Project Title: FindMyPaper

Project Summary:

This project is a web application to help users find academic papers by performing a vector search on a database of paper abstracts. Users will create an account to save their liked papers.

Additionally, the platform will allow users to vote on the relevance of the papers they receive, enabling continuous fine-tuning of the search algorithm for improved accuracy. The application will also feature a leaderboard displaying the top papers of the day and of all time, highlighting trending research and helping users discover the most popular literature within the specified timeframes. This combination of interactive feedback and dynamic content presentation enhances the overall user experience, making it easier for researchers, students, and professionals to find and engage with relevant academic papers.

Description of the application:

The application we are trying to develop is a comprehensive economic research paper search and recommendation system that focuses on simplifying the process of finding relevant economic publications. Users will be able to search for specific research papers using any keywords or phrases they want. Our goal with this is to make it simple for people to find papers based on their abstracts and crowd-sourced opinions.

In addition to the search feature, we will also integrate a voting system, in which users can upvote papers based on their usefulness or relevance. The system will allow users to locate the most liked papers of all time, ensuring the presence of the most high-quality and impactful publications. To further encourage community engagement, we will showcase a leaderboard that displays the most upvoted papers of the day and of all time, allowing users to seek trending research topics. Our goal here is to make reading papers a more social activity, where people can directly see paper popularity based solely on their community's opinions, rather than mixing in several other factors like politics and awards.

Technically challenging features (creativity):

Our search algorithm is the creative portion of this project. This algorithm comes with several complexities. The first part of the algorithm will be conducting a vector search and finding the most relevant papers using techniques such as cosine similarity to find the papers with the most similar abstracts. After finding the most similar papers by abstract content, we add another feature of consideration into the mix: the paper's popularity on our website. This popularity is determined by the number of users who have marked the paper as useful. We'll apply a weight to both of these factors (similarity and popularity) to determine and return the papers that are most relevant to the user's query. This portion will be technically challenging, as it requires the implementation and amalgamation of several complex algorithms.

Usefulness:
Our application helps users find relevant published economic-related research papers within a short time. It is more useful because our database is more focused on the economic field, and the application only considers the abstracts instead of the whole paper so that the vector search process won't be distracted by other useless and irrelevant information.

This application also provides users with personal designs. Keeping track of what papers they liked allows users to easily come back to their favorite papers without the hassle of having to go back through their file history. The voting tool of our application not only creates a friendly and shared environment among all users but also enhances the search algorithm for better results. Showing the most voted papers helps the community be aware of trending topics and novel ideas, especially since our application only focuses on one specific field of study. The voting strategy improves the search algorithm as more and more users give feedback on the queries and returned papers, making our application more user-friendly and time-efficient.

One of the most successful similar applications is Google Scholar. Our main differences are as follows:

