SKKU Lab Recommendation Service: FindMyLab

MinSeok Jang, HyunJin Kim, MinSeok Song, and JaeHee Jo

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

Abstract. This document is a proposal for the development of a keyword search-based lab recommendation platform. The main purpose of designing this platform is to assist students in pursuing graduate studies and reduce situations where research performance declines due to studying topics they are not interested in. This service uses GPT-based prompts, the SBERT model, and Mean Average Precision@K to recommend labs that align with users' interests and research preferences.

Keywords: GRAD Lab, Similarity based recommendation, GPT prompt, SBERT, mAP@K

1 Introduction

Over the past decade, the graduate school enrollment rate among students in the Department of Computer Science and Engineering at Sungkyunkwan University has significantly increased. However, one of the challenges of pursuing graduate studies is that students find it cumbersome to locate professors who research their area of interest. As a result, a significant amount of time and effort is wasted in the process of searching for labs that align with their interests. These issues stem from the lack of a keyword-based lab recommendation feature.

Our team's service aims to lower the barriers for undergraduate students who only know their fields of interest by making the process of pursuing graduate studies more accessible. To achieve this, our service offers the following key features. First, we provide a lab page for each lab, introducing the professors and their research groups. Second, we offer recommendations for related labs based on keyword searches of research fields, along with links to their lab pages. To implement these features, we use the following core technologies: First, we extract key terms from major papers through GPT-based prompts. Second, we use the SBERT model to extract key vector embeddings for similarity search. Third, we recommend labs that are most relevant to the searched keywords using Mean Average Precision@K.

2 Motivation

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. Therefore, our team identified that the current method of finding relevant labs is time-consuming and inefficient. We investigated existing services that provide information about labs and professors for students considering graduate school. However, these services primarily allow users to search for professors to gather lab-related information, which is similar to the department's website, posing a significant limitation. Furthermore, the information provided is often subjective evaluations of professors and labs, rather than being based on objective statistics and data about research fields and achievements, making it less helpful for students trying to find a lab that matches their interests. This situation can have the following consequences.

First, the asymmetry of lab information may lead to outstanding labs not being sufficiently promoted or students not being able to select the right lab. Second, existing services that rely on subjective evaluations of labs and professors increase the likelihood that students will choose a lab based on impressions or reputation, which could negatively impact research suitability and performance in the long term. If these issues are not addressed, students may face difficulties in deciding on graduate school, and research outcomes may suffer due to poor lab selection.

To prevent these problems and enhance research performance and satisfaction after entering graduate school, we propose a service that recommends labs tailored to each student's interests and research tendencies.

3 Problem Statement

During our university life, there are many instances where we need to contact professors. However, when it comes to applying for a lab or asking a professor to supervise a research paper, it often requires significant effort to find out which professor to approach and what their research field is.

3.1 Difficulty in Understanding Professors' Research Areas

Currently, the university website provides brief information about professors' research areas and academic papers. However, this information is not available for all professors and is often limited to brief introductions rather than detailed explanations. This lack of comprehensive information negatively affects students' ability to understand professors' research areas accurately.

3.2 Limited Access to Professors Outside of One's Department

When students look for a supervisor or lab in a specific field, they tend to consider professors within their own department. However, with the increasing importance of interdisciplinary topics, it is becoming more common for related labs to be located in other departments. Yet, it is challenging to identify and approach these labs due to the difficulty in accessing information about research activities in other departments.

3.3 Need for Specific Research Area Information Beyond General Topics

Even when comprehensive introductions to labs are available, there is still a challenge. These introductions are often general, lacking detailed information on specific research subfields. When writing papers or conducting research, it is crucial to understand specific sub-branches of a topic, but the current resources make it difficult to identify these nuances.

