INTRODUCTION TO WEB SCIENCES: Assignment 3

Babitha Bokka

4 october 2014

Contents

1 Question 1

Download HTML content of 1000 URIs extracted in assignment 2.

1.1 Approach Towards the Solution

There are many good ways to extract the HTML from a URL. I approached the problem by using

1. Requests
2. urllib2
3. curl

1.1.1 Desciption of extractHtml.sh

1. Open ,Read each line from uniqueUri.txt.
2. Generate a md5 for each URI and store in a file.
3. Using curl extract the HTML content .

1.1.2 Desciption of scrapeHtml.sh

1. Open a folder extract each file.
2. Get the basename of the file.
3. Using lynx get the data by stripping of the HTML.
4. Store it in Plain file.

1.2 Observation

Due to the different approaches to extract the HTML content an observation that curl extracts more HTML content that requests and urllib2 library in python.So ,for this part of the assignment Shell script using curl is the best approach.

1.3 Source Code to extract the HTML content

1.3.1 extractHtml.sh

```
#!/bin/bash

if  [ $# -ne 1 ]
then
    echo "usage <extractHtml.sh> <filename> "
    echo "e.g., extractHtml.sh  uniqueUri.txt "
    exit
fi
md5uri="md5Uri.txt"
filename=`readlink -f $1`

for line in `cat $filename`
do
    md5=$line
    hash="$(echo " $md5 "|md5sum | cut -f1 -d' ' )"
    echo "$line  $hash" >> $md5uri
    curl -A "Mozilla/4.0" --connect-timeout 30 $line  -o "$hash.htm"
done
```

1.3.2 requestsExtractHtml.py

```
#!/usr/bin/env python
import md5
import requests
import urllib2
import socket

# Main Function
def main():
  f        = open('uniqueUri.txt', 'r')
  saveFile   = open('md5_uri.txt','w')
  count = 0
  connect = 0
  socket = 0
  timeo =0
  unko=0
  for url in f.readlines():
    hash_md5  = md5.new(url).hexdigest()
    saveFile.write("{:<10}{} " .format( hash_md5 ,url))
    try :
      response    = requests.get(url,timeout = 30)
      html_content= response.content

    except requests.exceptions.Timeout:
      timeo=timeo+1
      pass
    except requests.exceptions.ChunkedEncodingError:
      unko=unko+1
      pass
    except requests.exceptions.ConnectionError :
      connect=connect+1
      pass
    except socket.timeout:
      socket = socket+1
      pass

    filename     = "%s.htm" % hash_md5
    count =count +1
    print filename   , hash_md5  ,'----',  url  ,'**',count,connect,socket,  timeo,unko
    # writing content to file created
    content_file= open(filename,"w")
    content_file.write(html_content )
    content_file.close()
  print connect
  saveFile.close()
  f.close()


if __name__ =="__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)


#os.system("wget -O  hash_md5 %s" %url)

# request      = urllib2.Request(url)
# response     = urllib2.urlopen(url,timeout=20)
# html_content= response.read()
```

1.3.3 urllibExtractHtml.py

```
#!/usr/bin/env python
import md5
import sys
import urllib2
import socket

# Main Function
def main():
  f        = open('uniqueUri.txt',  'r')
  saveFile   = open('md5_uri.txt','w')
  count = 0
  for url in f.readlines():
    hash_md5   = md5.new(url).hexdigest()
    saveFile.write("{:<10}{} " .format( hash_md5  ,url))
    try  :
      request      = urllib2.Request(url)
      response     = urllib2.urlopen(url,timeout=30)
      html_content= response.read()

    except urllib2.HTTPError:
      pass
    except urllib2.URLError  :
      pass
    except socket.timeout  :
      pass

    filename     = "%s.htm" % hash_md5
    count =count +1
    print filename  , hash_md5 , url ,count
    # writing content to file created
    content_file= open(filename,"w")
    content_file.write(html_content )
    content_file.close()

  saveFile.close()
  f.close()


if __name__ =="__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

1.4 Source Code to strip the HTML from raw files

1.4.1 stripHtml.sh

```
#!/bin/bash

# to scrape the html from the files.
# checking the arguments
if  [ $# -ne 1 ]
then
    echo "usage stripHtml.sh <folder name> "
    echo "e.g., stripHtml  <html> "
    exit
fi

dir=`readlink -f "$1"`
for file in `ls $dir`
do
    echo "helo"
    plain=`basename "$file" .htm`
    lynx -dump -force_html $dir/$file > $plain.txt
done
```

1.5 InputFile : uniqueUri.txt

The input is taken from the assignment 2 , the 1000 unique URIs.

