**CAP6777 Web Mining**

Homework 1: Web crawler [15 pts, Due: 05/31/2016]

**Part 1:** Question and Answers [0.75 pt/each].

1. What is HTTP? What is HTML? What are their relationships, and how are they related to web mining?
2. What is a web crawler? What is the taxonomy of the web crawler? Please list major components of a web crawler.
3. To develop a large scale universal crawler, the two major issues include “performance” and “policy”, please identify at least three components on how to improve the performance and how to design good policy, respectively.
4. What is a spider trap (or crawler trap) in web crawling? Please suggest at least three approaches (heuristics) for a web crawler to identify a spider trap.
5. What is Robots Exclusion Standard? Please use a real-world robots.txt example (e.g., <http://www.google.com/robots.txt>) to explain the key fields (allow, disallow, sitemap etc.) and the setting of the robots exclusion protocol.
6. What is Cloaking and Spamdexing? Please explain how they are used to disguise the webservers and affect a search engine’s ranking.
7. What is “frontier” in the web crawler? Please suggest two approaches to manage the frontier, and explain the strength and possible weakness of these approaches.

**Part 2:** Simple web crawling practice [6 pts]

The attached java file (Webcrawler.java) includes a simple web crawler, which starts from a user provided URL (specified by the 1st parameter) to crawl and download web pages, and save the content to a local file (specified by the 2nd parameter). Users can also specify the maximum number of web pages N to be download (3rd parameter). If ignored, the default value is N=20.

**Tasks:**

1. Please download the java file, and use Eclipse or Netbean to build a web crawling project. Please provide a seed url and collect at least 50 web pages from the web crawler [1 pt]
   1. Please submit the original seed URL, and also include all web pages (in one file).
2. In the webcrawler.java, the crawler uses a FIFO queue and a Breadth-First-Search strategy to download web pages. Please change the preference of the crawler (i.e., modify the policy), to collect webpages related to a specific topic. For example, you can modify the program to only crawl web pages related to special topics, such as “biology”, “sports”, “information technology” etc. The purpose here is to build a simple preferential crawler, so web crawler will collect documents related to specific topics. [Please note that you must design/implement a heuristic to check whether an URL (i.e., a document) is related to the defined topic or not, before fetching the content of the URL. A simple heuristic is that, if current document is related to the defined topic, all URLs inside the document are related to the defined topic. You cannot fetch the document then discard the irrelevant ones. All fetched documents must be saved to your repository]
   1. Please draw the flow chart (or the pseudo code) of your preferential web crawler design [1 pt], and explain how does your approach/design make web crawling focusing on special topics [1 pt].
   2. Please turn in the revised source code of your program [1 pt]
   3. Given the same set of seed URLs, please use original Webcrawler.java to collect 200 web pages, and also use your new preferential web crawler to collect 200 web pages. For 200 web pages collected from different web crawlers, please check the percentage of the web pages containing the specific keywords, and report the values in your report [1 pt].
   4. Please suggest one additional approach (show your design as a flow chart or a diagram) which may help improve the accuracy of preferential web crawler, so the collected documents are closely related to the topics [1 pt] (No need to implement this design).

**Part 3:** Hands-on Apache Nutch Web Crawler Project Practice [3.75 pts]

Apache Nutch (<http://nutch.apache.org/>) is a powerful and robust open-source web crawler project, which can support large scale (and distributed) web crawling. Apache also has sorl indexing engine (<http://lucene.apache.org/solr/>) which can combine Nutch and Sorl to build a powerful search engine.

In the third part of the homework, you will practice the installation and configuration of Apache Nutch web crawler. You will need to properly install and configure an Apache Nutch web crawler, fetch files from some web servers, and report the results.

Requirement:

1. Please follow the instructions in the “Apache.Nutch.installation.docx” file, install and configure a Nutch web crawler. Please capture three screenshots to show that
   1. Cygwin has been properly installed and running on your computer [0.5 pt]
   2. Notch has been downloaded and configured as showing in the instructions [0.5 pt]
   3. Notch can successfully lunch a web crawler task. [1 pt]
2. Please provide a seed URL (such as [www.amazon.com](http://www.amazon.com), [www.mit.edu](http://www.mit.edu), [www.yelp.com](http://www.yelp.com)) to collect at least 1000 web pages online. Please include the fetched files (the original downloaded files from the web servers) in one file, and submit the file online through blackboard. [1 pt].
3. Please explain the meaning of the Nutch parameters, and change at least three parameters to design three crawling tasks. Please report and explain the crawling results for each parameter setting [0.75 pt].