4 Proposed Solution

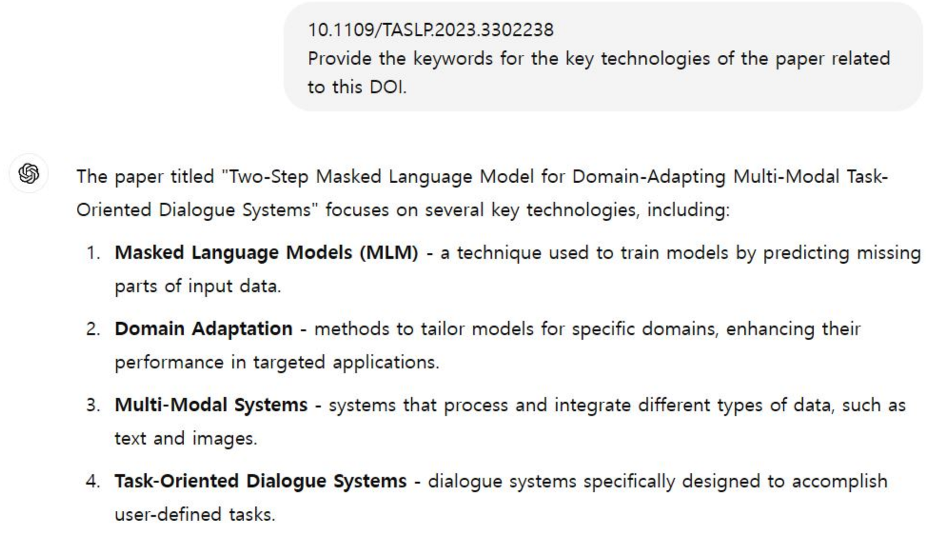
To effectively identify the research areas of a professor, we plan to develop a program that analyzes the professor's published papers. This program will be offered through a website, providing keywords related to the professor's research area. When users search for a specific research area, the program will list professors who primarily study that field. Selecting a professor will provide information such as the proportion of relevant publications and a list of other professors conducting similar research.

The process of extracting a professor's research area from their papers involves three main components: Crossref API, GPT prompting, and a search method.

4.1 Main Components

Web Crawling API We will first obtain the professor's ORCID to secure the DOI of their papers via the Crossref API, which provides access to various metadata, including DOIs for research publications.

GPT-4 Prompt Using GPT-4 prompting, we will extract key technologies from the abstract of each paper. These technologies will be stored in a database and registered as research topics for the professor. We will also count how many papers have been published on each topic to calculate the frequency of research topics.



10.1109/TASLP.2023.3302238
Provide the keywords for the key technologies of the paper related to this DOI.

The paper titled "Two-Step Masked Language Model for Domain-Adapting Multi-Modal Task-Oriented Dialogue Systems" focuses on several key technologies, including:

1. **Masked Language Models (MLM)** - a technique used to train models by predicting missing parts of input data.
2. **Domain Adaptation** - methods to tailor models for specific domains, enhancing their performance in targeted applications.
3. **Multi-Modal Systems** - systems that process and integrate different types of data, such as text and images.
4. **Task-Oriented Dialogue Systems** - dialogue systems specifically designed to accomplish user-defined tasks.

Fig. 1. Core keywords of technologies are extracted through DOI using GPT prompting.
OpenAI. ChatGPT: A language model. Available at: https://www.openai.com/chatgpt (Accessed: 2024-10-02).

Search Method As a core technology of this project, we will fine-tune a SBERT model to build a search system that integrates technical synonyms and related terms. This will allow us to accurately determine how frequently the professor researches specific technologies. Additionally, we plan to enable the retrieval of related higher and lower-level terms using cosine similarity.

4.2 Areas for Improvement

Dependence on ORCID This project is based on ORCID, a widely used researcher and contributor identifier. While many professors use ORCID, the program may not function properly for those who do not use or update their ORCID profiles. The Google Scholar API can be used to partially supplement this, but since it operates through crawling, it may result in a time disadvantage.

