Team Diet Coke

Proposal

Group leader:
suxiang2@illinois.edu
Team members:
Suxian Hang: suxiang2@illinois.edu
Joshua Wu: jgwu3@illionis.edu
Juntong Song: juntong7@illinois.edu

What topic has our group chosen:

Better Search box Chrome Extension (Intelligent Browsing)

Why is it a problem:

Google's search engine is extremely robust and powerful, but not without flaws. Google regularly changes how it's search engine works, which can cause some frustrations for users. The abstracts are very short, and aren't always as relevant as they could be. Our group noticed that there's still a gap between utilizing search engines and actually gaining the relevant information.

After the search engine prompts out the result page, it actually only provides abstracts of many webpages which potentially could provide useful information to the users. Many times, the users still have to access these websites and keep searching for the information they want manually. Unfortunately, such a process could be very painful sometimes.

- ## Solution:

  Chrome provides a built-in search box function where users can search for text in the webpage. However this function is limited to exact keyword match and cannot perform any relevance search. Therefore, we can build a chrome extension to replace the built-in search box where we will use BM25 or similar approach to perform relevance search and will highlight the result in different colors based on their likelihood.

  By this approach, users can use this extension as an upgraded version of Chrome built-in search box. What's more, when users access a website via google search we will automatically record their query. Next, the extension will use the query to find the relevant texts in the web page and notify them to users, which will address the problem posted above.

How does it relate to the theme and to the class:

The main idea of our project is relevance search, which is a classic text retrieval problem and has been covered a lot in this class. We will be utilizing relevant methods that are covered in class, like BM25 or something similar, as well as providing feedback to our system through explicit feedback.

Team Diet Coke

What kind of datasets, algorithms, or techniques we plan to use:

We plan to consider a variety of techniques for our relevance searches. We will most likely use a ranking function in the form of BM25 or some variant of it for our relevance calculations, and we can use explicit feedback from us (the user in this case) in order to provide feedback to the ranking function. We will use this feedback to adjust the parameters of the function in order to provide better results.

Justification to the feasibility of our plan. What will be the expected outcome:

Our team members are very confident in the success of this project. Here are the reasons:

- **Robust and usable system:**
  There are many text retrieval algorithms that have been thoroughly tested and proved robust and effective. And our task has high similarities with the one that these text retrieval algorithms are dealing with.

- **High accuracy expected:**
  Finding relevent informations fragment in an website is actually a simpler version of information retrieval problem. Before the information is passed in and processed by our extension, the user and search engine actually has filtered out a lot of noise. This means most of the time, the extension will be working on a relatively clean dataset. Many possible challenges in the text retrieval area will not be the case in this project.

- **User tolerable response time:**
  Although, people will have higher expectations on the response time of our product, considering the difference of data volume between what regular text retrieval system and our system need to handle, efficiency would be a solvable problem in team's time limit.

Programming language plan to use:

For Chrome extension: React (HTML+CSS+JS)
For Rest API interacted by the Chrome extension: Node.js

Workload justification (20 * N hours):

We have three group members for an expected workload of 60 hours. The task is very robust, and will involve developing an application using React. For every subtask we will work in a new branch and merge to master once finished. Also, we will record our tasks and time spent using trello/pull requests. Therefore, by carefully documenting our work, each of our team members can always increase or decrease workload to match the project requirements. We also can continually adjust our algorithms and provide feedback, which is a lengthy process that can take many iterations.