GRADUATE CERTIFICATE
INTELLIGENT REASONING SYSTEM (IRS)

---------- **PROJECT REPORT** ----------

"🥭 Mango" - An Academic Knowledge Platform

GROUP 24

Li Jiacheng (A0285823W)

Mao Zhihong (A0285799X)

Goh Min Hua (A0285810A)

Table of Contents

1. EXECUTIVE SUMMARY

As a proud member of our team, I am delighted to present this executive summary, encapsulating the successful development of our academic knowledge platform, "Mango." Mango has emerged as a powerful tool that addresses the academic community's need for fast and efficient paper search and recommendations. Leveraging cutting-edge technologies, including Dense Passage Retrieval for searching and Dual Tower DNN retrieval for recommendation, we have already made substantial progress. Our vision is to further refine and enhance Mango to better serve our users.

Key Achievements:

- **Dense Passage Retrieval （DPR） for Efficient Searching:** Mango utilizes DPR to facilitate rapid and precise paper searching. This technology enables us to efficiently narrow down search results to provide users with the most relevant academic papers.
- **Dual Tower DNN Retrieval for Recommendations:** Our platform employs Dual Tower Deep Neural Networks for personalized paper recommendations, ensuring that users receive tailored suggestions that align with their academic interests and preferences.
- **User Feedback Integration:** To continuously enhance user experience and relevance, we have implemented automated user feedback collection mechanisms. This iterative feedback loop ensures that Mango evolves in response to user needs and preferences.

Future Enhancements:

- **Feature Expansion:** We are committed to augmenting Mango's capabilities with additional features that will enrich the academic research experience. These may include advanced search filters, collaborative research tools, and integration with academic databases.
- **Automated User Feedback:** To streamline the feedback collection process further, we will develop advanced algorithms and mechanisms to capture user insights seamlessly, helping us make data-driven improvements.
- **UI Refinement:** An intuitive and user-friendly interface is pivotal to Mango's success. Our team will focus on polishing the UI to ensure a seamless and delightful user experience.

Conclusion:

Project Mango represents a significant milestone in our mission to revolutionize academic knowledge retrieval. Our use of DPR and Dual Tower DNN retrieval demonstrates our commitment to excellence and innovation.

As we continue to invest in additional features, automated feedback collection, and UI refinement, we are confident that Mango will remain at the forefront of academic knowledge platforms.

By listening to our users and harnessing the power of advanced technologies, we are poised to create a transformative tool that empowers researchers and academics worldwide. Together, we are shaping the future of academic knowledge discovery with Mango.

2. INTRODUCTION

2.1. Business Problem Background

As the use and reliance of the Internet gets increasingly prolific, there is a need for search engines to be more effective to reach the demands of users. Students ranging from bachelor's to PhDs, researchers and even teachers have struggled to search for the relevant information they need. While there are existing search engines like Google Scholar, Semantic Scholar out there to aid in literature crawling, users still end up spending loads of time only to find an insufficient amount of relevant information required, with some even taking up to 6 months to review multiple revisions before completion [1].

While it is unsure what is the worth of search engine in literature review since it is not disclosed, Google Scholar has around 390 millions of queries as of 2018, demonstrating how much demand there is in effective search engines, or literature review platforms [2].

2.2. Current Solutions

We have investigated some of the existing platforms, that potentially serve as our competitors, like the techniques they have deployed and the methods to deploy that made them stand out from the rest, as seen in Table 1. From here onwards, we can derive our own platform that has a unique and competitive edge than the rest of the competitors.

Platforms	Techniques	How they employ
Semantic Scholar	Use the articles to analyze your preferences.	If the user hasn't searched by the engine, the website will ask you to input the area you are interested in. Otherwise, the model will just recommend the latest work or influential thesis your interested area.
Google Scholar	Determine relevance using statistical models	Models required topics of your articles, the places where you publish, the authors you work with and cite, the authors that work in the same area as you and the citation graph.
Papers With Code	Recommend the latest trending research in Computer Science.	Since PWC only contains code-related works, there's no obvious recommendation system applied.
Connected Papers	Knowledge Graph.	By a graph including Citation, Prior Works and Derivative Works.
Paper Weekly	Trending research and Personalized Thesis in AI area.	Multi-functional Recommendation by the users' history.

Table 1 Competitors and their Unique Techniques and Methods

2.3. Market Research

We did a survey amongst our peers to understand their habits and to find out what are the pain points to be answered. Questions ranging from the user demographic to the platforms they use will ensure that we understand users' demands and enhance the platform usage experience, setting the foundation to our platform. In total, we have around 150 non-duplicated feedback from our survey, proving the credibility of the statistics.

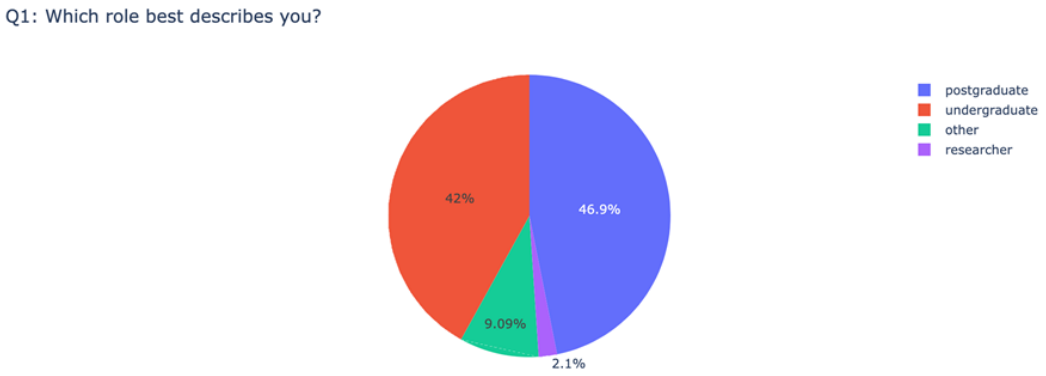


Figure 1: Demographic of Target Audience

Figure 1 above shows that majority of our potential users are either undergraduates or postgraduates, which falls under the category of college students, proving the suitability of our academic platform and recognizing the research needs of our target audience.

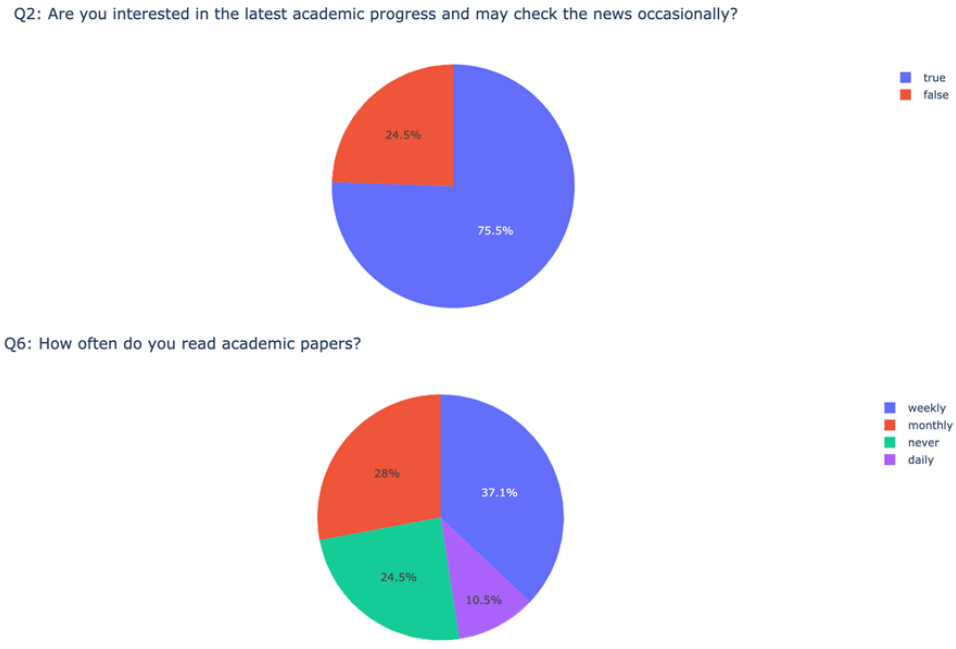


Figure 2: Interests in Academic Trends and Frequency in Literature Review

According to Figure 2 above, about 1/3 of the target audiences read academic paper on a weekly basis and that ¾ of the target audiences are interested in catching up on the latest academic trends and progress. Combining these 2 sets of information allows us to approximate the frequency of recommending up-to-date information on our platform.

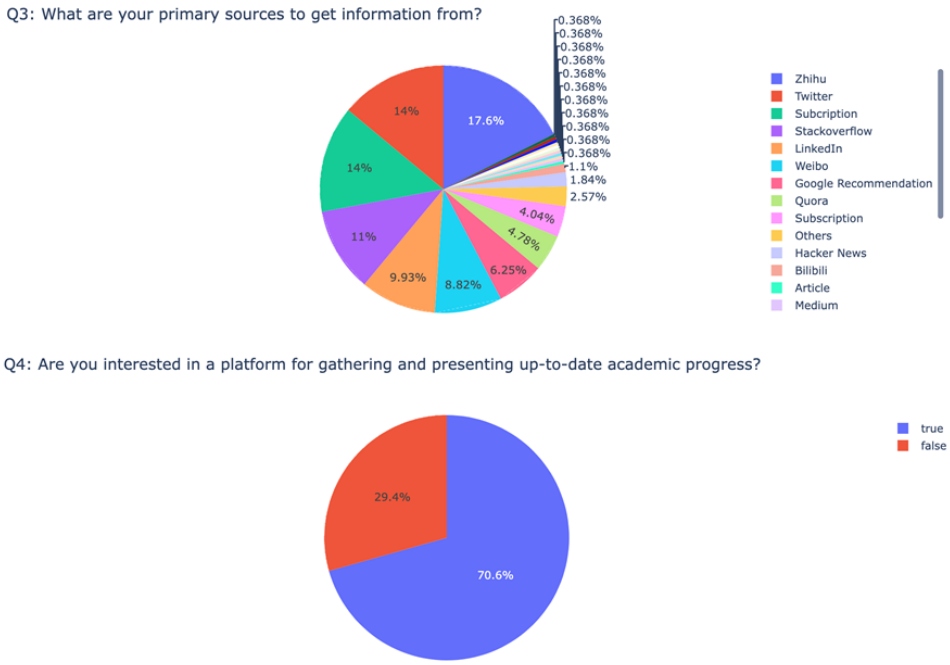
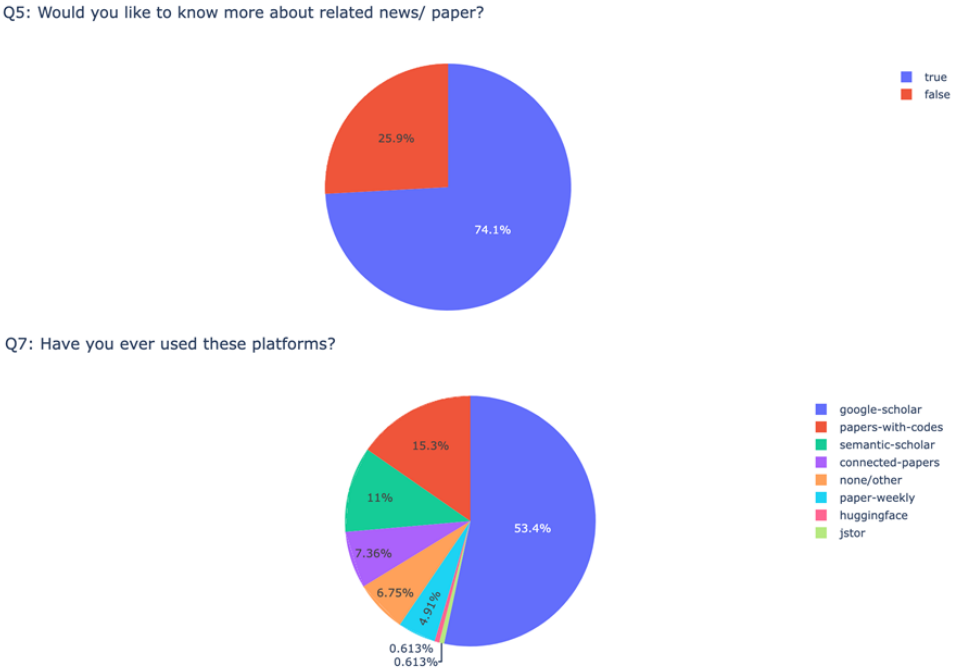


Figure 3: Information sources

In Figure 3 above, we found that Zhihu, Twitter, Stackoverflow and LinkedIn are some of the more popular platforms target audiences tend to rely on for news and information and that around ¾ of the respondents are interested in a platform that presents the latest academic trends and progress. As of October 2023, there is no such platform that unifies information from various sources and it is a highlight we want to pursue for our academic platform.

