
-howe-nikki-ortega-irish-eagles-acc-sam-apuzzo

0.001   0.0003   3.3513315059   https://www.liverpoolecho.co.uk/sport/football/
football-news/jurgen-klopp-reveals-details-conversation-14276564

0.0044   0.0013   3.3513315059   https://www.coventrytelegraph.net/sport/football/
football-news/coventry-city-fc-player-ratings-14273283?service=responsive

0.0158   0.0047   3.3513315059   https://www.sportsmole.co.uk/football/wolves/news/
nuno-wolves-worthy-winners-against-qpr_318467.html

0.001   0.0003   3.3513315059   https://itunes.apple.com/us/app/soccer-
agent-football-manager/id1021569826?mt=8

0.001   0.0003   3.3513315059   https://www.youtube.com/watch?v=bKMZmEeqHQA&
feature=youtu.be

0.002   0.0006   3.3513315059   http://www.skysports.com/football/news/
11661/11245990/southampton-0-2-liverpool-mohamed-salah-roberto-firmino-
star-on-virgil-van-dijk-return
```

Problem 3

Now rank the same 10 URIs from question 2, but this time by their PageRank. Use any of the free PR estimaters on the web, Such as:

http://pr.eyedomain.com/
http://www.prchecker.info/check_page_rank.php
http://www.seocentro.com/tools/search-engines/pagerank.html
http://www.checkpagerank.net/

SOLUTION

The below Page Ranks are obtained manually from the below ranking site :

http://www.checkpagerank.net/

Listing 4: PageRank

```
PageRank        URI
--------        ---
0.5             https://www.thescottishsun.co.uk
0.7             https://www.chelseafc.com
0.9             http://www.independent.co.uk
0.7             http://www.gulf-times.com
0.5             http://www.und.com
0.4             https://pechalbata.com
0.7             https://www.birminghammail.co.uk
0.8             https://www.mirror.co.uk
0.5             https://www.onefootdown.com
0.6             https://www.liverpoolecho.co.uk
0.8             https://www.coventrytelegraph.net
0.5             https://www.sportsmole.co.uk
0.9             https://itunes.apple.com
1               https://www.youtube.com
0.7             http://www.skysports.com
```