CS 595: Assignment #3

Due on Thursday, October 3, 2014

Dr. Nelson 4:20pm

Holly Harkins

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.
from the command line:


% curl http://www.cnn.com/ > www.cnn.com
% wget -O www.cnn.com http://www.cnn.com/
% lynx -source http://www.cnn.com/ > www.cnn.com

"www.cnn.com" is just an example output file name, keep in mind
that the shell will not like some of the characters that can occur
in URIs (e.g., "?", "&").You might want to hash the URIs,like:

% echo -n "http://www.cs.odu.edu/show_features.shtml\?72" | md5
41d5f125d13b4bb554e6e31b6b591eeb


("md5sum" on some machines; note the "\- n" in echo \-\- this removes
the trailing newline.) Now use a tool to remove (most) of the HTML markup.  "lynx" will
do a fair job:


% lynx -dump - force_html www.cnn.com > www.cnn.com.processed

Use another (better) tool if you know of one.  Keep both files
for each URI (i.e., raw HTML and processed).
```

Problem 1 continued on next page...

Listing 1: Python 1000 Links

```
# -*- encoding: utf-8 -*-

import urllib2
import re
import subprocess
from subprocess import call
import md5
import os

##New Dir for Output
dir1_name='../Pages'
try:
    os.makedirs(dir1_name)

##Grab Final List- 1000 Links
file1=open("FinalList.txt","r")

lineCount=0
for line in file1:
    lineCount=lineCount+1

    ##Remove \n
    Oneline=line.rstrip("\n")

    ##Curl
    try:
      r = urllib2.Request(Oneline)
      l = urllib2.urlopen(r)

      ##Check for 200 Response
      if l.code==200:
       allData=l.read()

       ##Create Hash, md5
       test=md5.new(Oneline)
       filename=test.hexdigest()

       with open(os.path.join(dir1_name,filename+'.txt'), 'w') as file3:
          file3.write(allData)

       ##Lynx
       cmd = os.popen("lynx -dump -force_html %s %s" %(line,filename))
       output_no_tags = cmd.read()
       cmd.close()
       with open(os.path.join(dir1_name,filename+'.processed.txt'), 'w') as file4:
          file4.write(output_no_tags)

file1.close()
```

Problem 2

```
Choose a query term (e.g., "shadow") that is not a stop word
(see week 4 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.  (If you do not end up with a list of 10
URIs, you've done something wrong).

As per the example in the week 4 slides, 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.  For
example:

Table 1. 10 Hits for the term "shadow", ranked by TFIDF.

TFIDF TF IDF URI
-------------
0.150 0.014 10.680 http://foo.com/
0.085 0.008 10.680 http://bar.com/


You can use Google or Bing for the DF estimation.  To count the
number of words in the processed document (i.e., the deonminator
for TF), you can use "wc":

% wc -w www.cnn.com.processed
    2370 www.cnn.com.processed
```

Answer: term "redskins" - 81 matches
The list of the selected URIs:

```
http://www.redskins.com/news-and-events/article-1/Redskins-Prepare-Minds-Bodies-
For-Short-Week/56ce5efd-288c-4d88-9204-b48338ce9d7acampaign=social_20140924_32223356/
https://twitter.com/Redskins/status/514818552491036673/photo/1/
https://fedexfield.clickandpark.com/campaign=social_20140923_32175316/
http://www.redskins.com/media-gallery/videos/Garcon-Talks-The-Play-Of-QB-Kirk-
Cousins/ae390f87-36ae-4b36-82fa-b6091eaa804ecampaign=social_20140923_32169556/
http://blog.redskins.com/2014/09/24/kevin-durants-birthday-featured-redskins-jersey-
on-cake-burgundy-and-gold-shoes/
http://www.nfl.com/videos/nfl-countdowns/0ap3000000398878/Week-3-Top-5-catches
campaign=social_20140923_32147506/
https://twitter.com/Redskins/status/514874437829533696/photo/1/
http://blog.redskins.com/2014/09/24/dale-earnhardt-jr-records-pick-six-with-ryan-
kerrigan-on-madden/campaign=social_20140924_32213426more-20020201/
http://www.snappytv.com/snaps/tnf_pregame_crowd_nyg_was_15_h264_092214_nb5lze/45723/
https://twitter.com/Redskins/status/514814324599033856/photo/1/


grep o redskins *.processed | uniq c
```

```
Bing.com = 18,900,000
Google.com = 3,570,000


RANK TF IDF TFIDF SITE
```

Problem 3

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

http://www.prchecker.info/check_page_rank.php
http://www.seocentro.com/tools/s

Page Rank
1.0 https://twitter.com/
0.7 http://www.nfl.com/
0.6 http://www.redskins.com/
0.5 http://blog.redskins.com/
0.4 http://www.snappytv.com/
0.0 https://fedexfield.clickandpark.com/

References

[1] http://www.velocityreviews.com/forums/t357410-md5-from-python-different-then-md5-from-command-line.html

[2] http://www.prchecker.info/checkpagerank.php

[3] http://en.wikipedia.org/wiki/Correlationanddependence