Theme 1: Intelligent Browsing

Description

Most people who browse the Internet do so through popular browsers, such as Google Chrome, Mozilla Firefox, or Safari. Indeed, we complete many of our tasks by using a browser, thus how well those browsers support our tasks may significantly affect our productivity. The current browsers are great for quickly finding or saving information, but they lack "intelligence". Adding more intelligence to such browsers can turn a browser into a personal intelligent assistant that can broadly impact many users, potentially transforming how they access information on the entire Web.

The goal of this track is for groups to build on top of existing browsers using the information retrieval techniques learned in this course. One straightforward way to extend browser functionality is to develop browser extensions (primarily written in JavaScript). Some extension-based project examples include:

1. Index the current page and allow users to search over the page using a common retrieval function, such as BM25 (the current search capabilities are limited to exact keyword match)
2. Scrape and index Campuswire pages and Coursera pages in order to link questions to content, and vice versa (you can think about how in general you might leverage a browser extension to help linking the scattered educational content such as Coursera lecture videos, textbooks, and relevant discussion on the Web)
3. Create a collective bookmarking, question-answering, or annotation system among specified groups of users
    a. As an example, see Fermat's Library

The above examples are meant to illustrate the problem domain; groups are free to propose a topic within this track that isn't in the above list, especially if it solves a well-known problem or shortcoming of current browser systems. Students are also encouraged to coordinate group work: e.g. one group could focus on the front-end design and another group could focus on the back-end server. This coordination would allow groups to collectively solve problems beyond the scope of a single group.

The following links may be helpful to get started:

1. Chrome extensions
2. Mozilla Extensions
3. Safari Extensions

Requirements

If you choose this theme, please answer the following questions in your proposal:

1. **What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.**

- David Ho - davidsh3
- Ben - bhyang2
- Nicholas - ntruong3 (Captain)
- Jun - jmzhong2

2. **What topic have you chosen? Why is it a problem? How does it relate to the theme and to the class?**

- Intelligent browsing system that takes topic keywords as input, scrapes web for relevant documents and generates inverted index of most frequent relevant words.
- In research settings where a new project is embarked upon, a researcher might only have a general knowledge of a topic and is familiar with only a limited scope of keywords. Currently, such researchers would query upon keywords he is familiar with in order to browse and pull less-familiar keywords related to the research project. The researcher would then combine the familiar and less-familiar keywords to create more effective queries.
  This intelligent browsing project seeks to generate statistical visualizations relevant to a limited-keyword query, in order to help the researcher more quickly and more easily discover keywords that would help generate an effective query. The intelligent browsing program would take in some known keywords as input, scrape all docs and create an inverted index of the most frequent relevant words that appear in the docs resulting from the input query, and then generate statistical visualizations of those most frequent relevant words.

3. **Briefly describe any datasets, algorithms or techniques you plan to use**

- MeTa
- Okapi BM25
-

4. **How will you demonstrate that your approach will work as expected?**

- Demo video, screenshots
-

5. **Which programming language do you plan to use?**

- Javascript (Chrome extension)
- Python

6. **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.**

- Chrome Extension
    - 40 hours
- Web Scraper
    - 20 hours
- Inverted Index
    - 20 hours

At the final stage of your project, you need to deliver the following:

- Your documented source code.
- A demo that shows your implementation actually works. If you are improving a function, compare your results to the previously available function. If your implementation works better, show it off. If not, discuss why.