1. *Platform Utility & Interconnectivity*



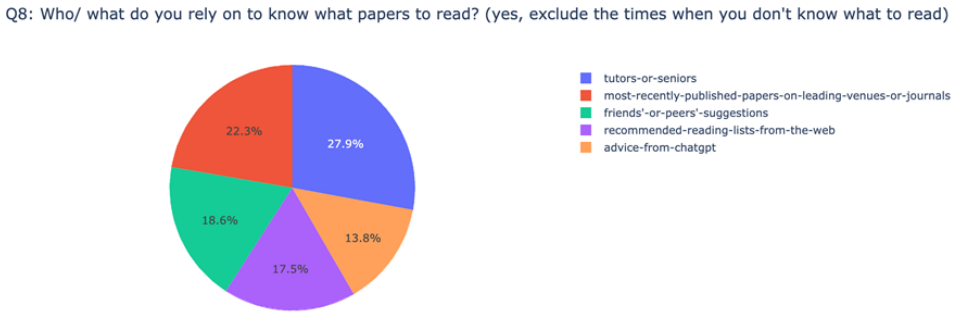


Figure 4: Platform Utility and Interconnectivity

In Figure 4 above, Q5's pie chart shows that about ¾ of the respondents are interested in related information to the current piece of information they are looking at, Q7 displays that more than half of the respondents use Google Scholar, followed by Papers with Code and Semantic Scholar, and Q8 shows a somewhat equal response of the sources users rely on for information.

From Q7's statistics we can derive that users usually spend majority of the time searching for papers and not being provided with customizable recommended papers of a specific research field since that is the nature of Google Scholar and Semantic Scholar as such. In general, the responds show the general need for a centralized hub for academic information.

Overall, the survey findings aligned perfectly with our initial assumptions. A significant level of interest and user feedback strongly supported the idea that there is a distinct need and potential for an academic paper search and recommendation platform.

2.4. Project Objectives

With the market research established and some of our competitors researched, we can define the goal of our platform and its functions distinctively. **Mango** is an academic knowledge platform designed for college students and researchers.

It has two main components:

- A **search engine** for papers
- A **recommendation system** for academic papers

Initially, users are forced to create an account in which they provide attributes like their interested fields, the role they play, the organization/ educational institute they are in, as a cold start for the recommendation system. As they continue to use our platform's search engine to look for papers they need, explicit feedback like clicking on the paper's link, bookmarking or liking the paper will train the platform's recommender system to recognise for more accurate patterns to suggest more relevant papers to the user.

This in turn will cut down user's time to look for the papers they require, form a generic understanding and to be updated on the latest development of a particular field of interest.

2.5. Success Measurements

According to the project objectives and some other relevant considerations, we can state our measurements towards success:

- User Growth and Retention. The first measure of success will be the number of users who register on Mango. Due to the limited time, this data will not be truly calculated, but it is a standard which must be taken into consideration. A consistent upward trajectory in user sign-ups would indicate that the platform's value proposition is resonating with its target audience. Moreover, it's crucial to monitor the retention rate; if users continue to engage with Mango over time, it shows that the platform is fulfilling its intended purpose.
- Recommendation Relevance. During Mango's development stage, the recommendation system's efficacy is assessed using faux user profiles created by ChatGPT. These simulated profiles, rich in varied interests and affiliations, guide the platform in generating paper recommendations.
- Search Engine Efficiency. The platform's search engine must deliver relevant results swiftly. Mango's search engine is designed not only to be fast but also to be highly intelligent in its approach. Efficiency and relevance are paramount. Here's a more detailed perspective on how its success can be gauged:
  - Keyword Matching: The most fundamental feature of any search engine is its ability to fetch results that match the input keywords. The accuracy of these results, and their ranking based on relevance, will be a primary measure of success.
  - Vector Search Algorithm Integration: Beyond mere keyword matching, Mango employs a vector search algorithm. This advanced feature maps out the semantic relationships between papers, enabling the platform to recommend academically related content even if it doesn't directly match the search term. The efficiency of this algorithm can be assessed by the relevance of the recommendations it provides, as well as by its ability to expose users to new yet pertinent academic content.
  - Speed of Retrieval: In today's fast-paced digital age, speed is of the essence. The time taken from inputting a search to receiving results is a crucial metric. Swift results not only enhance user experience but also signify a robust backend infrastructure.

3. PROJECT SOLUTION (SYSTEM DESIGN AND MODEL)

3.1 Architecture Overview

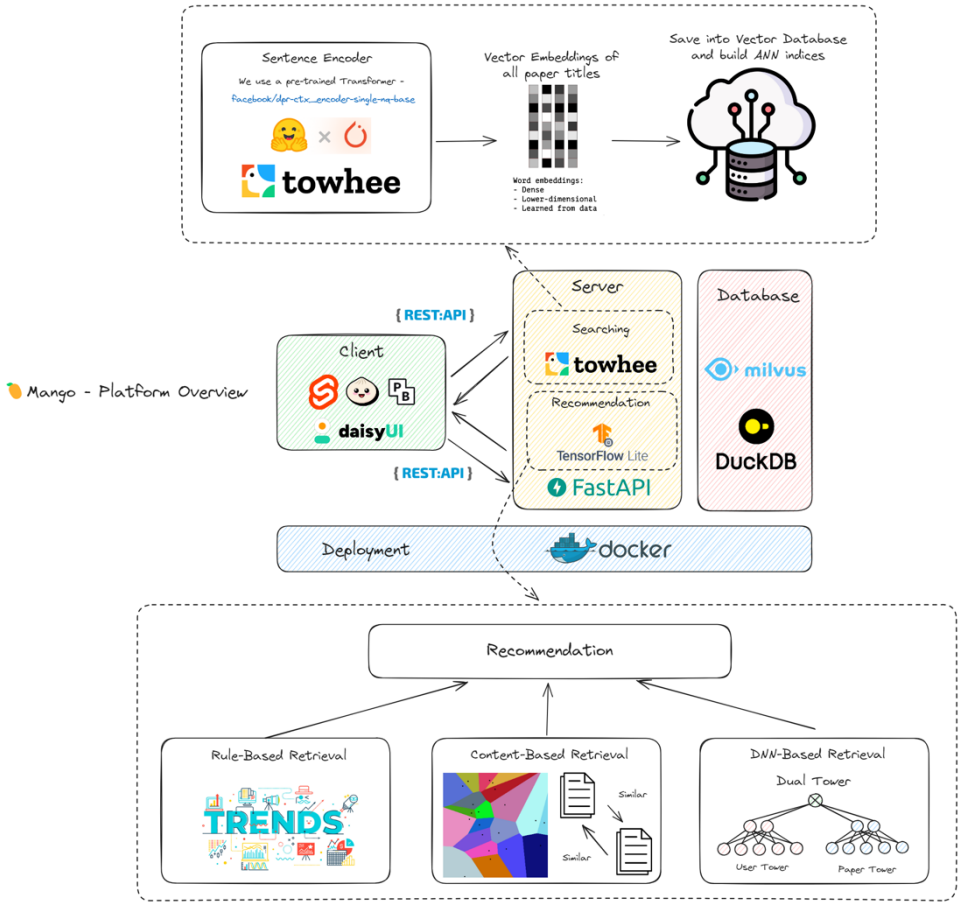


Figure 5: Architecture Overview

Detailed Workflow

1. Client:

- User can search and get recommendation of papers on UI.
- UI will also display paper details on the website.

2. Server:

- Handle client requests and gather results.

3. Database:

- RDBMS will store paper details, profiles, and user profiles
- Vector database stores pre-trained paper title embeddings and profile embeddings.

4. Deployment:

- Deploy Web UI and Server with Docker

Technical Stack and Architecture

1. Frontend:

- Technologies: Sveltekit, Bun, Pocketbase, DaisyUI, and TailwindCSS

- Interaction: RESTful API

2. Backend:

- Technologies: FastAPI, Tensorflow Lite, Towhee, AioHttp
- Interactions: RESTful API

3. Database

- Technologies: DuckDB, Milvus
- Interactions: Python Module - duckdb, pymilvus

3.2 Data Management

Data Sources

There are two main data sources -

- **Semantic Scholar** dataset (https://www.semanticscholar.org/product/api)
- **Paper With Code** dataset (https://github.com/paperswithcode/paperswithcode-data)

Data Preparation

We download the full datasets including **400,000+** papers. We filter and only keep papers which have rich information and in a selected fields of study. And finally, we got our full datasets at a level of **100,000+** papers.

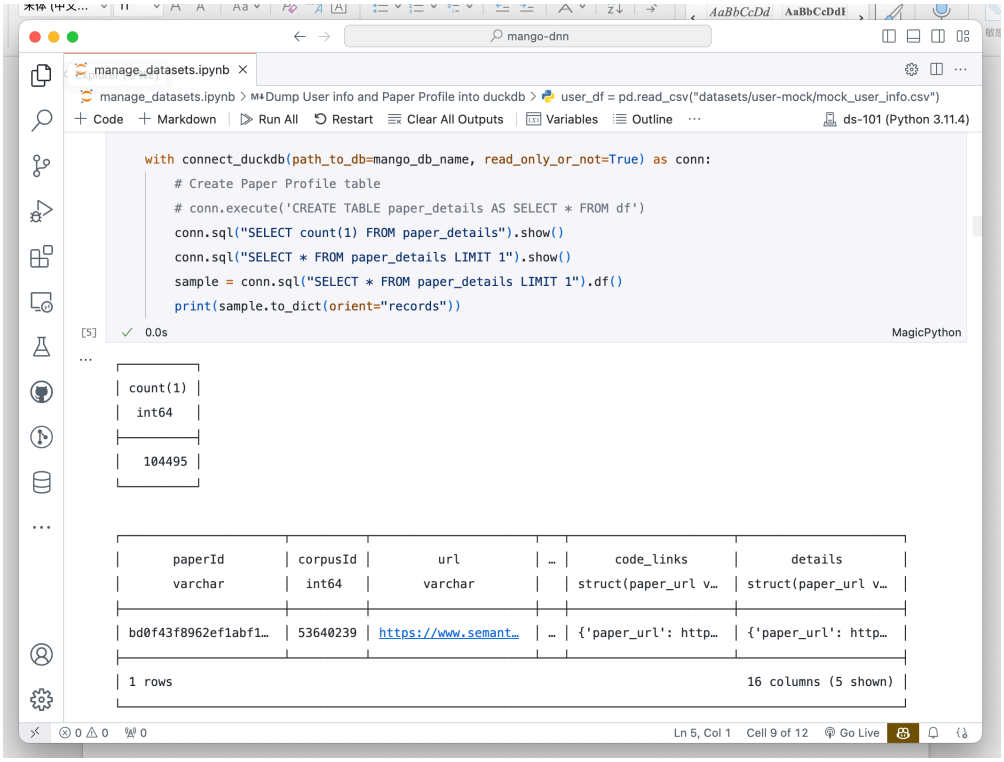


Figure 6: Raw datasets of all papers

```
dict_keys(['paperId', 'corpusId', 'url', 'title', 'venue', 'year', 'citationCount', 'isOpenAccess',
'fieldsOfStudy', 's2FieldsOfStudy', 'tldr', 'embedding', 'authors', 'id', 'code_links', 'details'])
```

Figure 7: Schema of a paper detail record

Pre-processed Training Dataset

Training dataset is generated as a mock dataset to prepare a set of dual tower model's pretrained weights. The training data is generated with features as shown in Table below. This set of data is supposed to mimic the recording of information when the user clicked or liked on the paper link. The act of liking the paper link is a form of explicit feedback while the act of clicking the paper link is a form of implicit feedback in which it tells us if the users prefer to understand the paper more in depth.

Encoding User Information	Encoding Paper Information
User ID	Paper ID
User's Role*	Author ID 1
User's Organization*	Author ID 2
User's Interested Field 1*	Author ID 3
User's Interested Field 2*	Field of Study 1
Day of clicking paper	Field of Study 2
Hour of clicking paper	Year
	isOpenAccess

*Static information to assist in cold start

Table 1: Features of Training Dataset

The mock data is generated under the condition of matching the user's interested fields with the paper's field of study at a certain probability, in hopes that the dual tower model will be trained to recognise the naive correlation between the user and the papers they prefer but also not overfit the model.

Vector Database for ANN indexes

We use - Milvus, a vector database to store all the pre-trained paper embeddings. There are two kinds of paper embeddings -

- Paper Title embeddings generated by a pretrained Transformer-based DPR Encoder:
  - *facebook/dpr-question_encoder-single-nq-base*
- Paper Title and Abstract embeddings generated by a pretrained BERT-based Encoder:
  - *allenai/specter2_aug2023refresh_base*

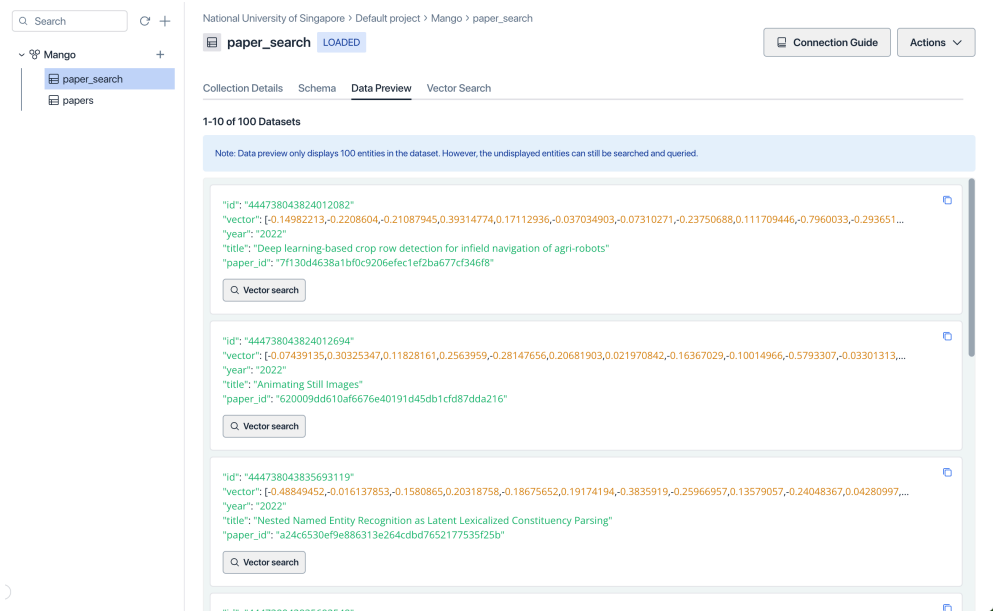


Figure 8: Preview of paper embeddings

4. PROJECT IMPLEMENTATION

This section talks about the tools and techniques used to implement different subsystems.

