CS410 - BM25 Chrome Extension - Project Progress Report

Sean Enright (seanre2)
Bhaveshkumar Manivannan (bm12)
Matthew Mechtly (mechtly2)

1) Which tasks have been completed?

- (~20 hr) Two out of the three members of the team have never written a single line of Javascript or done any work with browser extensions. Accordingly, significant time must be invested by these members to learn this language.
    - **Completed**
        - The learning curve was rather steep--especially for learning browser extension architecture, but both members in question are now reasonably competent with JavaScript.

- (~10 hr) The other part of the back-end will actually be calculating the BM25 score for each document and doing so in a quick manner at runtime.
    - **Completed and Implemented**
        - BM25 has been implemented entirely in Javascript, and it seems to work correctly.

- (~10 hr) Time will likely be needed to optimize both the BM25 and document-parsing algorithms to fully take advantage of the limited browser resources.
    - **Completed and Implemented**
        - The algorithm runs practically instantly for any typical or reasonably sized query (<15 words)

2) Which tasks are pending?

- (~15 hr) We must build a front-end interface with which the users can interact, input keywords, cycle through the various pseudo-documents, and adjust the parameters of BM25.
    - **Already Completed and Implemented:**
        - There exists a field in which the user inputs text, as well as a search button to kick off the BM25-calculating engine.
        - In order to enable the user to cycle back and forth through the most-relevant pseudo-documents, “Next” and “Previous” buttons have been provided and implemented correctly.
    - **In Progress**

        - While not technically part of our promised deliverable, a keyboard shortcut to automate the opening of the Chrome extension will be enabled in order to improve usability.
        - To allow more flexibility in the BM25 algorithm itself, A user-input field for the ‘b’ and ‘k_1’ parameters will be integrated.

- (~15 hr) Extending from the front-end interface, the back-end that actually performs the text retrieval calculations must be constructed as well. A significant portion of this back-end development will involve the creation of a robust DOM-parsing engine, capable of dividing the web pages into appropriate pseudo-documents.
    - **Already Completed and Implemented**
        - Parsing Engine for the range of HTML DOMs is completed and implemented.
    - **Pending**
        - Need to develop an independent parsing engine for PDFs and integrate that with the HTML parsing engine code already implemented.

- (~10 hr) Refinement / Testing of the extension based on feedback from tests on a broad range of web pages.
    - **Pending**
        - We need to verify that our implementation of BM25 in Javascript is correct; therefore, we must perform validation of our algorithm by evaluating test documents.
        - To ensure that this extension also runs on systems with low computing resources, we will perform testing of our extension on such a system.

3) Are you facing any challenges?

- Currently, our biggest challenge is ensuring that our document parser also functions with PDFs, given that they are in entirely different formats by definition than HTMLs.
- In order to implement the flexibility of users being able to input different 'b’ and ‘k_1’ values, we need to integrate multiple, independent event listeners within the content.js script. This is proving surprisingly difficult.
- Our final challenge is a bit more philosophical and marketing-related. Once BM25 is implemented correctly, we need to evaluate what the strongest use cases are while browsing: casual browsing or long-document searching research?