CS 495/595 Introduction to Web Science

Fall 2013

http://www.cs.odu.edu/~mln/teaching/cs595-f13/

Assignment #3

Due: 11:59pm Oct 5

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).

2. Choose a query term (e.g., "shadow") that is not a stop words (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/

0.044 0.008 5.510 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 denominator for TF), you can use "wc":

% wc -w www.cnn.com.processed

2370 www.cnn.com.processed

It won't be completely accurate, but it will be probably be consistently inaccurate across all files. You can use more accurate methods if you'd like.

Don't forget the log base 2 for IDF, and mind your significant digits!

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://www.prchecker.info/check\_page\_rank.php

http://www.seocentro.com/tools/search-engines/pagerank.html

http://www.checkpagerank.net/

If you use these tools, you'll have to do so by hand (they have anti-bot captchas), but there is only 10. Normalize the values they give you to be from 0 to 1.0. Use the same tool on all 10 (again, consistency is more important than accuracy).

Create a table similar to Table 1:

Table 2. 10 hits for the term "shadow", ranked by PageRank.

PageRank URI

-------- ---

0.9 http://bar.com/

0.5 http://foo.com/

Briefly compare and contrast the rankings produced in questions 2

and 3.

====================================================

======Question 4 is for 3 points extra credit=======

====================================================

4. Compute the Kendall Tau\_b score for both lists (use "b" because there will likely be tie values in the rankings). Report both the Tau value and the "p" value.

See:

http://stackoverflow.com/questions/2557863/measures-of-association-in-r-kendalls-tau-b-and-tau-c

http://en.wikipedia.org/wiki/Kendall\_tau\_rank\_correlation\_coefficient#Tau-b

http://en.wikipedia.org/wiki/Correlation\_and\_dependence