SKKU Lab Recommendation Service: FindMyLab

MinSeok Jang, HyunJin Kim, MinSeok Song, and Jaehee Cho

Department of Computer Science and Engineering, Sungkyunkwan University
{msj87190,jin4382,heroicms,rehap}@g.skku.edu

Abstract. The primary goal of designing platform(FindMyLab) is to assist students in pursuing graduate studies. FindMyLab helps students discover research labs that align with their interests through keyword searches and similarity-based recommendations. The platform's backend use GPT prompting to extract keywords from research paper abstracts, which are collected using researcher IDs and Google Custom Search. Subsequently, SBERT is utilized to generate vector embeddings, enabling similarity analysis. The generated embeddings are then used to calculate the similarity between a user's searched keyword and the available data, displaying the top research labs with the highest relevance.

Keywords: Similarity based recommendation, SBERT, GRAD Lab, GPT prompt

1 Introduction

FindMyLab provides information on relevant labs that match users' interests and research preferences based on keyword searches. The graduate school enrollment rate in the Department of Computer Science and Engineering at Sungkyunkwan University, which was 24.1% in 2012, increased significantly to 42.9% in 2022. As a result, the demand for information about pursuing graduate studies has also grown. The first and most important issue students face when pursuing graduate studies is finding a research advisor. However, our team realized that most undergraduate students lack information about research labs. In such situations, students must visit each lab's website or look up the professors' credentials to find a lab that matches their area of interest. Moreover, some labs either lack research performance information or do not have it organized. Due to these problems, the process of finding a suitable lab based on one's interests is very cumbersome for individuals. In these circumstances, Students planning to pursue graduate studies often struggle to find or access information about research labs, which may lead them to join labs unrelated to their interests or abandon their plans for graduate school altogether.
So we propose a service that allows undergraduate students to find related labs through keyword search. Our goal is to 1) Similarity-based research lab recommendations based on the entered keyword, 2) organize the main research topics of the lab, and 3) recommend other labs that do similar research to professor's lab. FindMyLab provides convenience for students in choosing a lab, and it will

be helpful for students who want to go to graduate school but have difficulty finding information by providing additional information about the lab.
The development process can be divided into three parts: AI, backend, and frontend. First, the AI part collected abstracts of the relevant lab papers and extracted key vector embeddings for similarity search using the SBERT model. The lab embedding used for the search was the average of the paper embedding vectors. Using the embedding vector extracted in this way, we recommend the research lab with the highest relevance to the input keyword. To collect these abstracts, we searched the entire list of professors' papers using researcher IDs such as ORCID or Scopus ID, and crawled the abstracts of the papers using the Google custom search engine. The backend part uses PostgreSQL to store embedding and lab-related data. In addition, the database was pre-stored with approximately 2,300 keyword data related to computer science using the GPT prompt, and the top 10 tags most similar to the lab's embeddings were designated as the lab's main research topics. Finally, the front-end uses React to create pages and build tagging functionality.
As a result, the final features implemented in our service are as follows: 1) Keyword search 2) Related lab list - includes professor information, main research topic information 3) Lab page - includes professor information, main research topic relevance, and similar lab information