```
Sample Unique URI's :
https://www.facebook.com/MyChickenRun
http://www.youtube.com/watch?v=Eubi9YI2dKE
http://geladoesntgiveadamn.tumblr.com/
http://youtu.be/1BKO2V9EaZ0?a
http://www.instagram.com/the_sanging_rebel
http://instagram.com/teustimao/
http://blogmylunch.com
http://Facebook.com/Robbiewhaylez
http://ifoodi.blogspot.com/
http://www.talkinggoodfood.co.uk
http://facebook.com/dimano.sterling
http://www.yellowkorner.com
http://Emerald.com
http://attackontiphan.tumblr.com
http://instagram.com/xxxmn88
http://linggez-network.blogspot.com
http://Instagram.com/hellocalm_
http://www.wagamama.com
http://pennyroyaltea.co.vu
http://www.oufancyphones.com
http://www.facebook.com/rappstartailgate
http://thedaintypig.com
http://instagram.com/victorparrini
http://instagram.com/_lafamiliaonly
http://rad-acid.tumblr.com
https://www.Youtube.com/iFarLiez
http://twitter.com
http://melonpatchtv.com
http://linkd.in/1qVvTtZ
http://www.huffingtonpost.com/alex-palombo/
http://youtube.com/user/IvanAlvir
http://www.musicsumo.com
http://www.KhanaPakana.com
http://www.streetdeal.sg/home/refer/60366/1384835006?utm_medium=Friends
http://www.oliviarosenman.wordpress.com/
https://www.facebook.com/TIMLLAINE?ref=tn_tnmn#
http://Instagram.com/khourtni_hearts
http://Ask.fm/Rebecca3440
http://es.favstar.fm/users/maroto43
http://Instagram.com/princess.ri
http://byunips.tumblr.com
http://viciousnoodles.com
http://www.youtube.com/user/cece020304
http://www.rochesterbuzz.com
http://instagram.com/geneveealcala
http://soundcloud.com/sohodusk
http://ilovenudes.com
http://www.facebook.com/dedrion
http://thenegress.wordpress.com
http://manvspink.com
http://introvertedaquarius.tumblr.com
```

1.6 OutputFiles

1.6.1 md5Uri.txt

A sample of URIs and their MD5 hash code.

```
URIs :                                        md5:
https://www.facebook.com/MyChickenRun         2c46a0201b5a19f93e19a0ace98cfb92
http://www.youtube.com/watch?v=Eubi9YI2dKE    e8ea8e5ac1500a2dad1bfcbb27239ac4
http://geladoesntgiveadamn.tumblr.com/        39c8ba3ab8ddbf67bae6dca6c1e3a285
http://youtu.be/1BKO2V9EaZ0?a                 ed2a0dc185bb8ee25de06f32f87110c3
http://www.instagram.com/the_sanging_rebel    896e23c0050291fe1a7d15ca60b357ed
http://instagram.com/teustimao/               f91bc9705dfb9015c835d0884fc848a1
http://blogmylunch.com                        5111b1b289f418d068d342d5e28d1e1a
http://Facebook.com/Robbiewhaylez             61860dda7e53c0b3480a1259960b0467
http://ifoodi.blogspot.com/                   76dd67f38f9fca99ed242972b4b21498
http://www.talkinggoodfood.co.uk              f58a7a3e519801a05310050a1fb43b8c
http://facebook.com/dimano.sterling           d07861c98572c5581ac2a4af5eeaae56
http://www.yellowkorner.com                   8e3157838aa258c1ec81547bf4124a2b
```

1.6.2 2c46a0201b5a19f93e19a0ace98cfb92.htm

A sample of raw HTML file.

