Project II
CS 410/510: Information Retrieval on the Internet
Project due: Tuesday June 1, 2010
20 points

Goal: The goal of this project is to experience designing and building a special-purpose search engine. The project will encourage you to think about the tradeoffs in various design decisions as well learn in more detail about how the various parts of a search engine fit together.

Details: For this project you may work individually or in groups of two or three students. You are to build a special-purpose search engine that will:

- index the websites of researchers in the database field, and
- accept and respond to queries by returning a ranked list of the homepages of database researchers.

The search engine should use the contents of all the pages in the researchers' website to determine which homepages should be returned, and in what order. Note that what constitutes a website is a somewhat fuzzy notion. You will need to make some design decisions about what links to follow as part of a site, and what types of documents you will index. You should use the contents of at least one other document type besides HTML and plain text (such as PDF, Word, or Powerpoint) but we do not expect that you'll be able to index every bit of information from every site. The search results should contain lists of homepages, not the other pages in a given site. Use of Lucene is recommended but not required.

You will start with the list of researchers from the DBLife site that is found at:
http://dblife.cs.wisc.edu/

Steps you will need to do:

- Crawl the website of each listed researcher, following links in order to index pages that have information about the researcher, such as publications, projects, students, courses, etc.
- Extract the content from the crawled pages. As noted above, you should extract text from at least once document type other than HTML and plain text files.
- Index the content.
- Somehow link the indexed content of each page to the homepage of the site where you found it.
- Determine how to use the content of the various linked pages to determine similarity to the query so that you can return useful ranked output.
- Develop a simple user interface that accepts queries and returns output. The UI does not need to be fancy. Remember that the output should return a list of homepages, not the "lower level" pages on the researchers' sites.

You will need to submit a written description of your project. Your write up should include:

- A description of the design of your search engine. The description should include a discussion of the design decisions you made, alternatives you considered, and reasons you made the choices you made.
- A description of how you implemented the system. The description should include the implementation decisions you made, alternatives, and reasons. For example, did you use

Nutch, write your own crawler, or use some other tools? Did you use an existing tool to extract text from PDF documents? What components did you build from scratch, if any?

- A description of any major problems, issues, lessons learned, iterative development cycles, or other interesting aspects of the project.
- A proposal of how you might evaluate your search engine. You don't have to perform the evaluation but you should outline what kinds of things you might evaluate and how you could do it. You may include comparing your search engine to those of your classmates, but your discussion should not be limited to just such a comparison.

Deliverables:

1. The code for your project. If you use Lucene, Nutch, or any other libraries, make sure you program against the API and write your own subclasses as needed. Don't change the existing libraries.
2. Your project write-up as described above.
3. The results (show at least the top ten) that your search engine returns in response to the following queries:
    a. *David Maier*
    b. *query optimization*
    c. *superimposed information*
    d. *data provenance*
    e. *Semantics and Evaluation Techniques for Window Aggregates in Data Streams*
    f. *database researcher*
    g. *computational geometry*

If you choose to do the project as a group, be sure everything you turn in has each group member identified.