2 Design for the System

2.1 Overall Dataflow

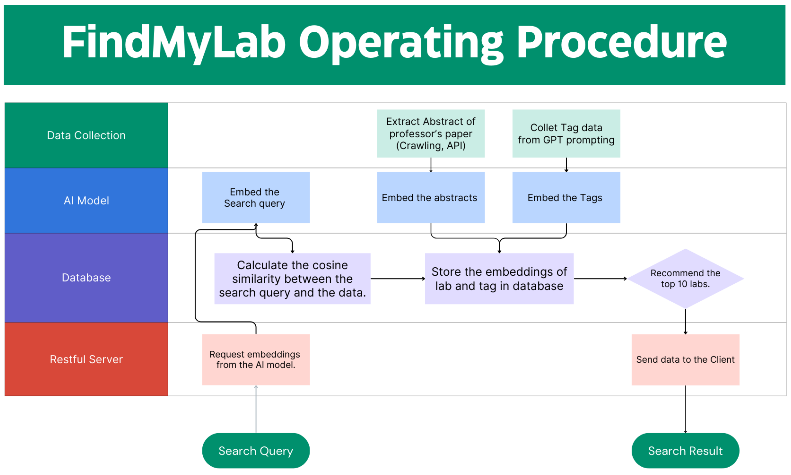

FindMyLab Operating Procedure
Data Collection
AI Model
Database
Restful Server
Extract Abstract of professor's paper (Crawling, API)
Collet Tag data from GPT prompting
Embed the Search query
Embed the abstracts
Embed the Tags
Calculate the cosine similarity between the search query and the data.
Store the embeddings of lab and tag in database
Recommend the top 10 labs.
Request embeddings from the AI model.
Send data to the Client
Search Query
Search Result

Fig. 1. Simple Data Flow

2.2 ML Model

SBERT (Sentence-BERT) is an improved version of BERT designed to handle sentence-level tasks more effectively. The original BERT model is a bidirectional model based on a deep stack of Transformer encoders. Unlike the vanilla Transformer, BERT performs bidirectional attention, enabling the [CLS] token to learn feature embeddings that encapsulate the meaning of the entire sentence. To use BERT in our model for performing tasks such as retrieving sentences or determining relationships between sentences (e.g., queries and target sentences), it operates in the following structure: The output's [CLS] token contains the re-

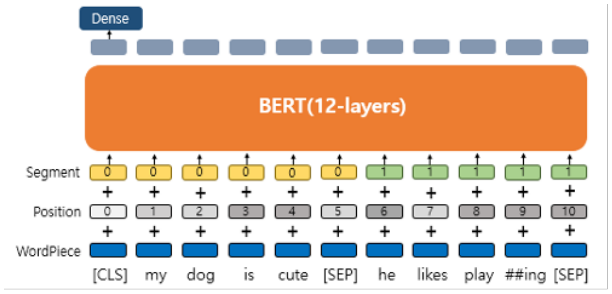


Fig. 2. BERT Model

sult of learning the relationship between two sentences. However, BERT requires both sentences to pass through a single Transformer together. Since BERT classifies the relationship between two sentences by processing them simultaneously, it is too time-consuming to be suitable for search recommendation algorithms. To address this issue, we used SBERT, as shown in the diagram below. This

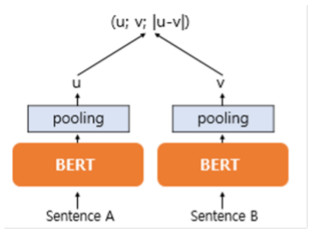


Fig. 3. SBERT Model

model computes embeddings for each sentence separately and then measures the cosine similarity between them. In our program, which precomputes and stores embeddings for paper abstracts, it is highly efficient in saving time and resources.

BGE-m The SBERT-based model used for fine-tuning is the BGE-m model. This model is a large-scale language model that supports over 100 languages. However, since paper abstracts are predominantly in English, we used the smallest model, bge-small-en-v1.5. This model has 384 dimensions, making it the smallest variant, and can process input sequences of up to 512 tokens. According to the model's paper, it uses a self-knowledge distillation technique to distinguish between positive and negative samples. The model is trained using InfoNCE loss, represented by the following formula:

$$L = -\log \frac{\exp\left(s(q,p^*)/\tau\right)}{\sum_{p\in\{p^*,P^-\}} \exp\left(s(q,p)/\tau\right)} \tag{1}$$

Additionally, this model is trained to perform hybrid retrieval by integrating three functionalities: dense retrieval, lexical retrieval, and multi-vector retrieval. In this process, the prediction scores are summed as follows:

$$s_{\text{inter}} \leftarrow s_{\text{dense}} + s_{\text{lex}} + s_{\text{mul}} \tag{2}$$

This score is used as a teacher during distillation, where it is normalized by applying softmax and defined as L$'$. The final loss is then defined by integrating L$'$ and L as L' + L.

Algorithm Using this SBERT model, key vector embeddings are extracted for similarity search to produce results relevant to the query. The abstracts of papers are converted into vector embeddings using the SBERT model and stored in the database. To generate embeddings for professors to be used in the search, the average value of the paper embedding vectors is utilized.