Challenges with GPT-4 Prompting Feedback Although GPT-4 prompting demonstrates powerful capabilities, the difficulty in receiving user feedback can lead to decreased reliability of results. To address this, we will implement an algorithm to review and potentially reject keywords that significantly deviate from existing keywords or the professor's broader research areas when storing data in the database. Additionally, we will refine the data through user feedback and ensure that the prompt design avoids generating false information for DOIs that cannot be extracted.

5 Related Work

There are various services available to help college students find laboratories or professors. After excluding platforms that restrict access to certain authenticated users, the most commonly used services were identified, both domestically and internationally. It is characterized by the fact that points are given for evaluation at the same time as the purpose of information delivery.

5.1 PhD.Kim.net

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. Reviews were based on five key evaluation factors: lab atmosphere, teaching ability, thesis guidance, actual wages, and personality. Given the nature of these evaluation criteria, which cover private and sensitive aspects, many users relied on the platform to gain insights that official sources might not provide. The site also hosted a community forum where frank and sometimes harsh evaluations were shared. However, this led to issues, such as criticisms of academic backgrounds and majors, creating a contentious environment. Interestingly, the site faced a defamation lawsuit from a professor over its review feature, resulting in the temporary suspension of the review function. The platform is now undergoing changes, focusing on enhancing features geared more toward professors and researchers.

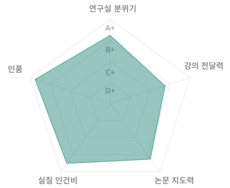
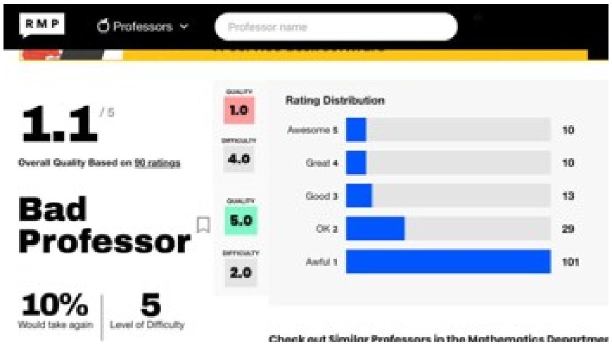


Fig. 2. Professor Evaluation Indicators in PhD.Kimnet

5.2 Rate My Professors

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. Also, there was a lawsuit filed against the platform as an online content provider and the lawsuit was unsuccessful. Additionally, since users are not required to create an account or log in to post reviews, there is no way to verify whether the reviews were written by actual students who took the course. It is also important to note that the site focuses on rating undergraduate courses based on teaching ability, rather than evaluating professors' research labs.



1.1 /5

Overall Quality Based on 90 ratings

Bad Professor

10% Would take again

5 Level of Difficulty

Check out Similar Professors in the Mathematics Departmer

Fig. 3. Evaluation Table shown at RateMyProfessors.com

6 Background

6.1 Prompt Engineering

Technology Prompt engineering refers to the process of guiding generative AI solutions to produce the desired output. In generative AI, responses vary depending on factors such as word choice, structure, and syntax, so it's necessary to use creativity and go through trial and error to create effective input prompts. Through this process of prompt engineering, efficiency and effectiveness can be improved, and control over the output can be strengthened.

In this Project, GPT will be used. OpenAI, the developer of GPT, has introduced six strategies to achieve better results in the prompting process: Write clear instructions, Provide reference text, Split complex tasks into simpler subtasks, Give the model time to "think", Use external tools, and Test changes systematically.

Goal To obtain information about a professor's research papers, we need to extract the professor's unique ID and key research paper keywords using their name, department, and paper titles. To achieve this, GPT will be pre-trained, and various types of information will be provided. This will allow us to break down the complex process into several subtasks, making it easier to acquire the necessary data.

