Assignment 2

This is an individual assignment. If you get help from others you must write their names down on your submission and explain how they helped you. If you use external resources you must mention them explicitly. You may use third party libraries but you need to cite them, too.

Date posted: Feb 5th, 2016

Date Due: February 17th, 2016

Goal: Link Analysis and PageRank Implementation

Task1: Obtaining directed web graphs

A. Build your own

Modify your web crawler, re-run it, and build the graph for the 1000 URLs you retrieved (using unfocused crawling) by following their links. You graph should look like follows.

D1 D2 D3 D4

D2 D5 D6

D3 D7 D8

….

Where, D1 is the webpage ***docID*** which is the article title directly extracted from the URL (e.g., Renewable_energy is the docID for https://en.wikipedia.org/wiki/Renewable_energy). Each line indicates the **in-link relationship**, which means that D1 has three in-coming links from D2, D3, and D4 respectively. You only need to build the graph for the 1000 web pages you have crawled, and do not need to consider any other web pages you might come across in the re-crawling process. Suppose we name this graph as WG1. Note: You may as well extract the links from the articles you downloaded if you do not want to re-crawl the pages.

B. Use existing ones

Download the *in-links file* for the *WT2g* collection. *WT2g* is a 2GB crawl of a subset of the web, containing **183,811** web documents. This in-links file is in the format described above, with the destination followed by a list of source documents. This graph is already built and cleaned. We will refer to this graph as WG2.

Task 2: Implementing and running PageRank

A. Implement the PageRank algorithm. PageRank can be computed iteratively by following the pseudocode as below.

```
// P is the set of all pages; |P| = N
// S is the set of sink nodes, i.e., pages that have no out links
// M(p) is the set (without duplicates) of pages that link to page p
// L(q) is the number of out-links (without duplicates) from page q
// d is the PageRank damping/teleportation factor; use d = 0.85 as a fairly typical
value

foreach page p in P
    PR(p) = 1/N                    /* initial value */
while PageRank has not converged do
    sinkPR = 0
    foreach page p in S             /* calculate total sink PR */
        sinkPR += PR(p)
    foreach page p in P
        newPR(p) = (1-d)/N            /* teleportation */
        newPR(p) += d*sinkPR/N      /* spread remaining sink PR evenly */
        foreach page q in M(p)        /* pages pointing to p */
            newPR(p) += d*PR(q)/L(q)       /* add share of PageRank from in-links */
    foreach page p
        PR(p) = newPR(p)
Return and output final PR score.
```

B. Run your iterative version of PageRank algorithm on WG1 and WG2 respectively until their PageRank values "converge". To test for convergence, calculate the *perplexity* of the PageRank distribution, where perplexity is simply 2 raised to the (Shannon) *entropy* of the PageRank distribution, i.e., 2^H (PR).

$$Perplexity = 2^{H(PR)}$$

$$H(PR) = -\sum_{i=1}^{N} P(x_i) \log_2 P(x_i)$$

Where, x_i denotes one web page, N is the total number of pages in the corpus, and $P(x_i)$ is the PageRank score for page x_i.

Perplexity is a measure of how "skewed" a distribution is: the more "skewed" (i.e., less uniform) a distribution is, the lower its perplexity. Informally, you can think of perplexity as measuring the number of elements that have a "reasonably large" probability weight; technically, the perplexity of a distribution with entropy h is the number of elements n such that a uniform distribution over n elements would also have entropy h. (Hence, both distributions would be equally "unpredictable").

PageRank can be considered as converged if the change in perplexity is less than 1 for at least four consecutive iterations.

C*: You can firstly test your PageRank algorithm on the following small graph:

```
A D E F
B A F
C A B D
D B C
E B C D F
F A B D
```

And the final ranking list would be: A>E>(F,C)>B>D (F and C have the same PageRank value)

Task 3. Analysis and Comparison

A. Compare with in-link counts

Sort the pages in WG1 and WG2 each by their in-link counts (from highest to lowest in-link count), and output the Top 50 pages for each graph. Applying ***Kendall Tau rank correlation coefficient*** to evaluate the difference between two ranking lists.