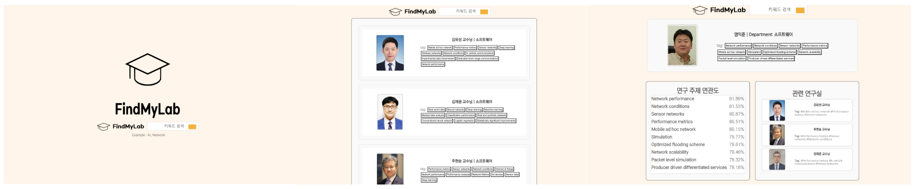
3 Implementation

3.1 UX/UI

Main Page The main page features the service's logo prominently displayed at the center, along with a search bar where users can input relevant keywords. To guide users, example keywords are displayed below the search bar as suggestions.

Result Page Based on the user's input, the results page displays a list of labs matching the search criteria. At the top of this page, a search bar is provided to allow users to input new keywords. Below it, the list of labs appears. Clicking on a specific lab's card navigates to that lab's detailed page. Additionally, clicking on any of the related keywords associated with a lab triggers a new search using those keywords.

Lab Page The Lab Page provides detailed information about the professor, their relevance to related keywords, and their associated research labs. At the top of the page, the professor's name and department tags are displayed, and users can click on these tags to be redirected to a search page filtered by that tag. The relevance between the lab and related keywords is shown in the lower-left section of the page. Additionally, the page includes cards for the top three labs closely connected to the current lab. If a user clicks on a research lab associated with these labs, they are immediately navigated to the respective lab's page.


FindMyLab
FindMyLab
키워드 검색
Example : AI, Network
FindMyLab
키워드 검색
김유성 교수님 | 소프트웨어
tag: Mobile ad hoc network Performance metrics Sensor networks Deep learning Wireless networks Network conditions In vehicle communications Experimental data transmission Dedicated short range communications Network performance
김재광 교수님 | 소프트웨어
tag: Real world data Neural networks Deep learning Machine learning Medical data analysis Classification performance Real and synthetic datasets Convolutional neural network Logistic regression Statistically significant improvements
추현승 교수님 | 소프트웨어
tag: Performance metrics Sensor networks Network conditions Internet of things Network performance Performance analysis Network lifetime Iot devices Sensor data Deep learning
FindMyLab
키워드 검색
염익준 | Department: 소프트웨어
tag: Network performance Network conditions Sensor networks Performance metrics Mobile ad hoc network Simulation Optimized flooding scheme Network scalability Packet level simulation Producer driven differentiated services
관련 연구실
김유성 교수님
Tag: #Mobile ad hoc network #Performance metrics #Sensor networks
추현승 교수님
Tag: #Performance metrics #Sensor networks #Network conditions
정재훈 교수님
Tag: #Performance metrics #In vehicle communications #Sensor networks

연구 주제 연관도

Network performance	81.89%
Network conditions	81.55%
Sensor networks	80.87%
Performance metrics	80.51%
Mobile ad hoc network	80.15%
Simulation	79.77%
Optimized flooding scheme	79.61%
Network scalability	79.46%
Packet level simulation	79.32%
Producer driven differentiated services	79.18%

Fig. 4. Main page, results page, lab page in order from left

3.2 Frontend

Search Page This is the landing page users see when they first visit the service. It features a central logo to convey the purpose of the service and provides a search bar for keyword input. The search bar is implemented using the search module, which is reused on both the Result Page and the Lab Page. To ensure consistency, the module includes not only the search bar but also a logo on the left. Clicking on the logo redirects users back to the Search Page. When users enter a keyword in the search bar and either press Enter or click the button on the right, the keyword is passed to the next page, and they are navigated to the Result Page.