4.1 Information Retrieval System – Dense Passage Retrieval

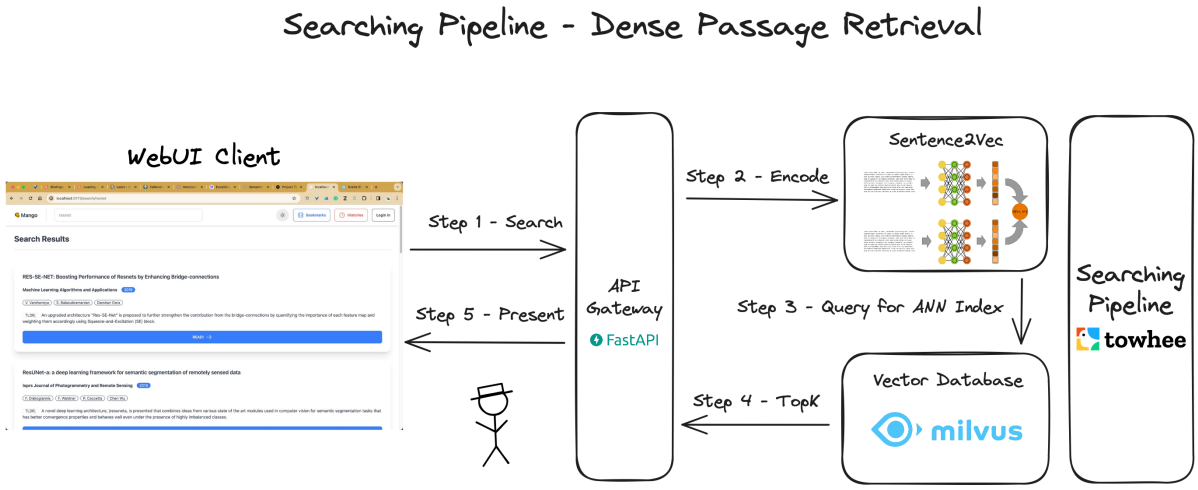


Figure 9: Information Retrieval System Pipeline

Mango's search system utilizes the principle of Dense Passage Retrieval (DPR), an innovative method that combines traditional keyword searches with dense vector comparisons to retrieve contextually relevant academic content.

Let's delve deeper into how Mango leverages DPR:

User Input and Query Understanding:

- The journey begins when a user inputs data, be it a singular word, an intricate phrase, or an entire academic paper title. In the realm of DPR, this input is more than just a keyword; it represents a semantic query that Mango will parse to understand the deeper intent and nuances.
- In the future, we plan to add a POSTAG mechanism on this phase. This mechanism can help the system to filter out the keywords which is professionally relevant to the user's need.

Semantic Scholar's Keyword Retrieval:

- The initial phase of the search is rooted in traditional mechanisms. Leveraging the functions of Semantic Scholar, Mango's system fetches content that directly matches the user's input keywords.
- This ensures the capture of academically relevant content based on explicit matches.

Vector Transformation and Dense Vector Comparison:

- Parallelly, the DPR mechanism springs into action. Using Towhee, which specializes in converting user queries into dense vectors spanning 768 dimensions, capture the essence of user input.
- The input is transformed into a 768-dimension dense vector, encapsulating the deeper semantic essence of the query. This dense representation is then matched against a vast database of vectors on the Milvus platform.
- Unlike traditional methods that hinge on exact keyword matches, DPR dives into the dense vector space to identify passages that are semantically close to the user's query, even if they don't share explicit keywords.

Integrated Result Presentation:

- Mango's prowess lies in its ability to cohesively blend the results from both the Semantic Scholar keyword search and the DPR vector comparison. Users are presented with an enriched result set on the search page, which includes direct matches and semantically relevant passages. This amalgamation ensures a panoramic view of content, from exact matches to those that share deep contextual relevance.

4.2. Recommendation System (Trending Page)

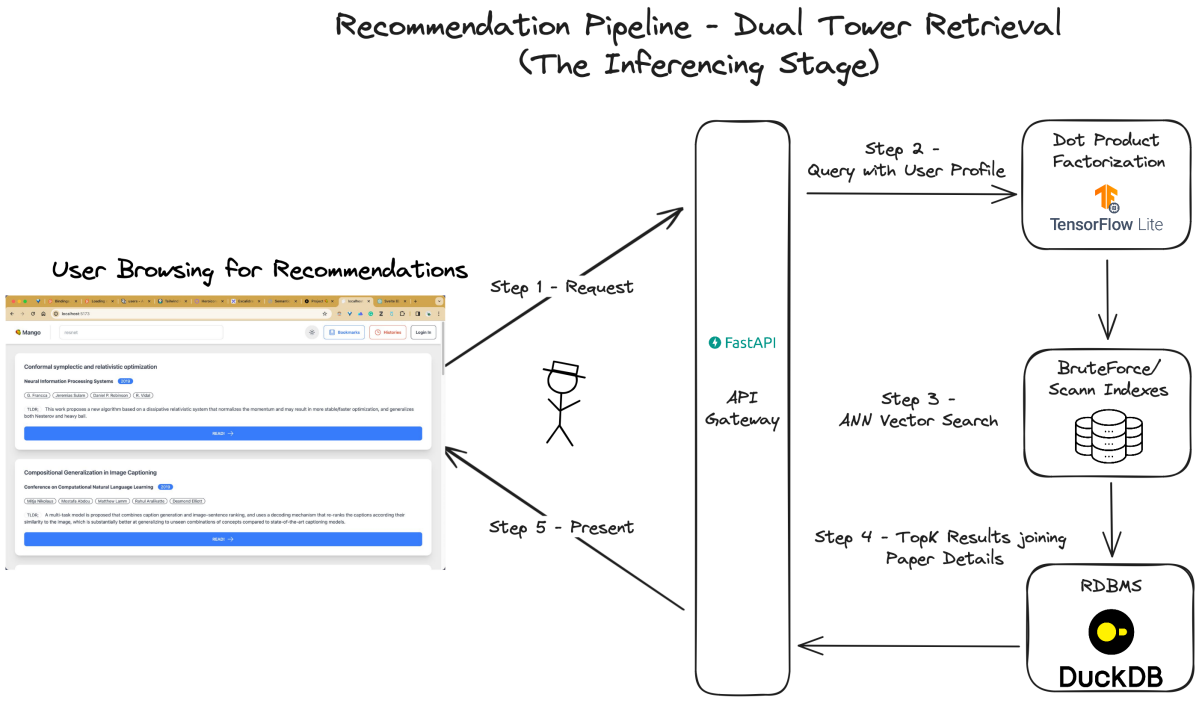


Figure 10: Pipeline of Recommendation System – Dual Tower Model Retrieval

On the trending page of our platform, the recommended papers are churned out by our recommendation system, which is a dual tower model. Figure 10 above shows the pipeline of the recommendation system, in which the dual tower model is responsible from receiving the user query to providing the top k candidates, in this case the academic papers, back to the platform.

Dual Tower Model

The two-tower model, also known as the dual-tower model, is a well-known model in recommender systems. The model utilizes 2 deep neural networks, the query model and the candidate model, to transform users and items into a lower-dimensional space. It can then retrieve candidates and predict ratings of candidates based on the spatial relationships between the embeddings of users and items in this reduced-dimensional space. The two-tower model essentially combines collaborative filtering and content-based filtering by utilizing low-dimensional representations for both users and items [8].

We use dual tower model as our recommendation system since it can tackle the long-standing cold-start issue by combining both the users' and papers' features in the form of creating embeddings of users and items using covariate representations, hence providing accurate recommendations for new users. Its deep neural network structure provides a flexible means of representing users and items, allowing it to capture nonlinear covariate effects much better than the linear models commonly found in the literature. By separately modelling users and items, it can capture nuanced user preferences and item characteristics, leading to more accurate and personalized recommendations [8].

Training Process

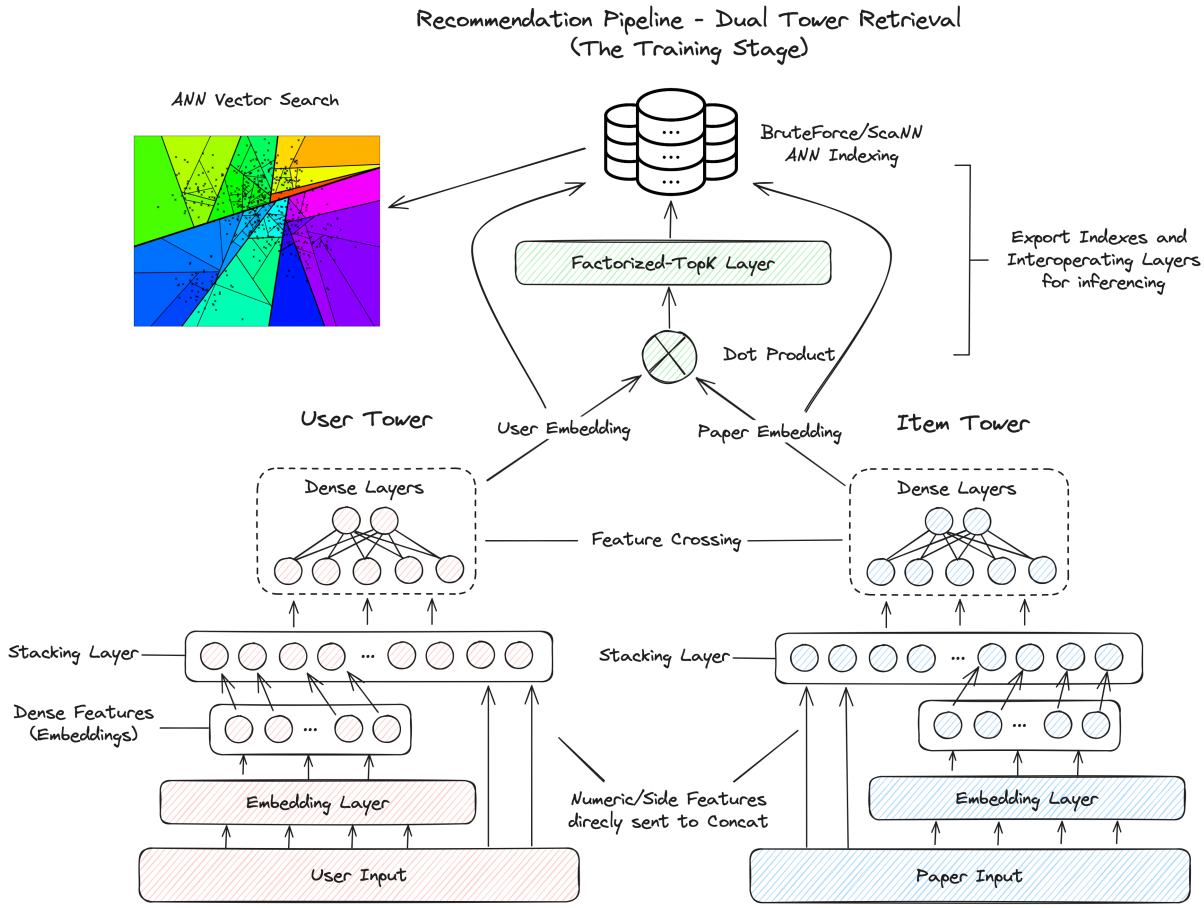


Figure 11: Training Sequence of Dual Tower Model

Figure 11 above shows the architecture of the dual tower model as well as the derivation of ANN vector indices of paper recommendations. In this case, the dual tower model is built using Tensorflow.

Both the paper dataset and training dataset are pre-processed to contain features as reflected in Table 1. Lookup tables for each feature are also set up to map strings to integers. The features are then passed through the respective query and candidate embedding layers, followed by the respective query and candidate dense layers (in this case, both query and candidate models have 2 hidden layers, reducing from 1024 embedding dimensions, down to 512 and lastly 256 in the output layer). The outputs will be a set of user and candidate embeddings in which via dot product will produce the top k candidates recommended for users. After training is done, the model's weights are saved, to be loaded for inferencing.

In order to conduct inferencing with the dual tower model, a Brute Force retrieval task is being set up to build the retrieval index. The query model is then provided in the constructor to process the sample data before performing retrieval. Upon passing the sample data into the Brute Force retrieval task, it will output k number of recommended papers via ANN vector search.

