Programming Assignment 3- Information Retrieval (CS F469)
Deadline: Nov 15, 2019 12:00 PM, Max Marks: 45

This assignment covers the Web Mining modules- Web Characteristics, Web Crawling, and Link Analysis models. The assignment is comprised of four tasks.
Students are allowed to use **nltk**, **spacy**, **numpy**, **sklearn**, **scipy**, and **networkx** libraries. No other libraries will be allowed. All the tasks in the assignment are basic IR and link analysis tasks and do not require any advanced libraries.

Task 1 [**8 Points**] **Web Crawler:** Implement a web crawler for World Wide Web. Follow the steps given below:

(a) Take a set of five random seed URLs (either from manual search, trending pages, or randomly selected URLs).

(b) [**2 Points**] Fetch the webpage using URL based libraries (feel free to refer to the tutorial or other libraries).

(c) [**2 Points**] Parse the webpage/document content (text and hyperlinks). Ignore the text but save the hyperlinks, Id of the webpage, the URLs, and the directed links. Example: Page A has four hyperlinks to Page B, C, D and A. Store all external URLs in the URL frontier queue. Store link information (source $\rightarrow$ target) $A \rightarrow B$, $A \rightarrow C$, $A \rightarrow D$, and $A \rightarrow A$

(d) [**2 Points**] Setup a Mercator URL frontier and keep waiting time as 2 seconds for the same domain. Implement a web crawler with back queues and min-heap. You will not require a front queue setup since all websites/domains are given the equal priority.

(e) [**1 Point**] Discard a URL if you have already fetched the content for it.

(f) [**1 Point**] Keep collecting the data until you have collected 100 documents in the corpus or there are no documents to fetch (whichever is minimum).

Upload the network file (source $\rightarrow$ target) file on moodle while submitting your assignment.

Task 2 **Nearly Duplicate Detection:** Follow the below steps to find the nearly duplicate documents in the collection (Task 1):

(a) [**2 + 2 Points**] Consider each document as a set of terms (duplicate terms are considered only once). Compute the Jaccard similarity between all the document pairs. You can take boolean vector representation as well.

  i. Here, you will get a N x N matrix where N is the number of documents in the corpora. Plot a diagonal heatmap showing the similarity score of all document pairs. You can refer to the seaborn library sample code here: `https://seaborn.pydata.org/examples/many_pairwise_correlations.html`

(b) Consider each document as a boolean vector of terms where 1 represents the presence of a term in the document and 0 represents the absence of a term. Compute the Signature similarity between all document pairs. Follow the below steps:

  i. Consider rows as unique terms and columns as the document vectors.

  ii. [**5 Points**] Apply 20 permutations on the rows and for each permutation find the signature of the documents. Row ID (after permutation) is the signature value.

  iii. [**2 Points**] Apply Jaccard similarity on the signatures.

  iv. [**1 Points**] Similar to previous sub-task, here, you will get a N x N matrix where N represents the number of documents. Plot a diagonal heatmap showing the similarity score of all document pairs.

Task 3 **Link Analysis:** Take the source $\rightarrow$ target data collected in Task 1.

(a) **[2 + 1 Points]** Rank and plot the webpages according to their PageRank. Where PageRank of a webpage is the number of incoming hyperlinks to that page.

(b) **[5 Points]** Compute the PageRank of all webpages using iterative PageRank method (equation system) without teleporting. Number of iterations = 3

(c) **[2 Points]** Discard all the nodes with no out-going edges and re-compute the PageRank of webpages using iterative method. Number of iterations = 3

(d) **[2 + 2 Points]** Compare the results of Task 3(b) and 3(c) with the help of a line chart. Justify the difference in the ranks/scores (for significant changes).

(e) **[5 Points]** Implement matrix multiplication method to compute the PageRank for original network for three iterations.

(f) **[2 Points]** Remove all the nodes with no outgoing edges and re-implement the matrix multiplication method for the remaining network.

(g) **[2 + 2 Points]** Compare the results of Task 3(e) and 3(f) with the help of a line chart. Justify the difference in the ranks/scores.

Do not submit assignments over email.