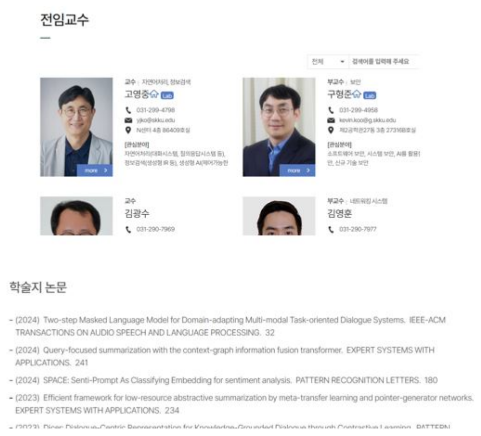
Result Page Based on the user's input, service provides research lab cards as relevant results. The Search module, previously used on the Search page, is positioned at the very top. Below the Search module, we allocate space to store and display research lab cards. The user's input is sent to the backend, and the professor-related information received is passed to the Lablist module, which arranges the information for the user to view. For professor images, the frontend could manage and store the images, but to address issues related to periodic updates, we opted to receive image files from the backend in text format via base64 encoding. When a user selects a research lab card, they are navigated to the corresponding LabPage. Similarly, when the user selects keywords, those keywords are sent to navigate to the Result Page.

Lab Page This page displays information about a research lab, keyword similarities, and related research labs. To achieve this, the functionality is divided into three main modules for better management. The research lab's ID is sent to the backend, which returns data containing keyword similarities and information about related research labs, including their IDs. This data is used to build the page by passing it to the respective modules. In the Related Labs module, the IDs of related labs are sent back to the backend to fetch additional details, which are then displayed on the screen. If a keyword listed in the lab's information is clicked, the user is redirected to the Result Page, showing search results for that keyword. Similarly, if a related research lab card is clicked, the user is navigated to the corresponding Lab Page.

3.3 Data Collection

Abstract Collection To identify the research specialties of professors, we analyzed the abstracts of their published papers. By embedding these abstracts using SBERT and comparing them with the embedding of a search query, we sorted the results based on cosine similarity. To implement this, we first needed to collect the abstracts of professors' papers.

The first resource we referred to for collecting paper data was the professor introduction section on the Sungkyunkwan University website. This section includes a list of professors and representative papers authored by them. Using this information, we identified the titles of the professors' representative papers. We then used Google Custom Search Engine to locate the URIs of these papers published by academic societies.



전임교수

전체 검색어를 입력해 주세요

교수 : 자연어처리, 정보검색
고영중 Lab
031-299-4798
yjko@skku.edu
N센터 4층 86409호실
[관심분야]
자연어처리(대화시스템, 질의응답시스템 등), 정보검색(생성형 IR 등), 생성형 AI(제어가능한
more

부교수 : 보안
구형준 Lab
031-299-4958
kevin.koo@g.skku.edu
제2공학관27동 3층 27316B호실
[관심분야]
소프트웨어 보안, 시스템 보안, AI를 활용한 [illegible] 안, 신규 기술 보안
more

교수
김광수
031-290-7969

부교수 : 네트워킹 시스템
김영훈
031-290-7977

학술지 논문

- (2024) Two-step Masked Language Model for Domain-adapting Multi-modal Task-oriented Dialogue Systems. IEEE-ACM TRANSACTIONS ON AUDIO SPEECH AND LANGUAGE PROCESSING. 32
- (2024) Query-focused summarization with the context-graph information fusion transformer. EXPERT SYSTEMS WITH APPLICATIONS. 241
- (2024) SPACE: Senti-Prompt As Classifying Embedding for sentiment analysis. PATTERN RECOGNITION LETTERS. 180
- (2023) Efficient framework for low-resource abstractive summarization by meta-transfer learning and pointer-generator networks. EXPERT SYSTEMS WITH APPLICATIONS. 234
- (2023) Dicer: Dialogue-Centric Representation for Knowledge-Grounded Dialogue through Contrastive Learning. PATTERN

Fig. 5. List of Professors at Sungkyunkwan University

Next, we performed two tasks using the collected URIs. First, we crawled the abstracts of the papers available at these URIs. This approach was particularly useful for professors who do not manage researcher IDs. Second, if researcher IDs such as ORCID or Scopus ID were available in the society's database, we collected these IDs and used APIs to retrieve the complete list of the professors' papers.