```
<!DOCTYPE html>
<html lang=”en” id=”facebook” class=”no_js”>
<head><meta charset=”utf-8” /><script>function envFlush(a){function b(c){for(var d in
    a)c[d]=a[d];} if(window.requireLazy){window.requireLazy([’Env’],b);}else{Env=window
    .Env||{};b(Env);}}envFlush({”ajaxpipe_token”:”AXg_zUjmAHi6xr5M”,”lhsh”:”bAQFjDuMG
    ”});</script><script>CavalryLogger=false;</script><noscript><meta http-equiv=”
    refresh” content=”0; URL=/MyChickenRun?_fb_noscript=1” /></noscript><meta name=”
    referrer” content=”default” id=”meta_referrer” /><title id=”pageTitle”>My Chicken
    Run | Facebook</title><meta property=”og:title” content=”My Chicken Run” /><meta
    property=”og:type” content=”website” /><meta property=”og:url” content=”https://
    www.facebook.com/MyChickenRun” /><meta property=”og:site_name” content=”Facebook”
    /><meta property=”og:image” content=”https://fbcdn-profile-a.akamaihd.net/hprofile
    -ak-xfp1/v/t1.0-1/p200x200/1170678_625163597516230_1617126590_n.jpg?oh=96366477
    df4e67f07408fe74fb2f37e4&amp;oe=54BCA8EE&amp;__gda__=1422606750
    _b6dbfe09753016bcc0a8638281ed238b” /><meta property=”og:description” content=”
    Chickens bring joy, amusement, fun and entertainment to any garden. My girls are
    Penny, Vicky,...” /><link rel=”alternate” media=”only screen and (max-width: 640px
    )” href=”https://www.facebook.com/MyChickenRun” /><link rel=”alternate” media=”
    handheld” href=”https://www.facebook.com/MyChickenRun” /><meta name=”description”
    content=”My Chicken Run. 3,045 likes &#xb7; 157 talking about this. Chickens bring
     joy, amusement, fun and entertainment to any garden. My girls are Penny, Vicky
    ,...” /><meta name=”robots” content=”noodp,noydir” /><noscript><meta http-equiv=”X
    -Frame-Options” content=”DENY” /></noscript><link rel=”shortcut icon” href=”https
    ://fbstatic-a.akamaihd.net/rsrc.php/yV/r/hzMapiNYYpW.ico” />
```

1.6.3 2c46a0201b5a19f93e19a0ace98cfb92.txt

A sample of processed HTML file.

```
   REFRESH(0 sec): [1]file://localhost/MyChickenRun?_fb_noscript=1
   #[2]alternate [3]alternate

   [4]Facebook logo
   Email or Phone        Password
   -------------------- --------------------    Log In
   [ ] Keep me logged in
                          [5]Forgot your password?

References

   1. file:///MyChickenRun?_fb_noscript=1
   2. https://www.facebook.com/MyChickenRun
   3. https://www.facebook.com/MyChickenRun
   4. file:///
   5. https://www.facebook.com/recover/initiate
```

1.7 Learnings

1. Shell script
2. Python

1.8 Mistakes

1. For some reason “lynx” program was not extracting the data from the htm files instead it had an error message “This site requires JavaScript and Cookies to be enabled. Please change your browser settings or upgrade your browser”. The files had this error message because in lynx program the path to the raw files if not specified so, that the lynx program would find them and strip out the html since there is no path the files had this error message.
2. One should always check for “CR” and “LF” returns from windows to unix environment.

1.9 Testing

To test why the error message is stored in the plain files instead of actual data I loaded my raw file into my friend Mallika program then I realized the mistake and learnt why it happened.

2 Question 2

Choose a keyword that matches atleast 10 documents .Count the occurrence of word and total number of words in each document and calculate the TF,IDF ,TF-IDF values.

2.1 Approach Towards the Solution

Keyword to searched in the extracted data is “food” . There are 835 total documents and 84 docs with the key word .This is when we assume that our the 1000 unique links are the total corpus.so,the IDF:
total docs in corpus = 835
docs with term = 84
lets assume the total corpus is 20B if bing has 20B documents indexed
then
total docs in corpus = 20000000000
docs with term = 653000000

the TF values would be keyword count over total number of words.

2.1.1 Desciption of findKeyword.sh

1. Find a keyword and check the count in all the documents.
2. Use grep to count the keyword in all the processed files.
3. save the count and file name .

2.1.2 Desciption of calculationProgram.py

1. Open ,read the file cal.txt
2. File has key word count(k_count),total count(t_count),filename,URI.
3. Store the (k_count),(t_count)and compute the TF,IDF,TFIDF.
4. The results are stored in output1.txt and output2.txt.

2.2 Source Code

2.2.1 shellCommands.txt