6.2 Overall Algorithm

Our program scrapes the web and, based on ORCID, extracts keywords and embedding vectors from the abstracts of all the papers that professors have participated in. Keywords are extracted using a GPT-4-based pretrained model through prompting, while embedding vectors are extracted using SentenceBERT (SBERT). Since the goal is to search for professors and labs, the embedding vectors of the extracted abstracts are used to calculate a comprehensive embedding vector for each professor's research, which is then stored in the professor table and the paper table. A similarity calculation is performed based on the embeddings of the search query and the professors' embeddings, and the professor most relevant to the search query is presented to the user. The library Faiss, which is discussed in detail in Section 6.4, can be used for this process.

To verify that the model performs the recommendation function correctly, we use mAP (mean Average Precision) metrics to evaluate the model. This is discussed in detail in Section 6.5.

Dataset The shape of the dataset after preprocessing is structured as follows:

Column Name	Data Type	Description
professor_id	INT	prefessor ID
name	TEXT	prefessor Name
department	TEXT	department
lab_name	TEXT	labs
`research_area`	TEXT[]	interst of research (key workds extracted from paper abstract)
`research_embedding`	VECTOR	comprehensive embedding vector of the professor's research

Table 1. Professors Table

Column Name	Data Type	Description
paper_id	INT	paper ID
professor_id	INT	prefessor ID (Foreign Key)
abstract	TEXT	paper abstract
title	TEXT	paper title
keywords	TEXT[]	key word
embedding	VECTOR	embedding vectors from paper abstract

Table 2. Papers Table

Goal Our goal is to use the SBERT model to extract key vector embeddings for similarity search, enabling us to deliver relevant results based on search queries. Specifically, we convert the abstract of each paper into vector embeddings using the SBERT model and store them in the database. The average of these paper embedding vectors is stored as the embedding representing the professor to whom the papers belong.

Later, when a search query is entered, these embedding vectors stored in the database will be used to calculate similarity scores, recommending relevant professors or labs, or suggesting labs conducting similar research. For example, when a user searches for the keyword "computer vision," the system will calculate the similarity based on the stored `research_embedding` values and recommend the professor with the highest similarity score.

Additionally, based on the embedding of a professor whose lab the user is currently interested in, the system can recommend other professors' labs with high similarity scores.

6.3 Model

As previously explained, we use SBERT to extract embeddings from the abstracts.

BERT The structure of the original BERT model is as follows (Figure 4).

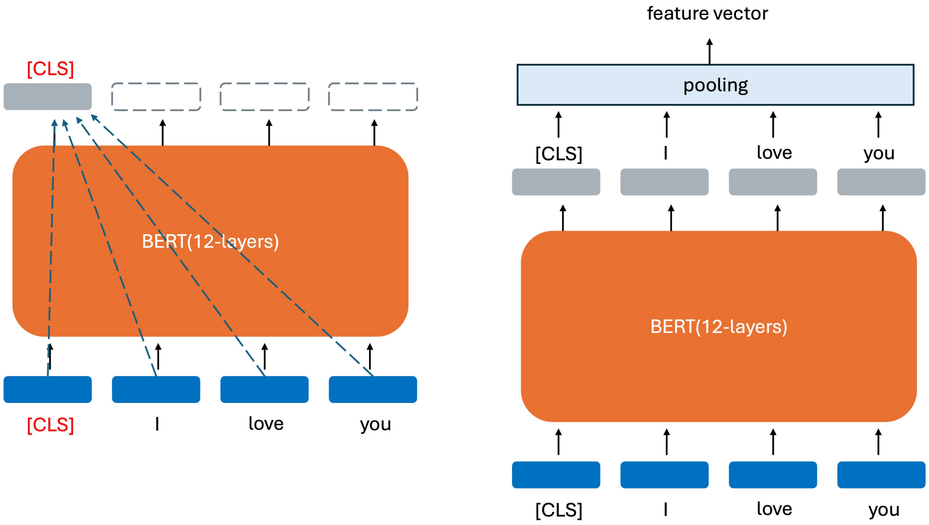


Fig. 4. BERT

BERT is a bidirectional model that stacks multiple layers of the original transformer's encoder structure. In the original transformer, the attention process, which mixes the information between tokens, was only applied to previous tokens relative to the current token. However, in BERT, this attention is applied bidirectionally, including future tokens, allowing the [CLS] token to learn a feature embedding that contains the meaning of all tokens in the sentence. The arrows in the left diagram represent this concept.