Tag Collection Simply displaying the professor's name and photo as a search result does not convey how relevant the professor is to the query or the general topics covered in their papers. To address this, we introduced a tag system. Tags are short keywords that represent the areas most closely related to the professor's embedding, independent of the search query. We pre-generated a list of tags associated with various fields, compared them with the professor's embedding, and assigned the top 10 tags based on cosine similarity.

For this service, pre-defining tags for all fields was necessary. However, manually preparing these tags posed challenges due to the vast number of specialized fields and the maintenance burden in accommodating emerging technologies and fields. To overcome this, we developed an automated tag collection process. We utilized GPT prompt engineering to generate tags tailored to each paper. The prompt used was: *"Read the abstract of the provided research paper and return the field and classification as a single, combined list of keywords in array format, without any headings or categorization."* Using this approach, we amassed a large-scale dataset of tags suited to each abstract.

However, the collected tag data often contained numerous synonyms and similar terms. Using this data as-is led to issues such as multiple tags with the same meaning being assigned. For instance, terms like "Database Manage," "Database Management," and "Database Management System" often appeared redundantly.

To resolve this and refine the tag data, we clustered the tags using the DBSCAN process, setting the epsilon (`eps`) value to 0.1. After clustering, we used the representative word and filtered out noise to generate clean tag data. The representative word for each cluster was chosen based on the most frequently occurring term. The figure below illustrates the clustering results for tags related to the Department of Software. Through clustering, we reduced 3,500 tag data points to 2,300.

Fig. 6. Clustering Results for Tags

3.4 Backend

Database PostgreSQL was used to store embedding and lab-related data. PostgreSQL 16 supports embedding data storage in vector format through the `pgvector` extension and facilitates cosine distance calculations between vectors. This makes it highly suitable for embedding storage and management, and it is natively supported by AWS RDS.

Column Name	Data Type	Constraints	Description
labid	SERIAL	PRIMARY KEY	Unique identifier for the lab
professor_img	TEXT		Base64 to professor's image
name	VARCHAR(255)	NOT NULL	Name of the professor
dept	VARCHAR(255)	NOT NULL	Department the lab belongs to
embedding	VECTOR(384)		384-dimensional embedding vector

Table 1. lab Table

Column Name	Data Type	Constraints	Description
tag	VARCHAR(255)	PRIMARY KEY	Tag or keyword
embedding	VECTOR(384)		384-dimensional embedding vector

Table 2. tags Table

Server The server architecture includes a backend server handling API operations, user endpoint devices, and a server running AI models. The backend server performing API tasks was deployed on AWS EC2, using a `t4g.micro` instance with 2 vCPUs, 1 GiB of memory, and up to 5 Gbps network burst bandwidth.

The AI model was run on a desktop server equipped with an RTX 4070 Ti GPU, 16 GB memory, and 8 CPUs. Initially, embedding computations were executed using child processes on the main server. However, this approach led to over 5 seconds of delay during searches, primarily due to the low server performance and the need to reload the model for each embedding calculation.

To resolve this, we transitioned to a dedicated desktop server where the model remained preloaded. Search queries were processed by computing embeddings and transmitting them over the network, reducing delay to under 100 ms. Additionally, when replacing the AI model, the server previously required downtime and code changes. We improved this process by downloading the model's checkpoint in advance on the desktop server, allowing the model to reboot independently. This ensured a replacement time of under 10 seconds.

AI Model The AI model was fine-tuned using the FlagModel. The fine-tuning dataset comprised abstracts collected earlier and tags extracted from these abstracts using GPT prompt engineering, which served as data and labels, respectively. This setup provided a flexible environment for collecting training data as new disciplines emerged, facilitating seamless extension of supported academic fields. The fine-tuned checkpoint was uploaded to the desktop server for model loading.

3.5 ML

Data The dataset collected through web scraping is stored in a CSV file in the following format.