```
Shell script commands:

cat *.txt | grep -c food                                  ---> To know total count of the
    keyword in all files.
./findKeyword.sh plain_content | sort -rn | head -n 10 ---> To sort in reverse order
    and take the top of sorted list.
```

2.2.2 findKeyword.sh

```
#!/bin/bash

if  [ $# -ne 1 ]
then
    echo "usage findKeyword.sh <folder name> "
    echo "e.g., findKeyword  <html> "
    exit
fi

plain="/home/bbokka/cs594/Assignment/programs/plain_content"

dir=`readlink -f "$1"`
for file in `ls $dir`
do
  filename=$plain/"$file"
    var=`grep -c "food" "$filename"`
    var1=`wc -w $filename`
    echo "$var  $file " #>> word_count.txt
done
```

2.2.3 calculationProgram.py

```
#!/usr/bin/env python

import sys
import math

# Main Function
def main():
    total_doc_corpus = 20000000000
    docs_with_term   = 653000000

    #open the file to read word_count , total_count ,
    f = open('cal.txt', 'r')

    inverse_term = math.log(
        (float(total_doc_corpus) / docs_with_term),
        2
    )


    for line in f.readlines():
        # splitting the line based on space
        data        = line.split()
        # data is stored the data list so pull out using index values
        # url is 4rth element stored at 3 index
        url         = data[3]
        # word_count is 1st element stored at 0 index
        word_count  = int(data[0])
        # total_count is 2nd element stored at 1 index
        total_count = int(data[1])
        # TF calculation
        term_frequency = float(word_count)/total_count
        # TFIDF calculation
        term_inverse_frequency= term_frequency* inverse_term

        print url , word_count, total_count,term_frequency ,inverse_term ,
    term_inverse_frequency


if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        sys.exit(1)
```

2.3 Output Files

2.3.1 sort.txt

```
k_count:   Preocessed files:
1435       76dd67f38f9fca99ed242972b4b21498.txt
86         291612f0391cedca5954c52fa124aea1.txt
81         9c0de32a26ec4b3e251c882e80349e5d.txt
56         e5177cd8cf0fbaa16116f2edb8ae8eca.txt
56         071d5c179ef3d88beafacc5468a3f3ed.txt
44         9e75fbae0686c37c21b78bf3008fe88b.txt
22         cff62b7f415bc539174a200b5bb32e62.txt
22         c1389f478aa62dfba5eb0637b080c77a.txt
20         e0397dc1d2e5760be4a952550aa801e1.txt
16         b65368dab975f7eece39ec337641f839.txt
```

2.3.2 countData.txt