4.3. Recommendation System (Relevant Papers)

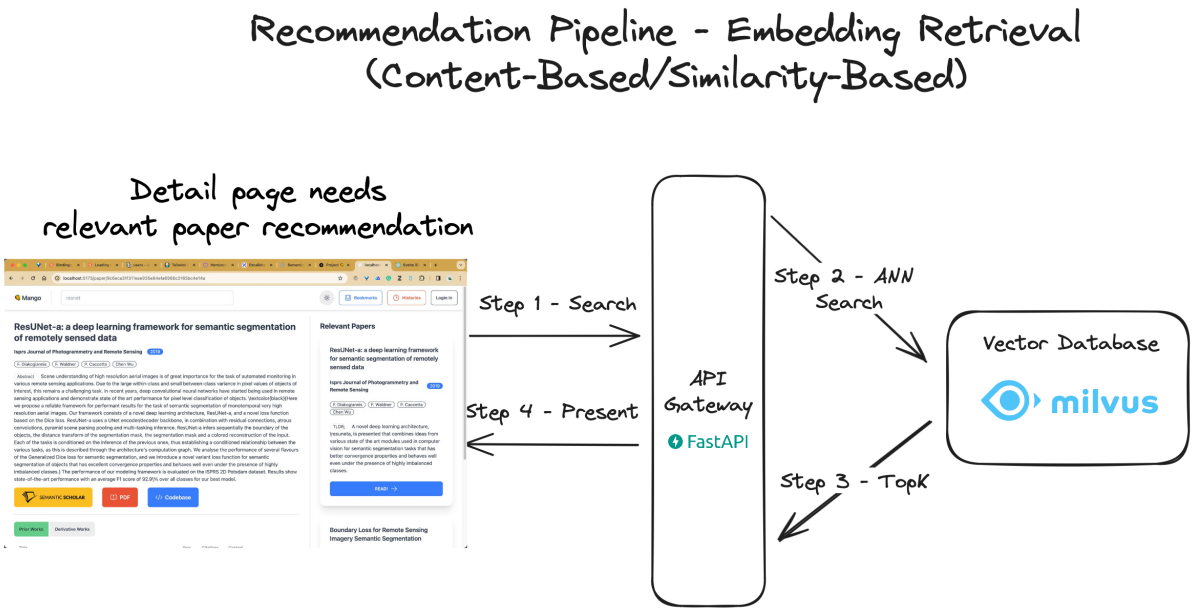


Figure 12: Pipeline of Recommendation System – Embedding Retrieval

In our quest to enhance the academic knowledge retrieval experience, we employ a sophisticated content-based recommendation retrieval system. This system uses the principle of leveraging the intrinsic characteristics of research papers to suggest highly relevant content to users. Much like our Information Retrieval approach, we focus on creating paper embeddings—mathematical representations of papers' content and context.

At the heart of this approach lies the formidable SPECTER2 model. This advanced deep-learning architecture specializes in the analysis of academic text. We employ SPECTER2 to encode the titles and abstracts of each research paper within our extensive database. This process transforms textual information into dense, high-dimensional representations, capturing the nuances and semantics of each paper's subject matter.

To enable rapid and efficient retrieval of relevant papers, we integrate the robust Milvus vector database into our system. Milvus serves as the repository for the encoded paper embeddings, facilitating the creation of specialized Approximate Nearest Neighbours (ANN) indexes. These indexes enable us to swiftly identify and serve the most pertinent research papers to users, significantly reducing retrieval times.

What sets our system apart is its dynamic, real-time recommendation capability. Whenever a user accesses the detailed page of a specific paper, our system springs into action. In addition to presenting the requested paper, it intelligently retrieves the top two ANN candidates from our meticulously constructed indexes. These recommendations are carefully selected based on their proximity to the viewed paper within the high-dimensional embedding space. Users can conveniently explore these related research materials, displayed prominently on the right-hand side of the screen.

5. PROJECT PERFORMANCE AND VALIDATION

We generated a survey for the same group of target audience to evaluate the usefulness and the accuracy of the platform.

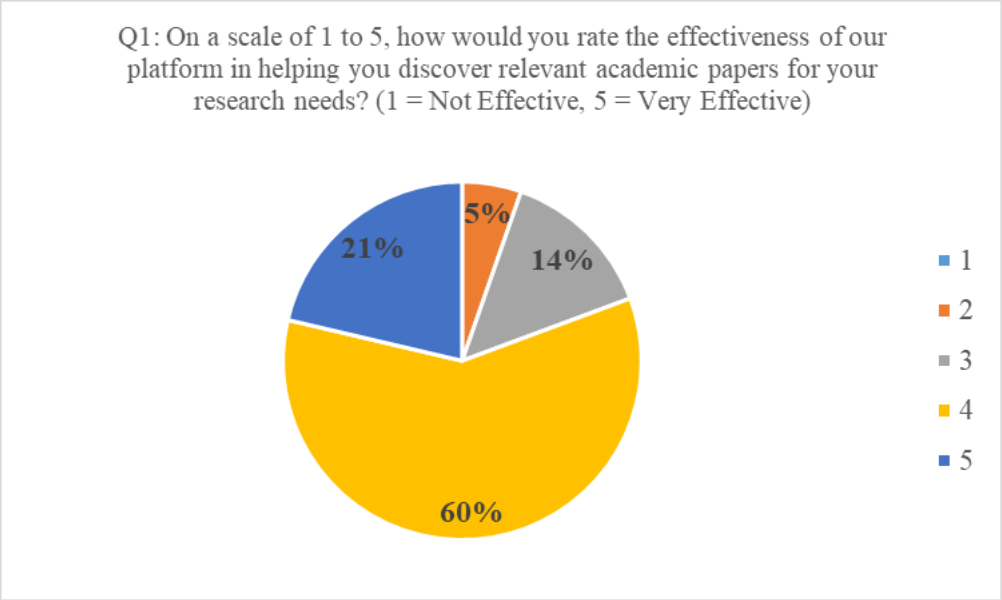


Figure 13: Feedback about Effectiveness of Relevant Papers section

Figure 13 above shows that about 2/3 of the respondents think that Mango's section of providing relevant papers under the search result is very effective in aiding their literature review quest, indicating that the section is essential.

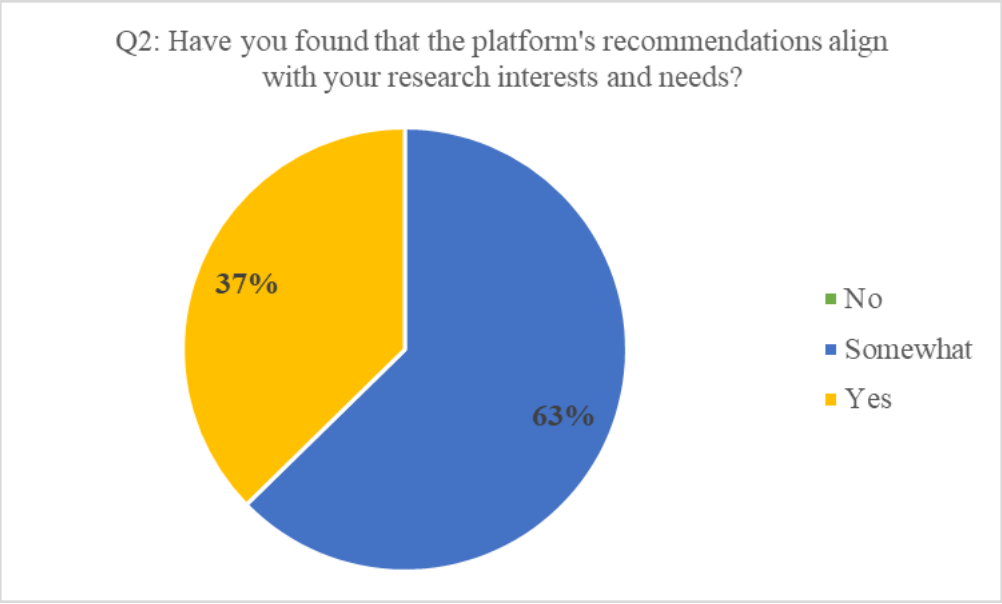


Figure 14: Feedback about Paper Recommendations of Mango

Figure 14 above shows that all the respondents think that the recommendation page is, at least, somewhat relevant to the information they are looking for, with 1/3 of the respondents thinking it is very relevant.

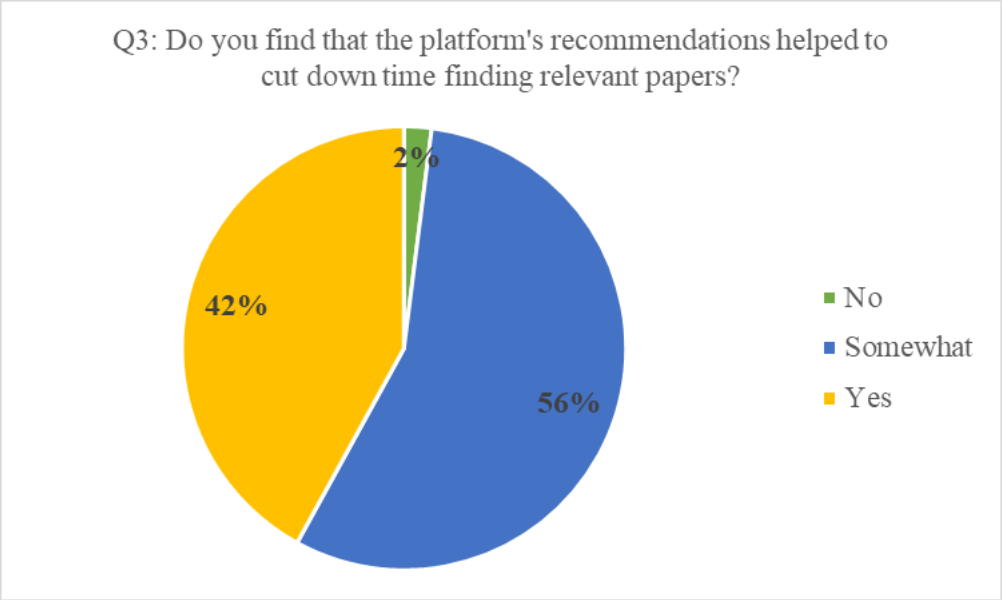


Figure 15: Feedback about Cutting down time with Recommended Papers

Figure 15 above shows that Mango's paper recommendations helped 98% of the respondents to cut down on time spent as compared to just purely searching on papers alone, with around 1/3 of them with a definite Yes for an answer, proving the accuracy of Mango's recommendation system.

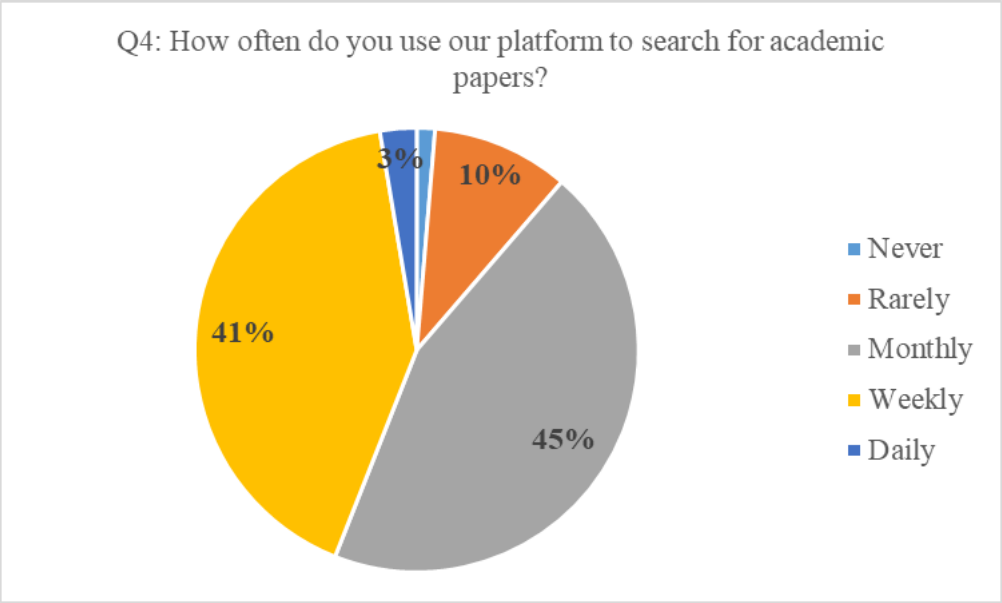


Figure 16: Feedback about frequency in using Mango.

Figure 16 above shows the frequency of using Mango as a source of information finding, with around 80% of the respondents confidently using it at least monthly, which is a good sign of how our platform helps the population.