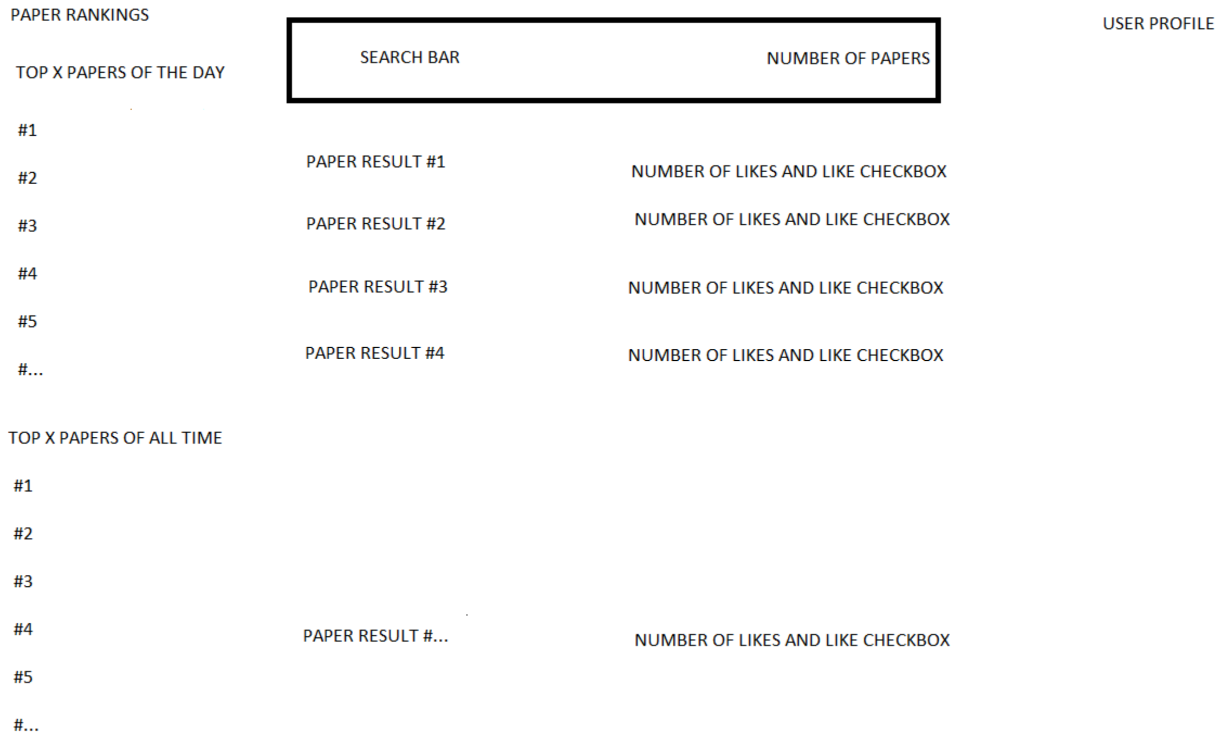
1. Our application provides a special voting implementation which Google Scholar doesn't have. This implementation makes our application more personal and easy to use by suggesting possible relevant papers to users before they do any search.

2. We only focus on economic papers and abstracts, whereas Google Scholar takes everything into account, which makes it possible to include some irrelevant information.

Realness
Our data is web scraped economic papers abstracts across multiple different publishing companies(Springer, Elsevier, Wiley, … etc). It is currently in csv format. It contains the Journal_Website, Journal_Name, Volume_Issue, Title, Authors, Abstract. It has a cardinality of 148,745 and a degree of 6. Having the abstracts and titles allows us to create our search algorithm while the Journal_Name and Journal_Website gives the user more insight as to where the paper is coming from.

Functionality
FindMyPaper has several different features. The user can log in, and manage their account information appropriately in a similar manner to other websites. Once logged in, the user can interact with the website in several ways. Our website allows a user to create a query that can include key words or relevant sentences. The website will then query the database using a vector search algorithm powered by cosine similarity and other NLP techniques to find the N most similar papers (where the user can specify N). Upon receiving the list of papers, the user can then mark papers as helpful or unhelpful. We store a popularity index for each paper, which is determined by the number of users that have queried and voted on the paper being helpful.

This popularity index helps finetune our search algorithm so that we return the most relevant papers and maximize user satisfaction. FindMyPaper also stores a page which contains the most popular papers of the day, and of all time. As a result, curious users can peruse the popular papers and satisfy their thirst for knowledge. Finally, users can query the papers that they have liked in the past, so they can easily come back to their favorite papers.



Division of Labor:

We aim to divide into groups of 2 to split work appropriately, where each group pair programs to complete each task.

Hao and Jason will work on the front end. This includes the User Login section, as well as rendering everything on the website, making sure the user experience is flawless. They'll display the information delivered by the backend appropriately and make the end to end website functional so that it is usable.

Prajwal and Alex will work on the backend. This includes the search algorithm, updating and maintaining our database tables appropriately (managing paper popularity, paper querying, etc), and determining the most popular papers of the day and all time (this includes some database operations, as well). They'll deliver information for the frontend to display appropriately.