Track: Development

Team Members: Archna Sobti (archnas2), Vasu Chalasani (sc83), Vibhav Adivi (vadivi2), Anish Meka (anishm2)

Coordinator: Anish Meka

[Functions and Users]

Our tool is an LLM-based document evaluation chrome extension, which responds to natural language queries the user has about the document on the screen. The chrome extension's functionality allows users to summarize, identify and analyze relevant text data quickly. The users of our tool are the general population who would like to interact and understand overarching themes about large text data in a quick and effective manner.

[Significance]

Currently, there is no quick way to interact with large amounts of text data on the web, other than having to copy / download the entire page and passing it to tools like ChatGPT. The other alternative is being able to Ctrl+F a document (matches by exact text), but that does not allow users to ask high-level questions of the text (i.e. "pros of photosynthesis" in a document that covers plant life). However, in time sensitive situations or academic environments, it is important that users are able to easily identify the information relevant to them on various sites and do so in a manner that does not place a burden on them to engineer their search (through Ctrl+F).

Understanding large bodies of text at a high-level is important, not just for the average user but for politicians, researchers, lawyers, and other professionals who have little time but lots of data to synthesize and understand. Our tool will allow them to spend more time making decisions and less time consuming text data.

[Approach]

We plan to build off a React chrome extension boilerplate. We will scrape the HTML text from the current page the user is on, and then we will use LLMs and Python libraries to analyze the user's query in relation to the text. Once we are able to pinpoint parts we consider relevant to the user's natural language request, we will highlight those parts of the site (similar to Ctrl+F), and we'll build out this frontend using various libraries.

[Evaluation]

We will test out our chrome extension on a few documents (i.e. news articles, research papers, etc.) of varying length and on various topics (i.e. laws / policies, history, etc.). We

will manually test the accuracy of the responses we get from our backend. If time permits, we would like to conduct user tests as well exposing various students to articles for the first time and seeing how their performance in extracting relevant data to questions we provide compares with and without the usage of our tool.

[Timeline]

- 4/12 - Compile resources, including boilerplate chrome extension code, backend frameworks and frontend libraries
- 4/15 - Test boilerplate code and begin frontend and backend development in parallel
- 4/22 - Complete frontend and backend development
- 4/23 - Connect frontend and backend
- 4/27 - Finish testing

[Task division]

Archna and Vasu plan to set up the boilerplate code. Vasu and Anish plan to work on the backend. Archna and Vibhav plan to work on the frontend. Everyone will work together to evaluate and conduct testing on the chrome extension.