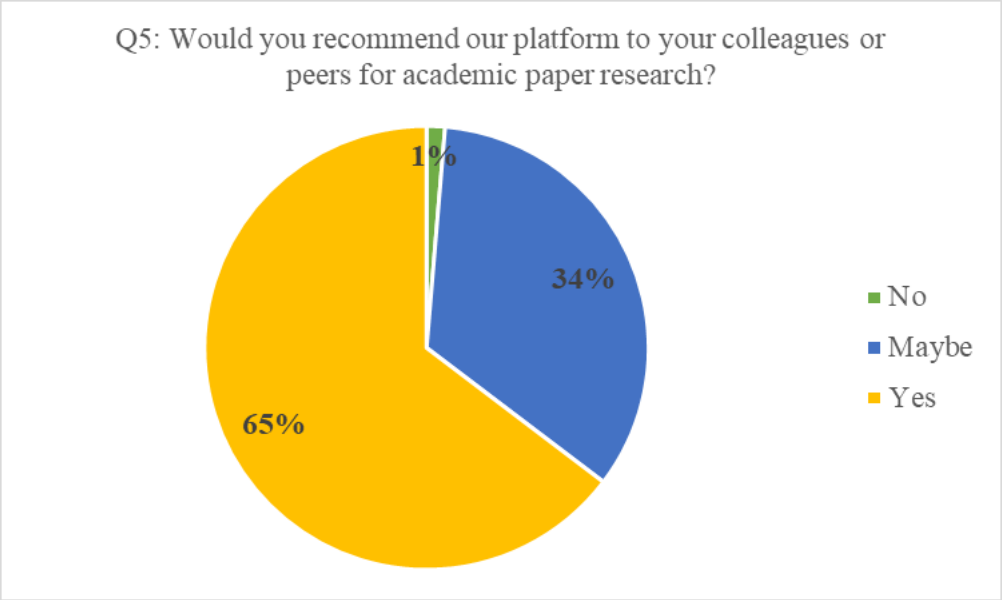


Figure 17: Feedback about recommending Mango to others

Figure 17 above is an indirect measurement of how relevant Mango is to helping practitioners and how satisfied our target audience are with Mango. As it shows, about 2/3 of the respondents have a definite answer of recommending our platform to the rest, with only 1% of the respondents having a negative response, which is a good sign of success.

Overall, the survey results provided positive feedback towards Mango, which dictates a success in the project.

6. CONCLUSION

6.1 Challenges and Solutions

Recommendation System: Dual Tower Retrieval Model

We did not implement a ranking task for the model, which could have helped to re-rank the recommended papers in terms of relevancy towards the user. Our mock data was not generated to include rankings of papers by users.

6.2 Future Considerations

Recommendation System: Dual Tower Retrieval Model

Consider to add more features like title of the paper, summary of the paper to be trained with

To build deeper query and candidate models stacked with more dense layers to capture more complex relationships and to recommend with higher accuracy

To add ranking task: Implement a ranking mechanism to prioritize the display of search results and recommendations. This ensures that users first see the most relevant papers based on their query or profile.

To scrape for more papers from more sources like Google Scholar, etc.

Broaden the database by extracting academic content from diverse platforms, including Google Scholar. This diversification enriches Mango's repository, making it more comprehensive and versatile.

To scrape for news posts from sources like Zhihu, Twitter

Capture real-time academic discourse and updates by scraping news posts from platforms like Zhihu and Twitter. Such content provides users with a pulse on current discussions and trends in their fields of interest.

To automate the recommendations at a certain frequency

Implement an automation mechanism to refresh recommendations at set intervals. This ensures that users consistently receive up-to-date suggestions aligned with the latest academic advancements and their evolving interests.

To improve the input keywords filtering

Implement the integration of a Part-of-Speech tagging (POSTag) mechanism. This advanced tool would sift through the user's input, isolating and emphasizing meaningful words, particularly those of professional or academic significance.

By filtering out the noise and neutralizing on these pivotal terms, Mango aims to enhance the precision and relevance of the DPR process. However, it's crucial to note that while this POSTag feature holds immense promise, it is not yet incorporated in the platform's current version.

APPENDIX A – PROJECT PROPOSAL

Date of proposal: Oct. 2nd, 2023

Project Title: "🥭 Mango" - An Academic Knowledge Platform

Group Members:

Goh Minhua A0285810A
Li Jiacheng A0285823W
Mao Zhihong A0285799X

Sponsor/Client:

As a young college student, Ken was filled with eagerness and readiness to dive deep into the intricate world of PhD research. His passion burned bright, but the path ahead was shrouded in a daunting expanse of papers and journals spanning decades. Ken recognized that, to truly grasp his research field, he needed to digest its most influential and up-to-date papers. Regrettably, there was no guiding compass illuminating his journey.

Countless nights were spent by Ken scrolling through databases, attempting to discern which papers held the key to the insights he sought. It wasn't merely about saving time; it was about ensuring he set out on the correct trajectory from the very beginning.

What Ken yearned for was a resource that could provide:

1. **Broad Coverage**: Enabling exploration of diverse fields.
2. **Timely Updates**: Keeping him informed about cutting-edge studies.
3. **Intuitive Interface**: Ensuring easy access and organization.
4. **Expert Curation**: Highlighting the essential papers.
5. **Collaboration Tools**: Facilitating connections with peers and mentors.
6. **Summaries**: Offering concise insights into papers.
7. **Customization**: Tailoring content to his evolving interests.

Ken's quest ultimately led him to an AI-driven research platform, empowering him with the tools he required for his academic journey. This platform wasn't just a dedicated resource; it was a virtual guide, streamlining the most recent and pivotal papers right to his fingertips. It was more than just a search engine—it was a beacon for young scholars, a curated library of crucial milestones that grounded and fostered innovation in any research field.

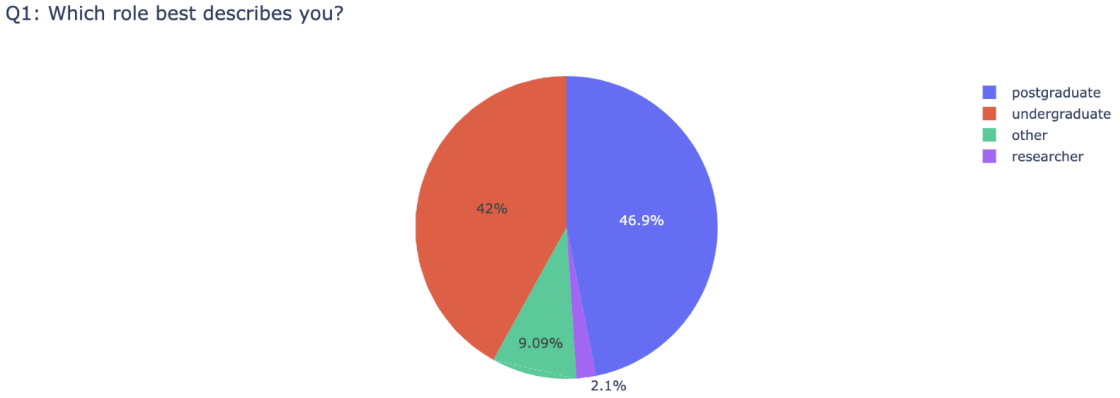
Ken firmly believed that such a tool wouldn't only aid his personal journey but would empower an entire generation of researchers to make the most out of their academic pursuits, ensuring that no seminal paper remained unread.

Background/Aims/Objectives:

Survey & Requirements Identification
To gauge the interest and needs of the academic community regarding a dedicated platform for academic essay searching and recommendation. By understanding users' current habits,

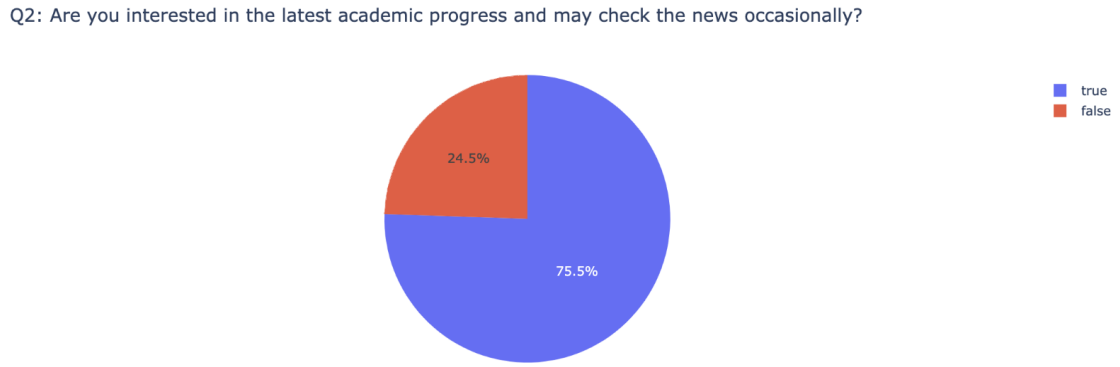
preferences, and challenges, we aim to design a tool that efficiently caters to their research demands and enhances their academic experience.

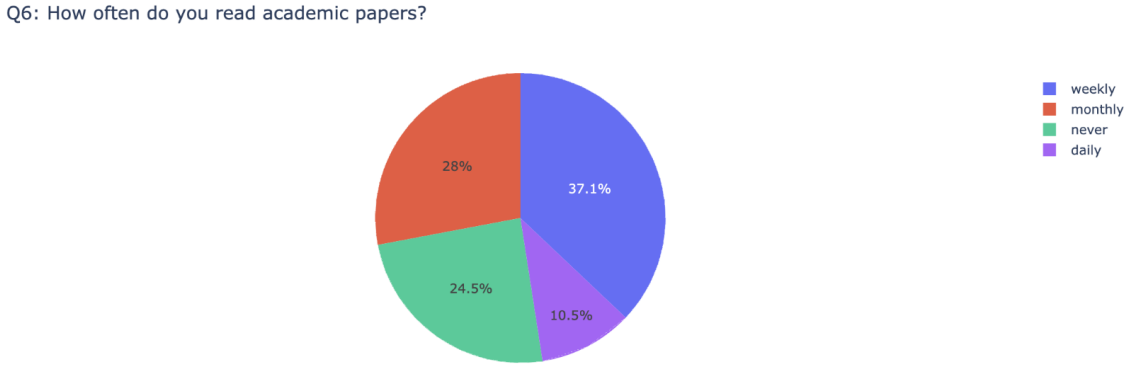
1. *User Demography & Role*



- The survey starts by determining the user's position: undergraduate, postgraduate, or researcher.
- This helps understand the academic stage and research needs of the user.
- We can tell from the result that our ***target users*** should be ***college students***.

2. *Frequency & Interest in Academic Updates*

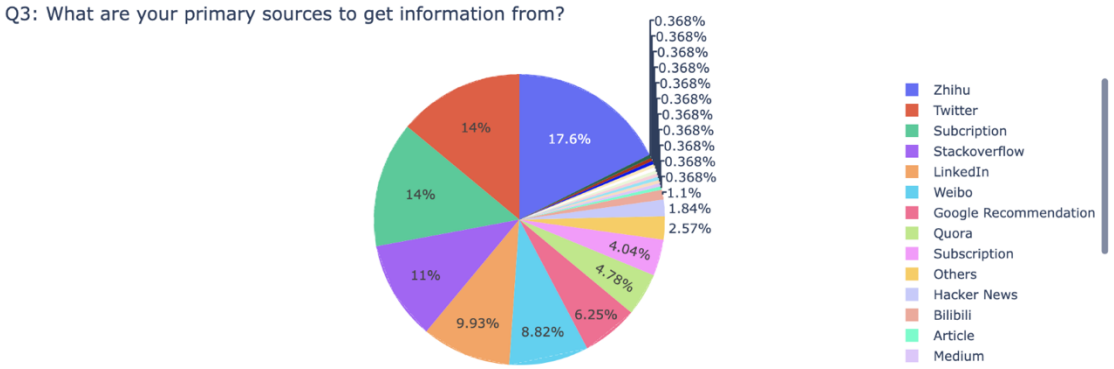


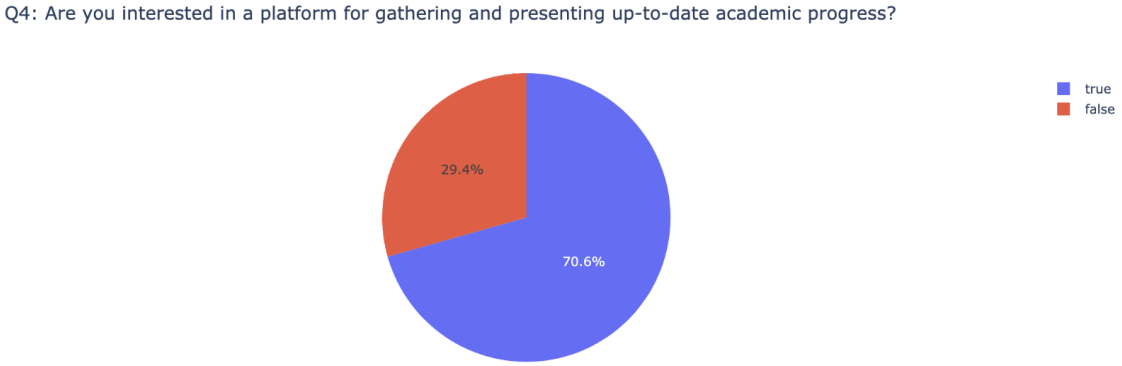


- Questions about the frequency and interest in the latest academic progress help gauge the general inclination towards being updated.
- This is fundamental for a platform focusing on up-to-date information.

- More than ¾ people state they are used to reading papers and tend to keep themselves updated on academic progress.