In this process, instead of using only a single CLS token as the sentence embedding, a pooling layer can be added to use the average of the output as the sentence embedding as the right diagram.

When performing tasks that require understanding the relationship between two sentences, BERT operates in a structure like the one shown below (Figure 5). In this case, the [SEP] token is included to perform segment embedding. The [SEP] token, placed at the end of each sentence, distinguishes the two sentences and enables segment embedding to represent which sentence each word token belongs to.

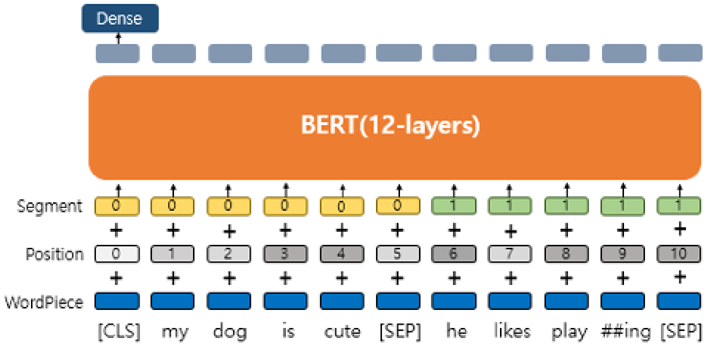


Fig. 5. BERT performing sentence understanding tasks

In the original BERT, two input sentences or different types of paragraphs are passed together through a single Transformer. In this case, the [CLS] token learns the relationship between the two sentences.

SBERT The structure of the model we intend to use in our program, SBERT, is as follows. The structure of SBERT, the model we intend to use in our program, adds a pooling layer to the output of the original BERT model to utilize the extracted embeddings. Unlike BERT, which passes two sentences through a single Transformer, SBERT processes each sentence separately through the Transformer.

SBERT is also efficient in terms of training time because it only requires fine-tuning on a pretrained BERT model. The fine-tuning is performed using the NLI (Natural Language Inference) and STS (Semantic Textual Similarity) tasks. The STS task is measuring the similarity between two sentences by calculating the cosine similarity between the sentence embeddings after the pooling operation. While this approach alone provides powerful learning, the authors also

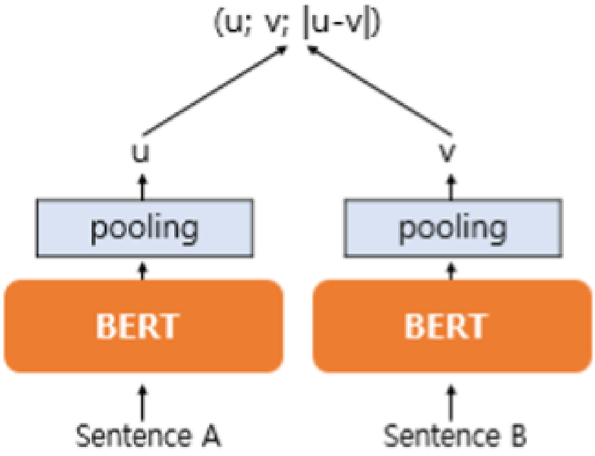


Fig. 6. SBERT

proposed "continued learning," where the model is first fine-tuned on the NLI task and then further trained on the STS task. The NLI task involves learning relationships between sentences, classifying them into entailment, neutral, or contradiction categories.

