Unix Tools
Courant Institute of Mathematical Sciences
Project
Due: April 30, 2008

The objective of this project is to design and implement a web search engine. Your programs should be written as a series of CGI scripts. The instructions for using CGI scripts on i5 are given under: http://i5.nyu.edu.

0. Cgi-bin. Familiarize yourself with writing CGI scripts under i5.

1. Crawl. Use wget to crawl and download 1,000 web pages with no more than 25 per site. Copy the files downloaded to

i5.nyu.edu:~mm3572/search/files

using as filenames userid_number, where userid is your userid on the system and number a number from 1 to 1,000. Create a tab-separated masterfile mm3572/search/userid_master.txt with the following format: filename URL PR, where filename is the filename with the format just described, URL is the URL associated to the content of filename and PR is the pagerank associated to that URL.

To obtain the pagerank use the following script:

i5.nyu.edu:~mm3572/search/getpr

which can be used with the syntax: getpr "URL".

2. Index. This section will help you create an index for $U$ the set of all pages downloaded by yourself and your classmates.

Create the list $D$ of all the words appearing in $U$, where words are defined by space or tab-separated strings with all punctuation removed, all turned into lower case. No HTML mark-up or other mark-ups should be indexed. Do not include words such as “a” that appear in more than 100 files.

Use $D$ to create an text file index mapping words to filenames.

3. Search. Write a search script that takes as input a regular expression and returns the list the of pages matching that expression as well as the number of matches, using the index previously constructed. Upper case letters should be replaced by lower case ones.
4. **Ranking.** Write a script that ranks a list of files according to the following criteria (the first two must be used, the others are for bonus points):

   (a) the number of matches.
   (b) the page’s pagerank.
   (c) the distance between the terms in the case of simple regular expressions.
   (d) the formatting of the terms: terms appearing in the title of the page, headers, in special fonts.

   Additionally, allow for quotes around search terms (in simple regular expressions) to retrieve documents that contain exactly the query string.

5. **Snippets.** Write a script that generates a snippet given a page and a query. A snippet is a short text sample surrounding the query matches in the document. For multiple occurrences, you may select the first occurrence or use other more relevant criteria of your choice.

6. **Rendering.** Create a simple query page for submitting a query. Use previous scripts and a CGI-bin to display for a given query the top 10 document URLs and their corresponding snippets.

7. **Other bonus features.**
   
   - Enhance search quality by displaying a document as matching the query when an anchor text in $U$ matching the query points to that document.
   - Enhance query capabilities with query expansion: allow for plural words to match singular ones, try to use morphological information, use synonyms (e.g., for proper names Mike = Michael).