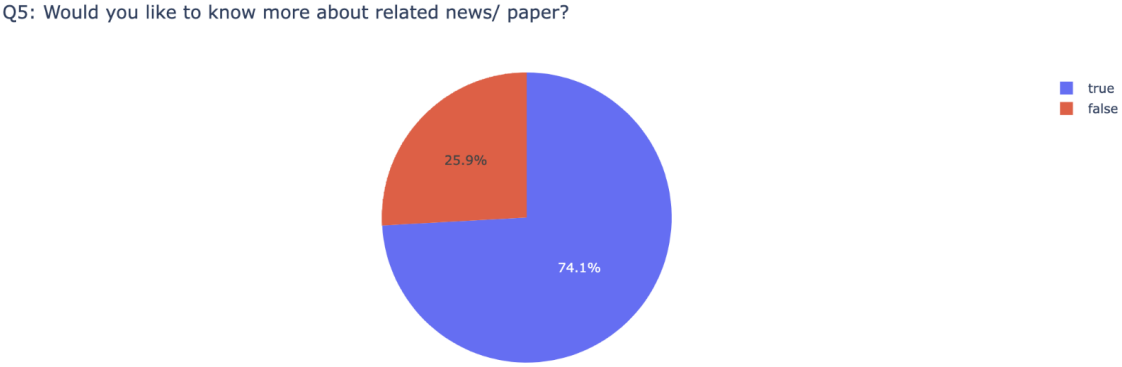
3. *Current Information Sources*

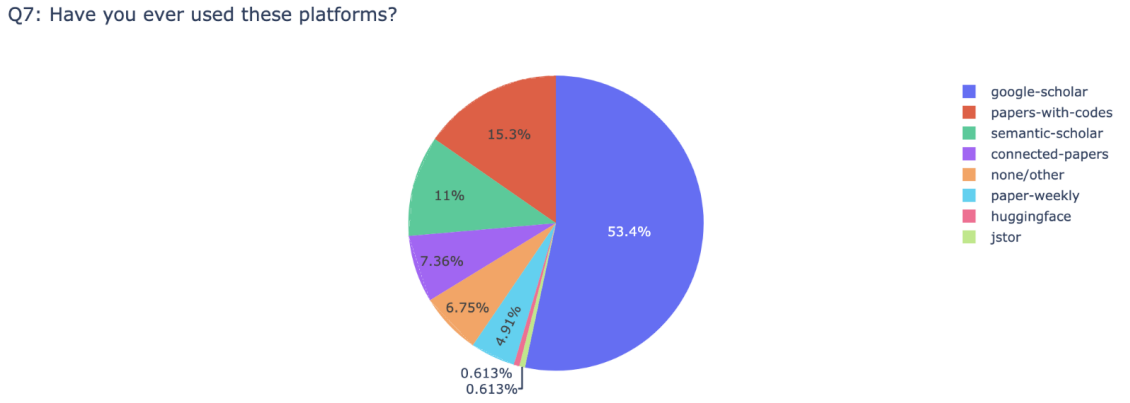


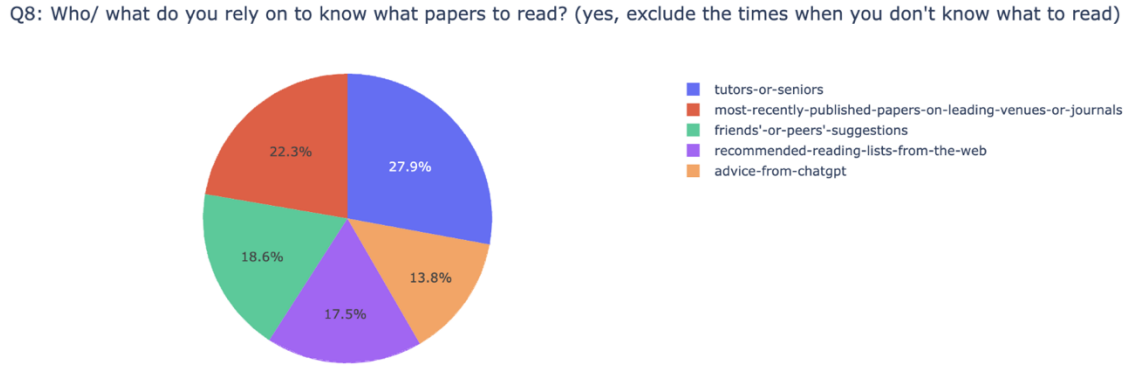


- Identifying the primary sources of information helps to ascertain current platforms' popularity and their possible shortcomings.
- Currently, students and researchers do not have a unified media platform for latest academic information, progress, and news. Most of our interviewees are interested in having such a platform.

4. *Platform Utility & Interconnectivity*







- Queries regarding interest in a platform for gathering recent academic developments and connecting related papers gauge the demand for a centralized hub for academic knowledge.
- According to the Q7 chart, we can find users usually spend much time in searching essays, but there is no such platform to provide customizable recommendation of academic readings to help them explore a certain research field.

The survey results resonated with our initial expectations. The overwhelming interest and feedback from users strongly advocate the necessity and viability of an academic essay searching and recommendation platform.

Objective

Mango is an ***academic knowledge*** platform for college students and researchers.

It has two main components -

- A ***Search Engine*** for essays.
- An ***Academic Papers Recommender***.

The platform aims to assist ***young college students*** and ***researchers*** to quickly grab a general understanding of a particular field and keep themselves up to date on the forefront development of that field.

Market Research - Competitor Research

	Techniques	How they employ
Semantic Scholar	Use the articles to analyze your preferences.	If the user hasn't searched by the engine, the website

		will ask you to input the area you are interested in. Otherwise, the model will just recommend the latest work or influential thesis your interested area.
Google Scholar	Determine relevance using statistical models	Models required topics of your articles, the places where you publish, the authors you work with and cite, the authors that work in the same area as you and the citation graph.
Papers With Code	Recommend the latest trending research in Computer Science.	Since PWC only contains code-related works, there's no obvious recommendation system applied.
Connected Papers	Knowledge Graph.	By a graph including Citation, Prior Works and Derivative Works.
Paper Weekly	Trending research and Personalized Thesis in AI area.	Multi-functional Recommendation by the users' history.

Potential Application

[Research Management] NUS students and professors can rely on our platform to aid in literature review by finding papers and information more easily. Some experts working in NUS research facilities may find relevant information to be scarce. With the search engine, it will only shrink the circle down to information relevant to their own fields.

[Educational Resources] NUS professors can curate educational resources, such as lectures, tutorials and guides, by reviewing academic papers and sometimes from academic news recommended to them, translating their research to NUS students' education in the classrooms.

[Research Funding Opportunities] NUS researchers belonging to facilities like NUS CRISP can look up about papers and receive news relevant to remote sensing in order to integrate information to craft their next upcoming research grants and funding opportunities.

Project Descriptions:

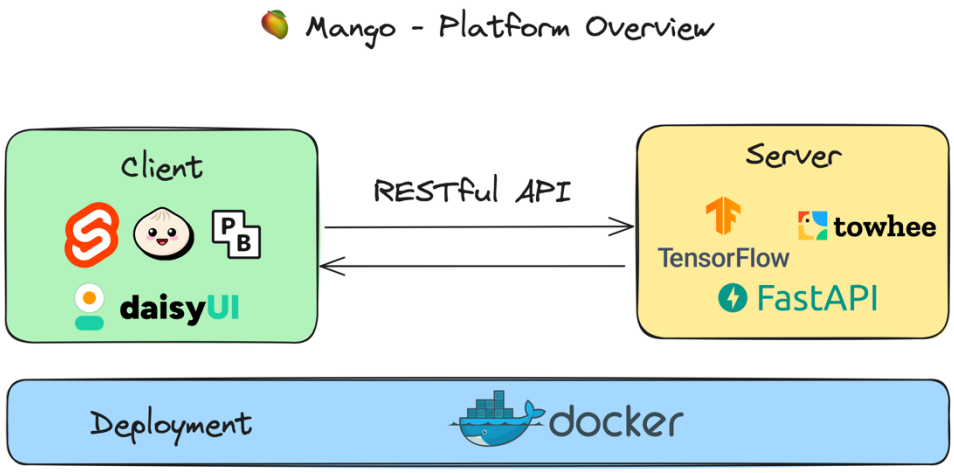
Literature Review

It is understood that some well-known search engines like Google Scholar have their proprietary algorithms kept a secret while others like Google's and Yahoo's are out in the open for the public to be studied. Since our platform closely resembles that of Google Scholar, one still should know as much as possible as to how much its algorithm works, albeit the complete algorithm is not public.

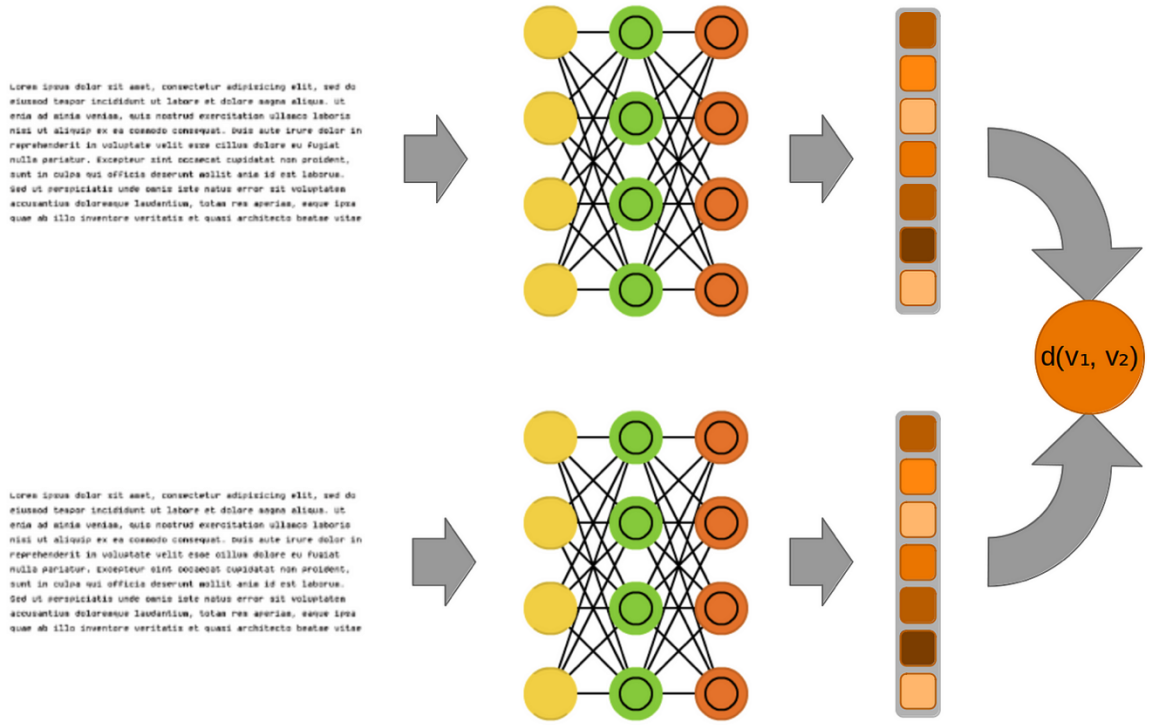
Google Scholar weighs heavily on words in the title, author and journal names, on articles' citation counts and takes into consideration of words directly included in an article but not synonyms of those words [1]. Google's search engine's first and most well-known ranking algorithm was PageRank, in which the number of links pointing to the web page influences its rank and links coming from high-quality sites are given more weight [2]. Yahoo's search engine utilizes Core Ranking function, which filters between good and bad URLs with Gradient Boosting Decision Tree, along with Logistic Loss [3].

More well-known recommendation systems like Google News developed a Bayesian framework for predicting users' current news interests from activities of the user like past click behaviour and the news trends as seen from user's activity. This framework of information filtering is combined with collaborative filtering mechanism to generate its recommendation system [4]. GroupLens, another news recommendation platform, adopts collaborative memory-based algorithm to gather the ratings of users and to predict scores of news articles for individual users, based on the heuristic rule that users who agreed in the past will probably agree again [5]. Amazon book recommendation website system also utilizes item-to-item collaborative filtering method to analyse users' book purchases and then recommend books purchased by other customers based on similar book(s) between the two customers have purchased. Other parameters like topics of interest, demographic characteristics aid to suggest books which the user may like [6]. Digg, another news curating platform, makes use of collaborative filtering to find interesting similarities between users and their search history, then recommending similar users' favorite articles to target user [7].

System Overview



Search Engine



Implementing a search engine using Dense Passage Retrieval (DPR) techniques involves several key steps. We begin by gathering a corpus of text data and preprocessing it. Next, we train two critical components: a retriever, which efficiently narrows down the search space using a pre-trained transformer model, and a reader, which comprehends selected passages and ranks them based on relevance.

