

CS 410 Text Information Systems Final Project Progress Report

Team: Ctrl Freaks

Fall 2023

1. Progress Made Thus Far

Up to this point, most of our time has been spent allocating tasks to group members and researching approaches for completing each of these tasks.

Each member of our group has been assigned a specific task to complete and is responsible for researching approaches for their solution. Then, we review these approaches as a team to ensure that our final solution can be synthesized back together.

Additionally, each team member has researched how to create Chrome extensions and publish them onto the Chrome Web Store.

Further information about the specific progress of each of our tasks is below:

- Visualizing Knowledge Graphs
 - Researched approaches in both Python via the Plotly library as well as in Node.js. Our usage of either Python or Node.js is dependent on how easy frontend-backend integration is; for now, we've devised proof-of-concept graph visualizations for both of these approaches.
- Similarity Algorithm
 - Research on different embedders and python libraries
- Frontend
 - Researched the best tool/library to such as using axios on the frontend to make requests to the backend. Decided to use React.
- Backend
 - Researched best backend to use for Data Manipulation and easy to use with React Frontend. Currently decided on python, with flask to make api endpoints for the frontend.

2. Remaining Tasks

- Visualization of Knowledge Graphs
 - Our preferred method for graphing would be Python because we will likely need it to run certain components such as the similarity algorithm. Additionally, Python is great for data analysis and Plotly in particular has some great features we would like to make use of. To this end, we need to determine the feasibility of frontend-backend integration with Python as well as how to embed our graphs into the UI.
- Similarity Algorithm
 - Research other possible embedders besides using BERT, and testing the other similarity algorithms besides cosine similarity (manhattan, jaccard, etc)
- Backend
 - Work on making a basic backend template and seeing if the backend and frontend are able to connect with no issues.
 - Make sure there are no CORS errors etc.
 - Look for a way to store web scraped data in order to make the knowledge graph that does not involve databases
- Frontend
 - Work on implementing a basic frontend using React and make sure that it can connect to the backend
 - Make sure that the chrome extension is easy to understand and is visually appealing

3. Challenges/Issues

- Visualization of Knowledge Graphs
 - The main issue we're facing right now concerns how to integrate our Python graphs with some JavaScript or Node.js-based frontend. Specifically, how we can embed those graphs onto the UI. Further research needs to be done here investigating the feasibility of this.
- Similarity Algorithm
 - Depending on the size of the dataset, embedding with certain libraries may take a very long time, may need to opt for a more efficient algorithm rather than accuracy in order to complete analyzing all data in time for project submission
- Backend

- The hardest challenge would most likely be the way we store the user's browser history. As we will not be using databases. Other than that any troubleshooting with getting the application working with the frontend.
- Frontend
 - The main challenge is to make sure that the frontend can connect to the backend and accurately show the knowledge graphs made. Also adding css to make the chrome extension look good.