Intelligent Browsing: Text Retrieval for Long Webpages

Team Discovery Channel - Project Proposal CS410

Team Members:

- Sean Enright; seanre2@illinois.edu
- Bhavesh Manivannan; bm12@illinois.edu
- Matthew Mechtly; mechtly2@illinois.edu (project coordinator)

As a team, we have selected “Theme 1: Intelligent Browsing” for our CS410 Course Project. Within this very broad theme, we will create a chrome browser extension that performs text retrieval on a given webpage. While this problem sounds similar to the problem that is often solved by simply executing a “find-in-page” search, it is distinct.

Often, finding the most relevant part of a lengthy web page is infeasible. For example, pressing “ctrl + f” and typing in a keyword isn’t particularly helpful when several hundred results are returned. Additionally, if we want to find certain parts of a long web page where a particular term coincides with a separate term, this cannot be done with the built in “find-in-page” search functionality. Therefore, we will create a browser extension that allows users to type in ‘n’ different keywords and returns the most relevant <div> within the webpage. This relates to the theme of the class since we will be developing a user-querying tool that facilitates faster text retrieval.

No datasets will be required to complete this assignment. However, we will be employing BM25 to search through the various pseudo-documents (where each pseudo-document is a subsection of the web page on which the browser extension is run) and return the most relevant pseudo-document. After the most relevant pseudo-document is found, the user will be transported there, after which the user can cycle to the next most relevant pseudo-document.

To demonstrate that our approach will work as expected, we will demonstrate the browser extension’s ability to identify the most relevant pseudo documents on lengthy web pages already out in the wild of the internet.

For the application we want to tackle, we don’t have much choice of language: we will write the browser extension in Javascript exclusively.

Anticipated Workload:

(~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.

(~15 hr) We must build a front-end with which the users can interact, input keywords, cycle through the various pseudo-documents, and adjust the parameters of BM25.

(~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.

(~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.

(~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.

(~10 hr) Refinement of the extension based on feedback from tests on a broad range of web pages.