For fine-tuning the model, this data is converted into a JSON file. The keyword column is treated as the expected input query and assigned as the value

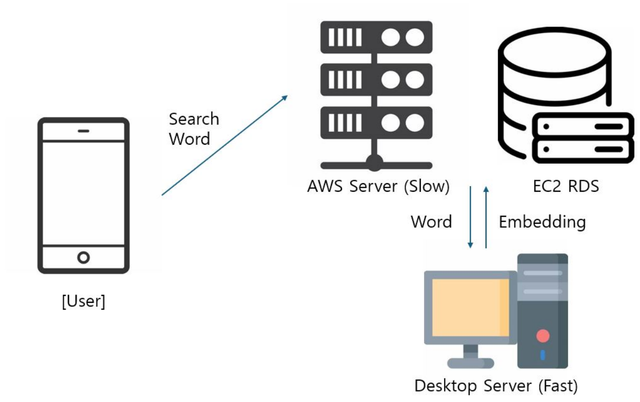


Fig. 7. Server Architecture

prof name	**abstract**	**keywords**

Table 3. dataset CSV file

for the query key, while the corresponding abstract is used as the value for the pos key. Since the maximum sentence length the model can accept is 512 tokens, abstracts exceeding this length are divided into multiple chunks of 512 tokens each. These are used to create multiple pairs in the format of (query A, pos1), (query A, pos2) for a single query. This approach minimizes information loss from the abstracts. As a result, the initial dataset of 5,000 rows was expanded to 18,536 rows.

negative mining For NCE loss learning (contrastive learning), negative data is required. However, as the current dataset only contains positive data, hard negative mining was performed using FlagEmbedding. The difficulty of "hard" negatives was controlled by adjusting the "range for sampling" parameter. This involved using a pretrained retrieval model to search for the top k documents most similar to the query and then sampling negative examples within the specified parameter range. This method selects incorrect but similar answers to help the model learn effectively. We selected 10 negatives within the range of 15–200. The range of 15–200 was chosen because our dataset has approximately three positive examples per query. Therefore, a value greater than 3 was selected to ensure flexibility.

4 Evaluation

As previously mentioned, the model used(bge-small-en-v1.5) is the smallest model from the BGE-m3 series developed by BAAI, featuring 384 dimensions and a sequence length capacity of up to 512 tokens. The input is crawled data file containing professor name, abstracts and the keywords and the output would be the embedding of professor or keywords which are used to rank and find the professor. During fine-tuning for our use case, we set the batch size to 32 and the number of epochs to 14. Training beyond this point did not reduce the training loss further and raised concerns of overfitting. Additionally, while experimenting with batch sizes of 4, 16, and 64, the batch size of 32 yielded the best performance due to the limitations of the dataset size. The learning rate was set to 1e-5, and all other parameters followed the default settings of the baseline model.

evaluation metrics We used mAP (Mean Average Precision),which takes the order of recommendations into account to quantitatively evaluate our model. The formula for mAP is as follows:

$$MAP@K = \frac{1}{|U|}\sum_{u=1}^{|U|}(AP@K)_u$$

Where U represents the number of users, and the AP value is calculated for each user. The mean Average Precision (mAP) is obtained by dividing the sum of AP values by the number of users, providing the average precision across all users. Below is a comparison with the baseline model:

K	**Finetune**	**Base**
3	0.48	0.45
5	0.51	0.48
7	0.52	0.49

Table 4. Comparison of Finetune and Base Models

Empirical results It can be observed that our model outperforms the baseline model. Not only does it show superior performance quantitatively, but qualitative evaluations also demonstrate improved effectiveness.

Rank	**Professor**	**Similarity**
1	구형준	0.7517
2	이지형	0.7372
3	우홍욱	0.7202
4	이은석	0.7200
5	우사이먼성일	0.7151

Table 5. baseline model

Rank	**Professor**	**Similarity**
1	구형준	0.9142
2	우사이먼성일	0.8734
3	이은석	0.8705
4	김재광	0.8604
5	이지형	0.8475

Table 6. our

As you can see on the table 5. and table 6., when testing a keyword that is expected to return Professor Hyung-Jun Koo as the result, we observed that the gap in similarity scores between Professor Koo and other professors in the Top 5 increased significantly. Additionally, even among lower-ranked professors, the model successfully retrieved names of professors conducting research in similar fields (e.g., cybersecurity). This demonstrates that our project has successfully implemented a search function specialized for our university, aligning with the goal of enabling efficient searches for professors and labs within our institution.

objectiveness However, due to time constraints, we were unable to conduct extensive user testing with students to verify whether the results align with their actual expectations, which remains a point of regret.

