CS410 Project Proposal

Team Members

Tyler Davis (tadavis2@illinois.edu)
Kee Dong (yuqingd2@illinois.edu)
Mukund Madhusudan (mukundm2@illinois.edu)

What is your free topic? Please give a detailed description. What is the task? Why is it important or interesting? What is your planned approach? What tools, systems or datasets are involved? What is the expected outcome? How are you going to evaluate your work?

We propose a web application that enables a user to bookmark a document by entering a URL or a set of URLs (and a tag) to be indexed by the system into an inverted index for later full-text search and retrieval. Users can then enter a tag and a query and review the returned list of relevant documents and click through to the original URL. This tool would enable us to explore the implementation of an inverted index as well as test the functionality of various retrieval functions in a real-world context. Additionally, this tool would be useful to a user with research needs for compiling web documents for later review without having to document their contents manually.

We plan to develop this tool as a web application that processes a documents' contents into an inverted index stored in a database. We will use the Django web server framework to implement this project and a MySQL database to store the inverted index. Asynchronous methods such as indexing will be implemented using the Celery library. We will evaluate our work by testing our system with a suite of tests against expected retrieval/indexing results.

Which programming language do you plan to use?

We will use the Python programming language and MySQL to implement the backend of this project. The frontend will include HTML/CSS and some JavaScript components.

Please justify that the workload of your topic is at least 20*N hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.

1. Architecture planning & data modeling (4 hours)
2. Dynamic web scraping module (10-15 hours)
    a. Ensuring the scraper can dynamically handle most URLs it is given will require some extra work compared to designing a scraper to handle a single page or a static set of pages.
3. Writing tests (4 hours)
    a. Each person should be responsible for writing tests for the module(s) they implement.
4. Front-end (10 hours)

5. Index module (creating the inverted index) (15 hours)
    a. None of us have any practical experience in constructing an inverted index, so this may take a more substantial amount of time compared to other web app components that we may previously have been exposed to in CS410 machine problems.
6. Retrieval module (15 hours)
    a. Similarly we don't have practical experience in text retrieval from an inverted index, which may add some additional complexity to the text retrieval module that needs to be accounted for in our estimate.
7. QA (10 hours)

Total: 68-73 hours