SBERT has achieved state-of-the-art (SOTA) results across various benchmark datasets and NLP downstream tasks. For example, it excelled on the challenging argument similarity dataset, where it had to distinguish between sentences on different topics within Wikipedia articles. The model's high accuracy on long documents like Wikipedia makes it particularly suitable for our program, where we aim to compare paper abstracts and sentences.

Although SBERT is a language model that handles sequences, it is based on the Transformer architecture rather than RNN-based models like LSTM or GRU. This is because LSTMs and GRUs involve sequential computations, which slow down processing speeds, and they struggle with long-range dependencies, making them less robust for handling long information such as abstracts.

We chose SBERT over BERT because SBERT demonstrated better performance on the STS task. In this task, BERT processes both sentences together and classifies the relationship between them, which takes too much time, making it unsuitable for a recommendation search algorithm. In contrast, SBERT computes the embeddings of two sentences separately and simply calculates the cosine similarity between them. This makes SBERT a better fit for our program, where we precompute and store embeddings of paper abstracts, thereby saving time and resources.

With the extracted vector embeddings, we can perform similarity searches. While we could calculate Euclidean distance or cosine similarity between vectors projected into the same embedding space, we must consider optimizing the cost of these operations, given that our program handles a large volume of data.

6.4 Searching Method

To ensure an efficient search function, we can use the "Faiss" library distributed by Facebook. By using this library, it is possible to perform filtering functions through the filter parameter with custom functions, and caching frequently used queries could help maintain fast search speeds. Additionally, Faiss supports the addition, deletion, saving, and loading of vector data, making it easier to maintain and manage the data.

However, when using Faiss, the quality of the vector values extracted from the embedding model is crucial, so a detailed preprocessing process is essential.

6.5 Metrics

For evaluation metrics such as Precision@K or Recall@K, the number of items of interest to the user among the top K recommended items is estimated. However, since the order of recommendations is likely to be important in our model, we will use Mean Average Precision @ K (mAP) as the evaluation metric, which takes the order of recommendations into account.

The formula for Average Precision (AP) is as follows:

$$AP@K = \frac{1}{m}\sum_{i=1}^{K} Precision@i \cdot rel(i)$$

Precision@i is the ratio of items the user is interested in, considering only up to the i-th position among the k recommended items. $rel(i)$ indicates whether the user liked the i-th item. Multiplying these two factors reflects the influence of the i-th item. m is the total number of items the user liked among all items.

The formula for the overall 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.

7 Planning in Detail

7.1 Role Distribution

The project is largely divided into two categories: AI and web development. Web includes front-end and back-end. AI includes data collection, data preprocessing and model implementation. We decided our main roles based on the interest and competence of each team member. Although each has their own part, it will be flexible if needed during the project.

Name	Rule
Minseok Jang	Front-end, data collection
Hyunjin Kim	Front-end, data collection
Minseok Song	Back-end, data preprocessing
Jaehee Cho	AI, data preprocessing

Table 3. Main Roles

7.2 Development Plan

Based on the roles that were largely divided into two categories, we planned the project schedule by dividing it into front-end, back-end, and AI parts. Although it is the schedule planned, it is aware some might change during the project depending on situation and progress.

	weekly plan					
Week	2 3 4	5	6 7 8	9	10 11 12 13	14 15
Overall	Project Proposal	Proposal LaTeX		Midterm Presentation		Final Presentation
AI			data collection data preprocessing		Model implementation	Beta Test
Web			UI/UX design Database setting		Front-end and Back-end work	Debugging

Table 4. weekly project schedule

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
6. *Prompt Engineering*. (n.d.). OpenAI Developer Platform. https://platform.openai.com/docs/guides/prompt-engineering
7. 한국교육개발원. (n.d.). *학교/학과별 데이터셋*. 교육통계사이트. https://kess.kedi.re.kr/index