```
k_count  t_count             file                                    URI
1435     29780   76dd67f38f9fca99ed242972b4b21498.txt   http://ifoodi.blogspot.com/
86       1707    291612f0391cedca5954c52fa124aea1.txt   http://turtlestravel.com
81       14190   9c0de32a26ec4b3e251c882e80349e5d.txt   http://howchow.blogspot.com/
56       4107    071d5c179ef3d88beafacc5468a3f3ed.txt   http://thescienceofeating.com
56       1111    e5177cd8cf0fbaa16116f2edb8ae8eca.txt   http://www.cookinglight.com
44       503     9e75fbae0686c37c21b78bf3008fe88b.txt   http://foodnex.tumblr.com
22       6850    c1389f478aa62dfba5eb0637b080c77a.txt   http://baconforthesoul.
   wordpress.com
22       3044    cff62b7f415bc539174a200b5bb32e62.txt   http://rockinrina.tumblr.com
20       1709    e0397dc1d2e5760be4a952550aa801e1.txt   http://thedaintypig.com
16       3929    b65368dab975f7eece39ec337641f839.txt   http://oxymoron101.wordpress.
   com/
```

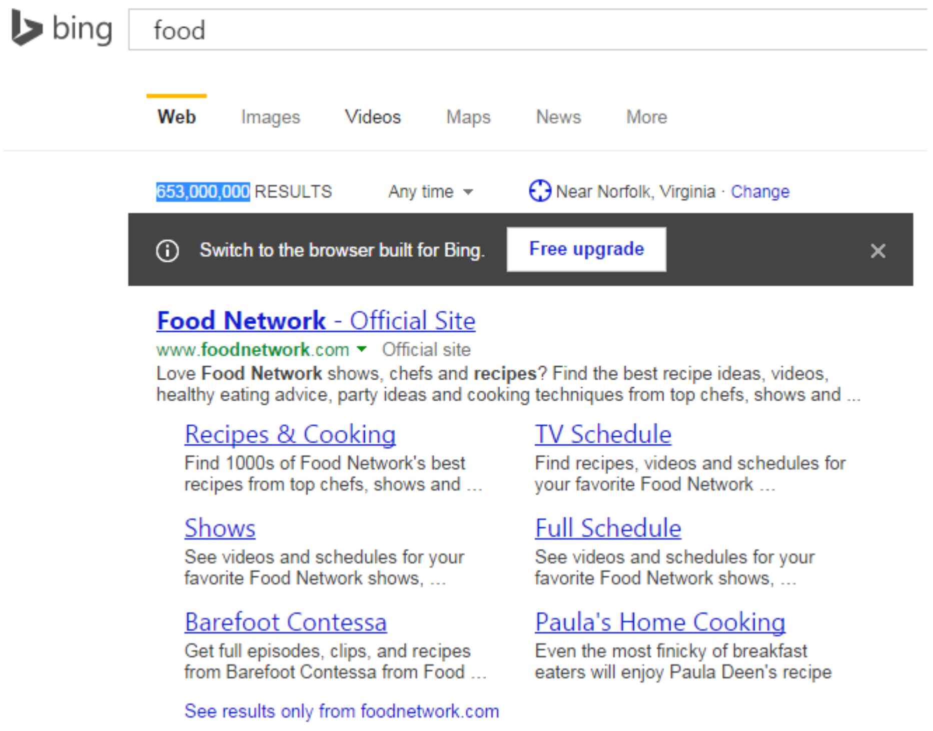
2.3.3 Assuming the total docs in corpus is the total (835/1000) unique URIs we are dealing with and docs with term is the number of docs with the keyword(84/835)

```
URI                                      k_count  t_count    TF      IDF   TFIDF

http://ifoodi.blogspot.com/                 1435  29780   0.048   3.313   0.159
http://turtlestravel.com                    86    1707    0.050   3.313   0.166
http://howchow.blogspot.com/                81    14190   0.005   3.313   0.018
http://thescienceofeating.com               56    4107    0.013   3.313   0.045
http://www.cookinglight.com                 56    1111    0.050   3.313   0.167
http://foodnex.tumblr.com                   44    503     0.087   3.313   0.289
http://baconforthesoul.wordpress.com        22    6850    0.003   3.313   0.010
http://rockinrina.tumblr.com                22    3044    0.007   3.313   0.023
http://thedaintypig.com                     20    1709    0.011   3.313   0.038
http://oxymoron101.wordpress.com/           16    3929    0.004   3.313   0.013
```

2.3.4 Assuming the total docs in corpus is the total (20B)and docs with term is the number of docs with the keyword(65M)

