Final Project Documentation

Project Presentation:
https://www.youtube.com/watch?v=wXeqGFaLIO8

Project Overview

The goal of this project was to create a chrome extension that allows someone to use the BM25 ranking algorithm to search within a webpage. Searching in chrome (via Command-F or Control-F) works via key word matching. Thus, if you have a more general search query you would like to use to find a particular paragraph or sentence within a specific webpage, you're out of luck. My chrome extension solves this issue by splitting up the text on each page into documents and then running the BM25 retrieval function over this corpus to retrieve the documents that best match your search query.

Implementation

At a high-level, the chrome extension works by getting a user query and collecting the text on the current webpage and then sending this data to a Google Cloud Function that runs the BM25 retrieval function over this data and returns the top 5 results. The Google Cloud Function is implemented in python and relies on the rank-bm25 python package for the search algorithm implementation. Once the top 5 results are returned to the client, a user can cycle through the results.

There are 4 main components used to implement my chrome extension

- The extension manifest file (manifest.json): Specifies important metadata for the chrome extension like: starting action, content scripts, host permissions (for invoking the Google Cloud Function), and permissions (activeTab, scripting, and storage).
- The Popup page/script (popup.html and popup.js): This is how the user interacts with the chrome extension. It's the row of the popup page and script to fetch user input, call the Google Cloud Function, and handle user commands like going to the next search result.
- The Content script (content.js): The content script allows the extension to interact with the DOM of the user's current webpage. It has two responsibilities:
    - Fetch the document text for the popup script on command.
    - Highlight and snap the webpage to the search result provided by the popup script.
- The Google Cloud Function: The Google Cloud Function handles requests from the popup script to calculate the top 5 search results according to the BM25 retrieval strategy. It starts by parsing the text into a corpus of individual documents delineated by new line characters. Next, it tokenizes each document and the search query and normalizes the text case. Lastly, it imports and runs the BM25 algorithm before using calculating and returning the top 5 search results. There are three files relevant to the Google Cloud Function:

    - main.py: Source code for the Google Cloud Function
    - requirements.txt: Specifies dependencies (only the rank_bm25 package)
    - deploy.sh: Script used to deploy the Google Cloud Function

The overall file structure is as follows:

- cloud (files relevant to the Google Cloud Function)
    - src (source code for Google Cloud Function)
        - **main.py**
        - **requirements.txt**
    - **deploy.sh**
- src (files relevant to the chrome extension itself)
    - scripts
        - **popup.js**
        - **content.js**
    - **manifest.json**
    - **popup.html**

For more detail regarding precisely what functions are in each file and what each file does, please refer to the documentation in the files themselves. popup.js, content.js, and main.py are all commented with relevant information and details.

There are a number of ways this extension could be improved. I think the three top improvements would be:

- Better UI: The current UI is very bare-bones, and a better-looking and more responsive UI would improve the extension
- More search options. The current search is limited to only returning the top 5 results. It would be great if users could specify how many results they wanted. Additionally, documents are always delineated via new line characters. It might be helpful if users could customize what a document was (sentence, paragraph, page, etc.)
- Improved performance: every time a document is queries the entirety of its text is sent to the Google Cloud Function and reindexed. It would be great if we could avoid indexing the same document multiple times

Setup and Usage

Setting up and using the chrome extension is very simple:

1. Download the repository
2. Open chrome and click on extensions -> manage extension
3. Click on “Load unpacked extension” then select the top-level “src” folder (the one containing the manifest.json file)
4. Navigate to the webpage you want to search in
5. Click on extensions and select the “BM25 Doc-Search” extension
6. Enter your search query in the box on the pop-up window then click submit
7. Wait for the “prev” and “next” buttons to become enabled
8. Click on the “next” button to begin cycling through the returned results

Note that this extension relies on a Google Cloud Function that I've deployed. I'll keep it deployed for another few weeks until grading is over but after that I plan on disabling the it. Once it is disabled, if you still want to try out the extension, you can! You just need to deploy your own Google Cloud Function:

- Create a Google Cloud Account
- install and setup the gcloud CLI: https://cloud.google.com/sdk/docs/install
- update the name of your Google Cloud Function in deploy.sh
- Run the deploy.sh script to deploy the cloud function
- Update the URLs in manifest.json and popup.js to match the url of your new Google Cloud function

Contribution:

I worked alone on this project, so all the contributions were by me. While the total number of lines of code in the project is relatively small, it was still a large effort for me since I have never worked with Google Cloud Functions or Chrome Extensions before. So, a large portion of this project was spent learning these technologies from scratch and troubleshooting/debugging.