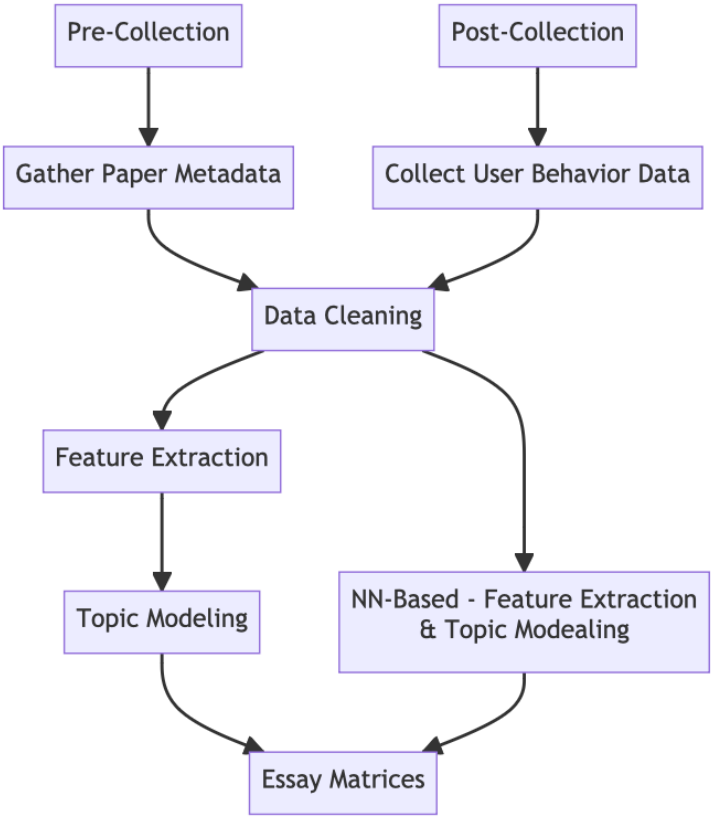
We then create an index for our corpus and, when a user submits a query, preprocess it and pass it through the retriever to obtain a list of candidate passages. These passages are further ranked for relevance using the reader.

Our search engine should have a user-friendly interface, be scalable for handling large volumes of data and queries, and continuously fine-tuned for better performance based on metrics like precision and recall.

Recommender System

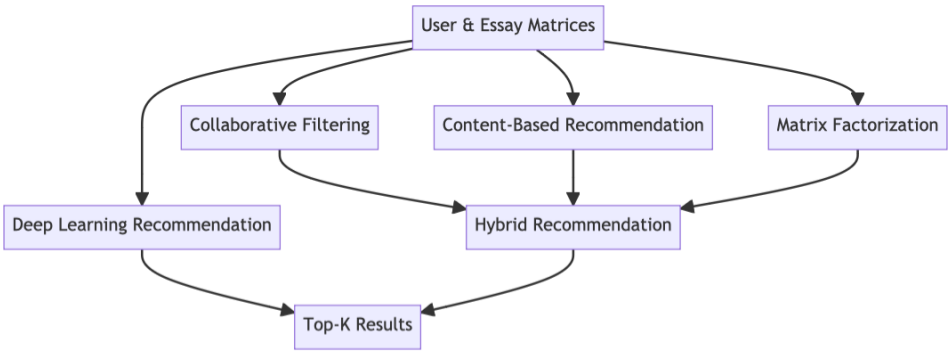
In this part, we will briefly elaborate how we plan to design our recommendation system -

1. ***Data Collection&Processing Process***



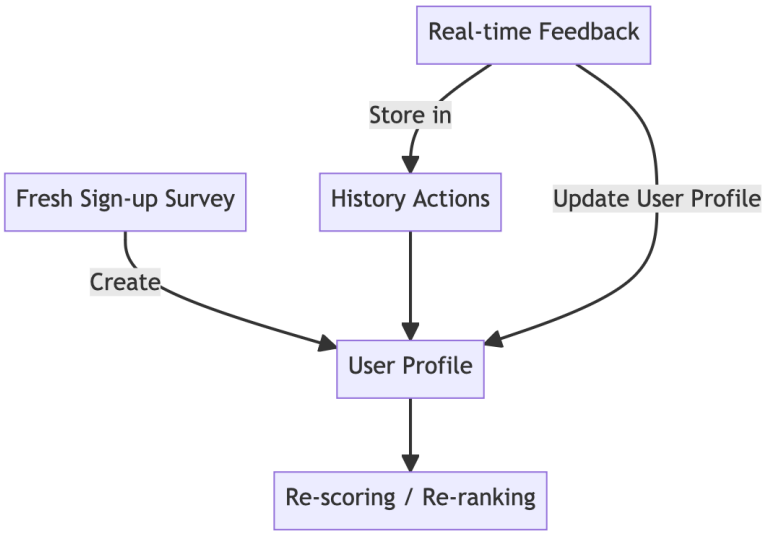
- **Paper Metadata**: Acquire details like title, abstract, keywords, authors, citation relationships, etc.
- **User Behavior Data**: Such as click-through rate, reading time, favorites, likes, etc.

2. ***Recommendation Algorithms***



- **Collaborative Filtering**: Uses user's past behavior and preferences to predict what else they might like. There's User-Item and Item-Item collaborative filtering.
- **Content-Based Recommendation**: Recommends papers based on their content, like keywords, abstract, etc.
- **Hybrid Recommendation**: Combines the techniques of collaborative filtering and content-based recommendations.
- **Deep Learning**: Leveraging deep neural networks like Neural Collaborative Filtering （NCF）, DeepFM, etc.

3. ***Personalized Recommendations***



- **User Profile**: Keep track of a user's research direction, browsing history, favorites, citations, etc., to create a detailed profile and enhance recommendation precision.
- **Real-time Feedback**: Allow users to give feedback on recommended papers, indicating "Interested" or "Not Interested".

4. ***System Features***

- **Real-time Updates**: As new papers get published, the system should update the recommendation pool promptly.
- **Interdisciplinary Recommendations**: For cross-disciplinary research, recommend papers from related fields.

Challenges and Roadblocks

Challenge:

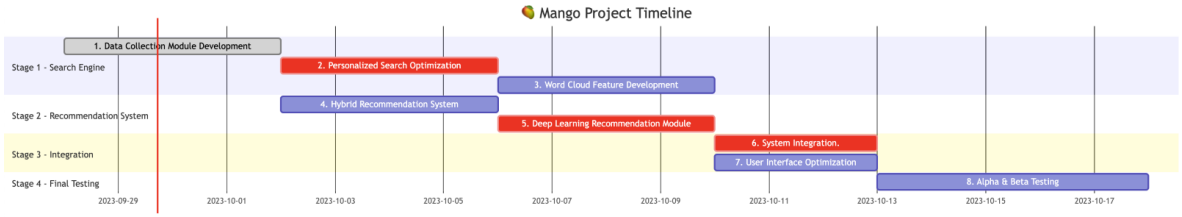
1. Chatbot's difficulty in secondary information extraction.
2. Sparse feedback due to the vastness of academic fields.
3. Limited data crawling scope.

Strategy:

1.Enhance with advanced NLP techniques. Use feedback loops for iterative learning.
2. Diversify feedback channels and employ matrix factorization for sparse data.
3. Expand source websites and consider partnerships with academic databases.

Future

Based on the tasks above and the estimated workload, we have drawn the following Gantt Chart to keep the development work on schedule:



References

[1] Beel, J., & Gipp, B. (2009). Google scholar's ranking algorithm: The impact of articles' age (an empirical study). *2009 Sixth International Conference on Information Technology: New Generations*. https://doi.org/10.1109/itng.2009.317

[2] Bar-Ilan, J. (2007), "Manipulating search engine algorithms: the case of Google", Journal of Information, Communication and Ethics in Society, Vol. 5 No. 2/3, pp. 155-166. https://doi.org/10.1108/14779960710837623

[3] Yin, D., Hu, Y., Tang, J., Daly, T., Zhou, M., Ouyang, H., Chen, J., Kang, C., Deng, H., Nobata, C., Langlois, J.-M., & Chang, Y. (2016). Ranking relevance in Yahoo Search. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/2939672.2939677

[4] Liu, J., Dolan, P., & Pedersen, E. R. (2010). Personalized news recommendation based on click behavior. Proceedings of the 15th International Conference on Intelligent User Interfaces. https://doi.org/10.1145/1719970.1719976

[5] Y. Xiao, P. Ai, C. -h. Hsu, H. Wang and X. Jiao, "Time-ordered collaborative filtering for news recommendation," in China Communications, vol. 12, no. 12, pp. 53-62, December 2015, doi: 10.1109/CC.2015.7385528.

[6] Han, K. (2020). Personalized news recommendation and simulation based on improved collaborative filtering algorithm. *Complexity*, *2020*, 1–12. https://doi.org/10.1155/2020/8834908

[7] Shan Liu, Yao Dong and Jianping Chai, "Research of personalized news recommendation system based on hybrid collaborative filtering algorithm," 2016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, 2016, pp. 865-869, doi: 10.1109/CompComm.2016.7924826.

APPENDIX B – MAPPED SYSTEMS FUNCTIONALITIES AGAINST KNOWLEDGE, TECHNIQUES AND SKILLS OF MODULE COURSES

MODULE COURSE	KNOWLEDGE/TECHNIQUES/SKILLS APPLIED
Machine Reasoning (MR)	Vector Database (milvus) RDBMS (DuckDB)
Reasoning Systems (RS)	Recommendation System • Dual Tower Model (deep collaborative filtering) • Content-based/ Similarity-based Embedding Retrieval
Cognitive Systems (CS)	Dense Passage Retrieval (sentence encoder via pretrained transformer)

APPENDIX C – INSTALLATION AND USER GUIDE

Mango - User Guide

Prerequisites

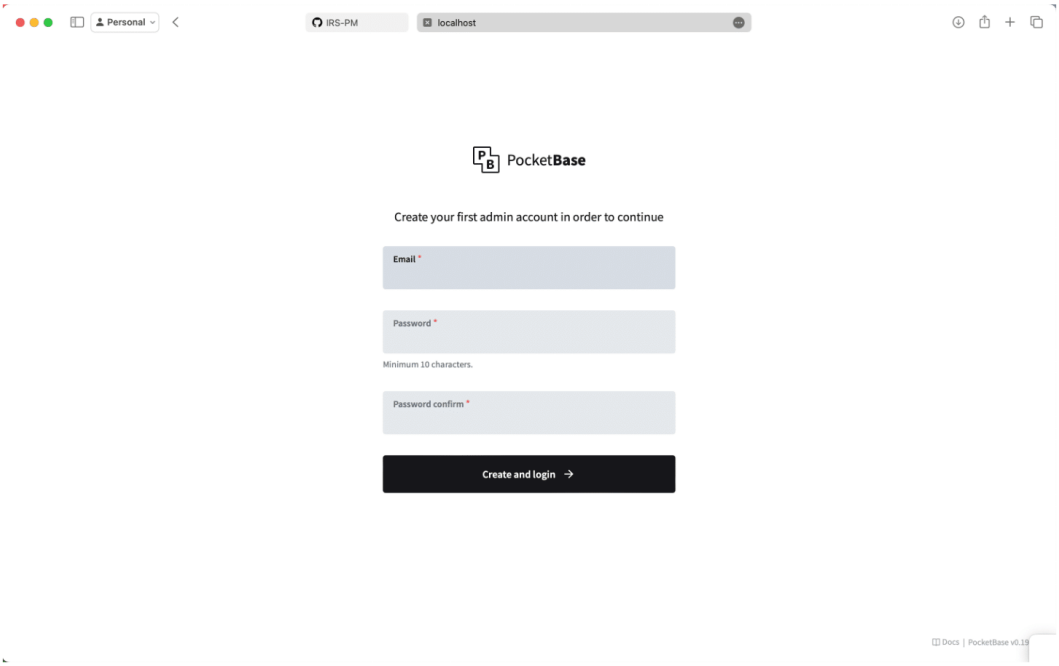
- **Docker**
- **Internet access:** Pre-trained paper embeddings are stored on ***Managed Milvus Cloud***

User Database Setup

Move to the same folder where `docker-compose.yaml` file are placed. Run the following command to set up your user profile database -

```
docker-compose up -d
```

Then, open webpage `localhost:8090/_/` to sign up your admin account -

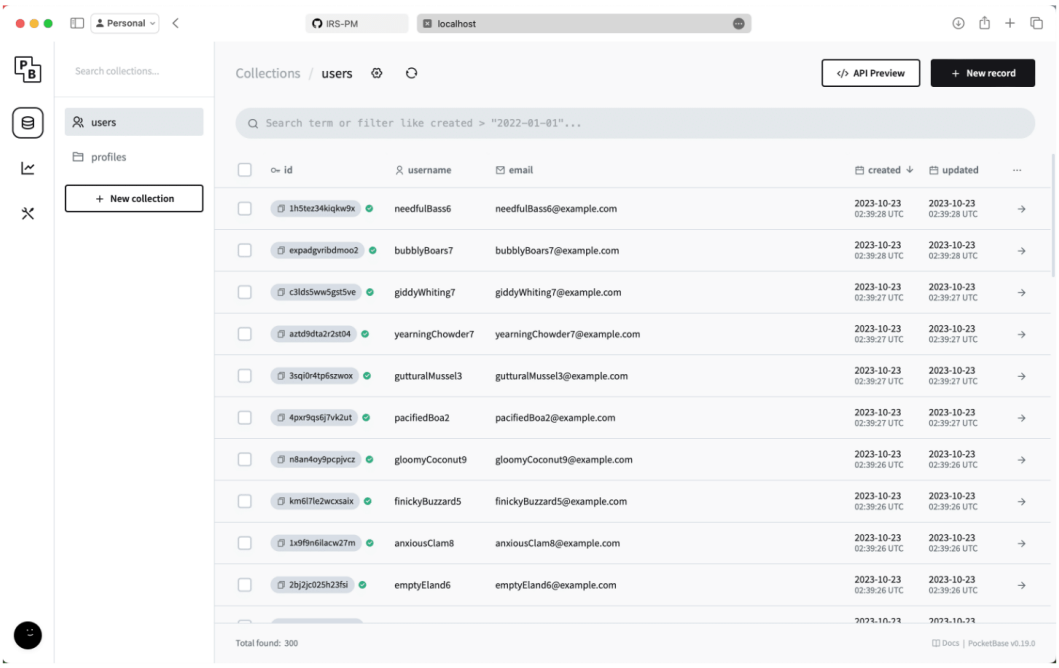


After you logged into the pocketable, move to the "settings" page and import the table schema with user data -

- Import `mango-user/pb_schema.json` to create collections.
- Upload `mango-user/backup1.zip` into the backup page and restore.