```
URI                                  k_count  t_count   TF       IDF   TFIDF

http://ifoodi.blogspot.com/             1435  29780   0.048   4.936  0.237
http://turtlestravel.com                86    1707    0.050   4.936  0.248
http://howchow.blogspot.com/            81    14190   0.005   4.936  0.028
http://thescienceofeating.com           56    4107    0.013   4.936  0.067
http://www.cookinglight.com             56    1111    0.050   4.936  0.248
http://foodnex.tumblr.com               44    503     0.087   4.936  0.431
http://baconforthesoul.wordpress.com 22    6850    0.003   4.936  0.015
http://rockinrina.tumblr.com            22    3044    0.007   4.936  0.035
http://thedaintypig.com                 20    1709    0.011   4.936  0.057
http://oxymoron101.wordpress.com/       16    3929    0.004   4.936  0.020
```

2.3.5 screenShot



653,000,000 RESULTS Any time Near Norfolk, Virginia · Change

Switch to the browser built for Bing. Free upgrade

Food Network - Official Site
www.foodnetwork.com Official site
Love Food Network shows, chefs and recipes? Find the best recipe ideas, videos, healthy eating advice, party ideas and cooking techniques from top chefs, shows and ...

Recipes & Cooking
Find 1000s of Food Network's best recipes from top chefs, shows and ...

TV Schedule
Find recipes, videos and schedules for your favorite Food Network ...

Shows
See videos and schedules for your favorite Food Network shows, ...

Full Schedule
See videos and schedules for your favorite Food Network shows, ...

Barefoot Contessa
Get full episodes, clips, and recipes from Barefoot Contessa from Food ...

Paula's Home Cooking
Even the most finicky of breakfast eaters will enjoy Paula Deen's recipe

See results only from foodnetwork.com

Figure 1: docs with term on web

TFIDF	**TF**	**IDF**	**URI**
0.159	0.048	3.313	`http://ifoodi.blogspot.com`
0.166	0.050	3.313	`http://turtlestravel.com`
0.018	0.005	3.313	`http://howchow.blogspot.com`
0.045	0.013	3.313	`http://thescienceofeating.com`
0.167	0.050	3.313	`http://www.cookinglight.com`
0.289	0.087	3.313	`http://foodnex.tumblr.com`
0.010	0.003	3.313	`http://baconforthesoul.wordpress.com`
0.023	0.007	3.313	`http://rockinrina.tumblr.com`
0.038	0.011	3.313	`http://thedaintypig.com`
0.013	0.004	3.313	`http://oxymoron101.wordpress.com`

Table 1: Total docs in corpus =835:docs with term = 84

TFIDF	**TF**	**IDF**	**URI**
0.237	0.048	4.936	`http://ifoodi.blogspot.com`
0.248	0.050	4.936	`http://turtlestravel.com`
0.028	0.005	4.936	`http://howchow.blogspot.com`
0.067	0.013	4.936	`http://thescienceofeating.com`
0.248	0.050	4.936	`http://www.cookinglight.com`
0.431	0.087	4.936	`http://foodnex.tumblr.com`
0.015	0.003	4.936	`http://baconforthesoul.wordpress.com`
0.035	0.007	4.936	`http://rockinrina.tumblr.com`
0.057	0.011	4.936	`http://thedaintypig.com`
0.020	0.004	4.936	`http://oxymoron101.wordpress.com`

Table 2: Total docs in corpus = 20B:docs with term = 65M

3 Question 3

Get the page rank for 10 URIs from question 2.

3.1 Approach Towards the Solution

Go to the provided link and type the URL and solve the captcha to prove that you are human. The result will be the page rank to normalize it divide it by 10 or you can take the highest value and divide each page rank with the highest value .

3.1.1 Desciption

1. Go to `http://www.prchecker.info/check_page_rank.php`.
2. Solve the captcha.
3. Write down the page rank given note that all pages on web are not ranked and the result of them will be not avialable.

3.2 Source

3.2.1 sourceLink.txt

```
 source to get the page rank of the URIs

 http://www.prchecker.info/check_page_rank.php
```

3.3 Output Files

3.3.1 pageRank.txt

```
page_rank                URIs

N/A                     http://ifoodi.blogspot.com
3                       http://turtlestravel.com
4                       http://howchow.blogspot.com
2                       http://thescienceofeating.com
7                       http://www.cookinglight.com
3                       http://foodnex.tumblr.com
N/A                     http://baconforthesoul.wordpress.com
N/A                     http://rockinrina.tumblr.com
1                       http://thedaintypig.com
1                       http://oxymoron101.wordpress.com
```

Page Rank	URI
0	`http://ifoodi.blogspot.com`
0.3	`http://turtlestravel.com`
0.4	`http://howchow.blogspot.com`
0.2	`http://thescienceofeating.com`
0.7	`http://www.cookinglight.com`
0.3	`http://foodnex.tumblr.com`
0	`http://baconforthesoul.wordpress.com`
0	`http://rockinrina.tumblr.com`
0.1	`http://thedaintypig.com`
0.1	`http://oxymoron101.wordpress.com`

Table 3: Normalized page rank

3.4 Compare and Contrast rankings in Question 2 and Question 3

When we compare the rankings based on TFIDF and page rank there is no relation between them.Rankings given by page rank and TFIDF have no corelation when you observe the page rank of `http://ifoodi.blogspot.com` and TFIDF value the page rank is 0 and TFIDF value is 0.237 . So, we cannot compare the values of page rank and TFIDF.

References

[1] pagerank. http://www.prchecker.info/check_page_rank.php.

[2] shellscript. http://www.thegeekstuff.com/2009/03/15-practical-unix-grep-command-examples/.

[]