Kendall Tau rank correlation coefficient can be computed as follows:

$$\tau = P(C) - P(D) = (C - D)/N$$

where N is the total number of pairs for comparing items (in our task, you can find the items as the shared common pages in two Top 50 ranking lists. For better measurement, we should compare the entire ranking lists, but for efficiency concern, we just compare the Top 50); C is the number of concordant pairs, i.e., the number of pairs for which their relative ordering is preserved in the two lists; D is the number of discordant pairs, i.e., number of pairs for which their relative ordering is reversed in the two lists; The greater the Kendall Tau value, the more consistent (agreement) the two ranking lists.

B. Effect of teleportation factor

In Task B, we set the damping/teleportation factor to be 0.85. Now, change this factor to be 0.95. Re-run the PageRank algorithm on WG1 and WG2, return the Top 50 pages in terms of the PageRank score, and compute the Kendall Tau between two pairs of ranking lists for each individual graph: 0.85 vs 0.95 and 0.95 vs. in-link counts.

C. Qualitative Analysis

Examine the *Top 5* pages by PageRank and the *Top 5* by in-link counts for WG1 and WG2. For WG1, you can check the original web page using its URL, for WG2, in order to check the original web page, you can get access via the Lemur web interface to the collection by using the "e=docID" option with database "d=0", which is the index of the WT2g collection. For example, the link

http://fiji4.ccs.neu.edu/~zerg/lemurcgi_IRclass/lemur.cgi?d=0&e=WT04-B22-268

or

http://karachi.ccs.neu.edu/~zerg/lemurcgi_IRclass/lemur.cgi?d=0&e=WT04-B22-268

will bring up document WT04-B22-268, which is an article on the Comprehensive Test Ban Treaty.

You are expected to speculate on why these web pages have high PageRank values, i.e., do these pages have high in-link counts? Does any one of its in-coming links also have a high PageRank score? Are all of these web pages the ones that users would likely want to see in response to an appropriate query? Give one possible good query and one bad query. For those that are not "interesting" web pages, why might they have high PageRank values? How do the pages with high PageRank compare to the pages with many in-links? In short, give a brief analysis of the PageRank results you obtained by addressing the questions above.

What to hand in:

1) The source code of your PageRank algorithm implementation.
2) A README.txt file for instructions on how to compile and run your code.
3) In Task 1, the graph file you generated for WG1 (in text file);
4) In Task 1, a brief report on simple statistics over WG1 and WG2 including for both graphs: (1) the proportion of pages with no in-links (sources); (2) the proportion of pages with no out-links (sinks); and (3) draw a log-log scale curve for the number of web pages (the X-axis) vs. the number of in-link count (the Y-axis) indicating the number of web pages having that number of in-link count. Does it follow the power-law distribution?
5) In Task 2, a file listing perplexity values you obtain in each round until convergence for WG1;
6) In Task 2, a file listing perplexity values you obtain in each round until convergence for WG2;
7) In Task 2, sort the pages in WG1 in terms of the final PageRank score. Report the Top 50 pages by their docID and PageRank score;

8) In Task 2, sort the pages in WG2 in terms of the final PageRank score. Report the Top 50 pages by their docID and PageRank score;
9) In Task 3-A, sort the pages in WG1 in terms of their in-link counts. Report the Top 50 pages by their docID and in-link counts; Report the Kendall Tau value you computed for PageRank ranking list vs. in-link counts ranking list. A very brief analysis on the Kendall Tau value.
10) In Task 3-A, sort the pages in WG2 in terms of their in-link counts. Report the Top 50 pages by their docID and in-link counts; Report the Kendall Tau value you computed for PageRank ranking list vs. in-link counts ranking list. A very brief analysis on the Kendall Tau value.
11) In Task 3-B, report the Kendall Tau values you computed for two pairs of ranking lists for each graph: 0.85 vs 0.95 and 0.95 vs. in-link counts. A very brief analysis on the Kendall Tau values.
12) In Task 3-C, your write-ups in 3-5 sentences for qualitative analysis on the Top 5 ranking results for WG1 and WG2.