5 Limitations and Discussions

AI Our project was limited to targeting the Department of Computer Science and Engineering at Sungkyunkwan University as the primary audience for our service, and this served as the biggest constraint. The first aspect of this limitation is that, in reality, students who plan to pursue graduate studies do not exclusively consider Sungkyunkwan University. While the Department of Computer Science and Engineering at Sungkyunkwan University has many professors, it does not cover all subfields comprehensively. Moreover, comparing information about professors from other universities would provide more useful information to users of our service. The second aspect is that students do not only consider the Department of Computer Science and Engineering when applying for graduate school. In fact, all of our team members are double majoring, and many students in the Department of Computer Science and Engineering at Sungkyunkwan University are double majors. This implies that they might seek information about professors from other departments related to computer science or AI. However, due to practical limitations, our service could not provide research lab information for all relevant departments.

Evaluation The next limitation of our project lies in the evaluation method. We built a model using SBERT to measure cosine similarity with keywords and evaluated it quantitatively using mAP@K. However, this merely represents a mechanical evaluation of similarity within the AI model. It does not necessarily indicate how useful or relevant the information is from the practical user perspective. While we conducted subjective evaluations using some test cases, these lacked objectivity, leaving us without concrete information on how useful real users would find our service.

App Lastly, another limitation of our project is the absence of various features for user convenience. Our project focused solely on keyword-based research lab searches and providing information about similar labs. However, students planning to pursue graduate studies look for more than just information about research labs. Our service does not provide other features, which we believe is an area that needs improvement from a user convenience perspective.

6 Related Work

There are various services available to help college students find laboratories or professors.
PhD.Kim.net was launched as a platform to provide essential information for selecting an academic advisor when applying for graduate programs. Its most popular features included the ability to search for laboratories and write laboratory reviews.
RateMyProfessors.com is a website where students can rate professors and campuses from institutions in the U.S., Canada, and the U.K. It is the largest professor rating site, covering over 8,000 schools, 1.7 million professors, and more than 19 million ratings. Professors are evaluated on a 1-5 scale in categories such as overall quality and difficulty level.

7 Conclusion

Our team found the problem of difficulty in finding a lab based on one's field of interest when entering graduate school, and planned and developed the FindMyLab service to solve that problem. We expected that finding a lab that studies one's field of interest, rather than fitting one's field of interest into the research field introduced by the lab, would increase the accessibility of graduate school. So as an alternative, we proposed a keyword-based lab search service using GPT prompts and AI models. In the process, we developed the service using tools such as React and PostgreSQL and AI models such as SBERT. The final form of the service was a web service that students could easily search.
As mentioned in the Limitations and Discussions, there are still tasks remaining, such as expanding to other departments and other schools and adding various user-friendly features. However, we expect that our project will play a positive role in increasing students' interest in lab information so that they do not end up doing research they are not interested in or give up on entering graduate school because they cannot find a lab that fits their field of interest.

References

1. Yongwoo Kim, Daeyoung Kim, Hyunhee Seo, Young-Min Kim. *Content-based Korean journal recommendation system using Sentence BERT*. Journal of Intelligence and Information Systems, Volume 29 Issue 3, Pages.37–55. https://doi.org/10.13088/jiis.2023.29.3.037
2. 빈이름. (2023, March 20). *Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks*. 의미있는블로그. https://all-the-meaning.tistory.com/9
3. *LLM+추천시스템 Large Language Models Meet Collaborative Filtering: An EfficientAll-Round LLM-Based Recommender System*. (2024, July 19). Anything. https://wigo.tistory.com/entry/LLM
4. J, H. (2023, May 6). *[SBERT] 키워드 추출 기반 유사 메뉴 검색 서비스*. 's Tory. https://hjkim5004.tistory.com/122
5. 테디노트. (2024, August 17). *02. FAISS*. <랭체인LangChain 노트> - LangChain 한국어 튜토리얼. https://wikidocs.net/234014