Project Progress Report

Please upload your progress report to the Github repo shared on CMT. The progress report should give us an idea of how you're implementing your proposal. It should answer 3 main questions: 1) Which tasks have been completed? 2) Which tasks are pending? 3) Are you facing any challenges?

Completed Tasks:
Frontend (Mostly DOM interacting and UI):

- Create a minimal working example of chrome extension
- Obtain Query and Results from the https://google.com/search*, and store them.
- Obtained the query information above when the link in the search page is accessed.
- Obtain text data from the result page DOM, set up an interface for TR model.
- Render result from text retrieval (currently a simple placeholder)

Backend (Text retrieval processing):

- Basic web scraper that pulls links from a google search and grabs words from the links
- Basic implementation of BM25 using an existing library

Algorithm:

- A basic framework of ranking text that allows us to add in more features and text retrieval algorithms in the future.
- Naive BM25 method that can return which parts in the given document are most relevant to our target query.

Pending Tasks:
Frontend (Mostly DOM interacting and UI):

- Replace all placeholder with native-messaging between backend text retrieval model
- Make the UI more aesthetic and practical (such as highlighting result)
- Make User able to navigate through result via keyboard up/down
- Make a Search Box which take custom query instead the google search query
- Make a Settings page where user can control search parameters/threshold (optional)

Backend (Text retrieval processing):

- Improving the text retrieval methods (see Challenges)
- Improving the web scraper to pull more accurate text data
- Running the algorithm on the aforementioned text data
- Combining the components together into final application

Algorithm:

- Explore better ranking algorithms. We are planning to include a distilled version of the BERT in our product.

Challenges:
As none of us have built a chrome extension before, we spent a lot of time learning Document Object Model, Chrome Extension Structure and Manifest V3. Also, we found out interacting with

the DOM element, connecting between background, active tab and newly-created tab is way more ambiguous and time consuming than we expected.

We are also considering how to run our Python code for scraping web pages and running our text analysis algorithm. However, we don’t believe that there are straightforward methods of running python scripts dependent on libraries in browser extensions, so it is likely that we will end up using some form of native messaging.

Meanwhile, although we have already built a basic BM25 model, to match elements in the web page seems to be more ambiguous than we expected and much harder to achieve an optimal result. Processing the DOM elements is tricky and it is really hard to determine the threshold of relative terms as different users might have different expectations. Those tasks won’t be super hard but are expected to be time-consuming.