> ***The container may shutdown but it is OK. Don't worry.***

Run `docker-compose down` and re-run `docker-compose up -d` to restart the container. Then heading to the admin page you will find the records are already restored.



Then, you have successfuly setup the user database.

Backend Setup

Go to the `Mango-server` folder, and download the models, stores, and config file from our Google Drive. Unzip the `zip` file after successfully downloaded it.

- Download Link: https://drive.google.com/file/d/1M2TNcVArPhBpWrU9-Ok4fNLn3LXj-u2B/view?usp=share_link

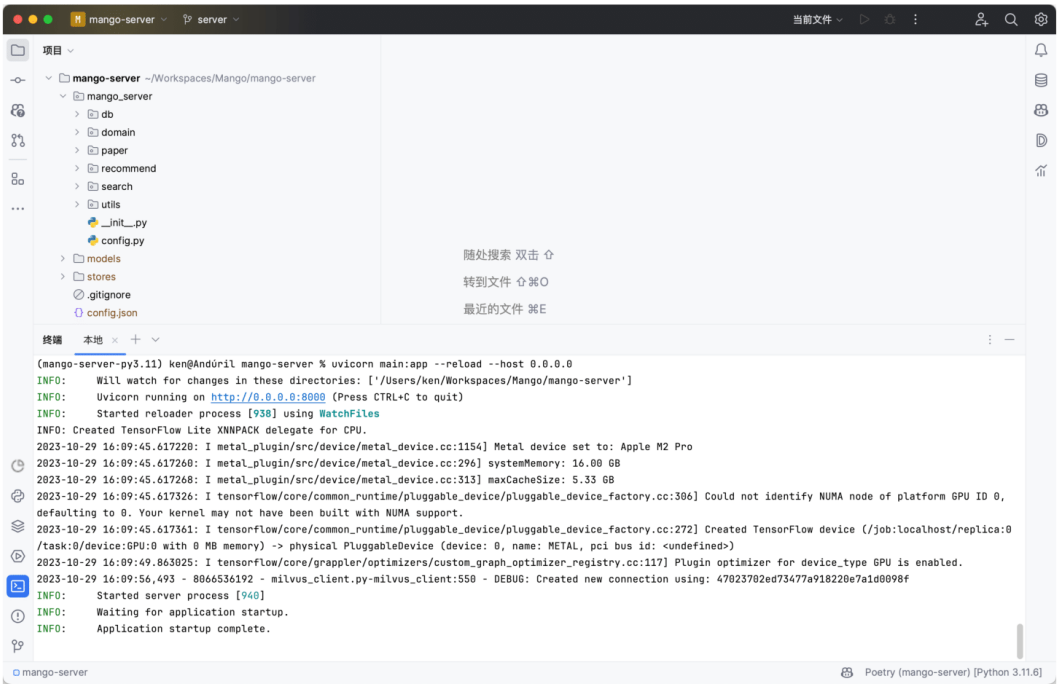
First, please follow official guide to install **Poetry** -

- https://python-poetry.org/docs/#installing-with-the-official-installer

Then run the command to install dependencies and setup Mango backend server -

```
# Install dependencies
poetry install

# Setup mango server
uvicorn main:app --reload --host 0.0.0.0
```

Once the server has been successfully activated, you will see logs like this -



Frontend Setup

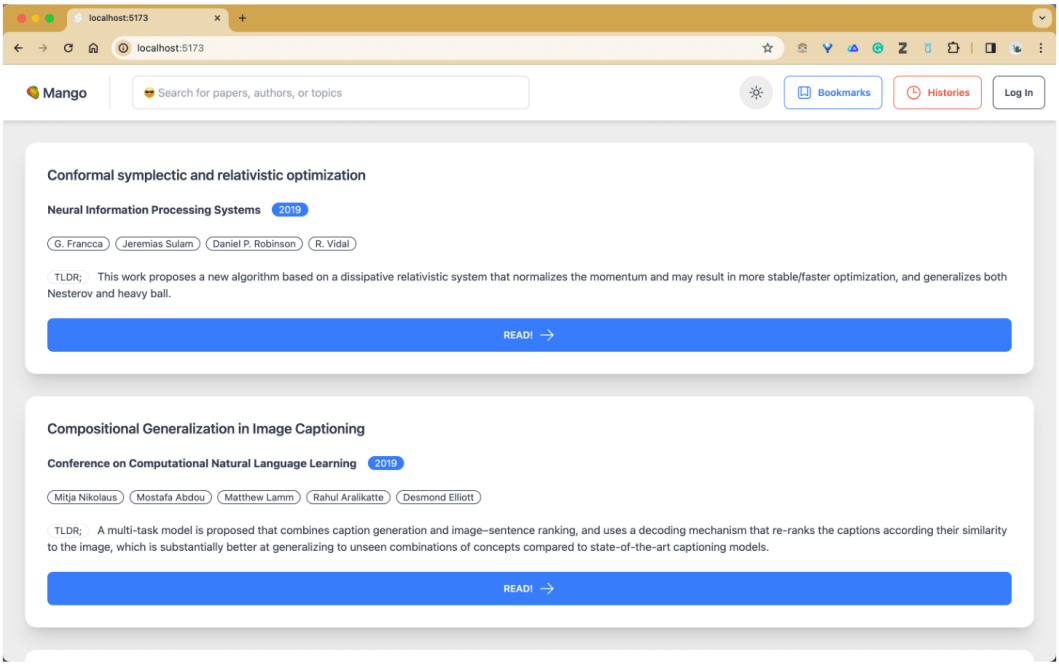
First, please install `node` or `bun` on your device.

Then go to the `Mango-web` folder install all the dependencies of the project with `npm install .` or `bun install .` . Once installed all the dependencies, run the commend to setup mango Web server for the UI -

```
# If using npm
```

```
npm run dev

# If using bun
bun --bun run dev

# ken@Andúril Mango-Web % bun --bun run dev
$ vite dev
▲ [WARNING] Cannot find base config file "./.svelte-
kit/tsconfig.json" [tsconfig.json]

    tsconfig.json:2:12:
      2 │    "extends": "./.svelte-kit/tsconfig.json",
        ╵              ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


Forced re-optimization of dependencies

  VITE v4.4.11  ready in 843 ms

  ➜  Local:   http://localhost:5173/
  ➜  Network: use --host to expose
  ➜  press h to show help
```

Then, open the local url in your browser, you will see the UI shown below and you are good to go!



APPENDIX D – INDIVIDUAL PROJECT REPORT

Li Jiacheng (A0285823W)

Personal Contribution

- **System Design:** I played a crucial role in designing the architecture of our system, ensuring it aligned with our project goals.
- **Data Collection and Preprocessing:** I was responsible for gathering and preprocessing datasets, ensuring data quality and relevance.
- **UI Prototype Design:** I designed the initial prototype of the user interface, laying the foundation for a user-friendly experience.
- **Frontend and Backend Implementation:** I actively contributed to frontend and backend development, bridging the gap between user experience and data processing.
- **Recommendation Component Building:** I led the development of recommendation components, enhancing our system's core functionality.

Key Learning

- **Effective Collaboration:** My experience taught me the value of collaboration and effective teamwork in achieving project success.
- **Dedication & Hard Work:** I learned the importance of dedication and hard work in overcoming challenges and delivering high-quality results.
- **Simplicity with Patience and Persistence:** Keeping things simple, combined with patience and persistence, proved a practical approach to problem-solving and achieving project objectives.

Application of Knowledge and Skills

- **Data Preprocessing:** I applied my expertise in data preprocessing using Pandas and PySpark to ensure data quality and readiness for analysis.
- **Frontend Development:** I built the front end using Sveltekit and Bun, creating an intuitive and responsive user interface.
- **Backend Development:** Leveraging FastAPI, Towhee, and Tensorflow Lite, I developed the backend, enabling efficient data processing and retrieval.
- **Model Development:** I used Tensorflow and Keras to build models, enhancing our system's recommendation capabilities with advanced machine-learning techniques.

Mao Zhihong (A0285799X)

Personal Contribution

- Designed user requirement surveys.
- Conducted competitive market research.
- Participated in the design of the word2vec algorithm.
- Investigated and attempted to implement various search algorithms.
- Contributed minor parts to front-end code development.

What learnt is most useful for you

During my investigation and research, I delved into existing word-to-vector algorithms and their mechanisms, gaining insights into how text is transformed into numerical vectors that computers can process. I learned about various filtering algorithms and how they extract keywords from phrases or sentences, enhancing the relevance and accuracy of information retrieval.

On the front-end side, through page design and back-end routing, I acquired knowledge on connecting the front and back ends. This involved understanding how to handle data sent from the front end using back-end code, ensuring seamless integration and communication between user interface and server-side functions.

Most importantly, I gained a comprehensive understanding of the full design process for deploying a web-based project. This encompassed everything from initial conception to final implementation, providing me with a holistic view of project development and deployment in a real-world setting. This experience not only enhanced my technical skills but also enriched my understanding of the lifecycle of web applications.

Application of Knowledge and Skills

My exploration into word-to-vector algorithms, particularly understanding the intricacies of models like word2vec, laid a foundational knowledge that was instrumental in applying similar concepts in sophisticated vector database systems like Milvus. Through this, I grasped the significance of efficient data indexing and similarity search, which are pivotal for scaling applications in AI and search engines.

The exposure to different filtering algorithms, which are crucial for keyword extraction, paralleled the functionalities of tools like Towhee, a framework that simplifies the process of transforming unstructured data into a structured format. This reinforced my ability to craft algorithms that can sieve through data, a skill that's directly applicable in the creation of efficient data pipelines and enhancing information retrieval processes.

Goh Min Hua (A0285810A)

Personal Contribution

My contributions extend to gathering the feedback of target audience through surveys, data preprocessing, building the recommendation system pipeline, in this case the dual-tower model, as well as report writing.

What learnt is most useful for you

It took some time to overcome the challenges of building a dual tower model recommendation system, to build in Tensorflow was one since I have never learnt how to use it, not to mention building a recommendation system from scratch. It took me a while to get used to how the pipeline should be, what I found is that Tensorflow's guide is not the most complete set of example to follow through, hence I had to do a bit of research to find what other practitioners have come up with to combine the knowledge needed to build the pipeline we dreamed for, in which the results turned out to be quite satisfactory. However,

This brings to the next point of learning time management and communication. We took more time as expected to plan the project and settling with the discord amongst our team members, as well as the distribution of workload amongst ourselves. While it is clear some team members have more coding experience in relevant fields than the rest, we tried to split the workload such that it aligns to our skillsets in order to spend time more efficiently.

Application of Knowledge and Skills

After learning how to code a dual tower model in Tensorflow, I found that to make the whole pipeline able to work in different environments, I had to build the pipeline in a very specific manner which I found the manner of coding of the model to be very inflexible. Perhaps for future considerations, I should research more when it comes to building any types of models, not just recommendation systems, in other frameworks like Pytorch instead, which could be more customizable and flexible.

When we tried to distribute the workload according to our skillsets, this helps me to highlight the shortcomings that I have like the frontend and backend development. It also brought light to me that majority of my peers do not have frontend and backend experience, hence in future, I know what to prioritise to learn first to ensure in my future group projects, we have all the relevant skillsets equipped sufficiently.

REFERENCES

[1] Willyard, Cassandra. “Literature Reviews Made Easy.” Apa.org, Mar. 2012, www.apa.org/gradpsych/2012/03/literature#:~:text=How%20much%20time%3F.

[2] Gusenbauer, Michael. “Google Scholar to Overshadow Them All? Comparing the Sizes of 12 Academic Search Engines and Bibliographic Databases.” Scientometrics, vol. 118, no. 1, 10 Nov. 2018, pp. 177–214, link.springer.com/article/10.1007/s11192-018-2958-5, https://doi.org/10.1007/s11192-018-2958-5.

[3] Shan, Ying et al. “Deep Crossing: Web-Scale Modeling without Manually Crafted Combinatorial Features.” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016): n. pag.

[4] Karpukhin, Vladimir et al. “Dense Passage Retrieval for Open-Domain Question Answering.” ArXiv abs/2004.04906 (2020): n. pag.

[5] Singh, Amanpreet et al. “SciRepEval: A Multi-Format Benchmark for Scientific Document Representations.” ArXiv abs/2211.13308 (2022): n. pag.

[6] Covington, Paul et al. “Deep Neural Networks for YouTube Recommendations.” Proceedings of the 10th ACM Conference on Recommender Systems (2016): n. pag.

[7] “TensorFlow Recommenders.” TensorFlow, www.tensorflow.org/recommenders.

[8] Yi, X., Yang, J., Hong, L., Cheng, D. Z., Heldt, L., Kumthekar, A., Zhao, Z., Wei, L., and Chi, E. (2019). Sampling bias-corrected neural modeling for large corpus item recommendations. In Proceedings of the13thACMConference on Recommender Systems, pages 269–277