

# LangChain and RAG-Based Q&A System for University Policies

*In-Hye Park, Min-Jeong Kim, and Kyung-Ae Cha<sup>\*</sup>*

Dept. of Artificial Intelligence, Daegu University, Gyeongsan-si, Korea

**Abstract.** Recent advancements in language models have led to their widespread application across various fields, achieving remarkable success. However, these models often exhibit a phenomenon known as ‘hallucination’, where they generate responses that present false information as factual. This issue has emerged as a critical challenge, particularly in scenarios where reliability and accuracy are paramount. To address this problem, the RAG (Retrieval-Augmented Generation) technique has gained significant attention. RAG combines generative language models with information retrieval systems, allowing the model to search for relevant information from external databases before generating a response. This approach helps mitigate hallucinations and delivers more trustworthy and accurate information.

This study leverages the LangChain framework to develop a RAG-based Q&A system using university policies and administrative regulations as the dataset, which serves as the foundation for students’ academic and institutional operations. By systematically organizing and managing complex data, the proposed system aims to provide accurate and reliable responses, offering users valuable and actionable information efficiently.

**Keywords;** Large Language Models (LLMs), Retrieval-Augmented Generation, LangChain

---

<sup>\*</sup> Corresponding author: [chaka@daegu.ac.kr](mailto:chaka@daegu.ac.kr)

Received: Nov. 2. 2024 Accepted: Jan. 12. 2025 Published: Mar. 31. 2025

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

---

Cite this paper as : In-Hye Park, Min-Jeong Kim, and Kyung-Ae Cha (2025) “LangChain and RAG-Based Q&A System for University Policies”, Journal of Industrial Information Technology and Application, Vol. 9. No. 1, pp. 1039 - 1050

---

## 1. Introduction

The recent advancements in Large Language Models (LLMs) have introduced new possibilities in the field of Natural Language Processing (NLP), establishing themselves as innovative tools across various domains[1]. Notably, state-of-the-art models such as OpenAI's GPT-4 leverage vast training datasets to provide advanced capabilities in language understanding and generation. These models have been widely applied in areas such as conversational AI, content creation, translation, and question-answering systems[2, 3].

However, large-scale language models are often limited by the phenomenon of hallucination[4]. Hallucination refers to the generation of nonexistent or inaccurate information by the model, which undermines the reliability of its responses. This issue has become particularly critical in fields where trustworthy information is essential.

To address this challenge, the Retrieval-Augmented Generation (RAG) technique has gained significant attention in recent years[5]. RAG combines generative language models with information retrieval systems, allowing the model to search external databases for relevant information before generating responses. This approach improves the reliability of model outputs while ensuring that responses reflect the most up-to-date information.

In this study, the RAG technique is employed to design and implement a question-answering system based on university regulations and administrative information.

## 2. Related research content

LangChain is an open-source framework designed to facilitate the use of large language models (LLMs), and various studies have been proposed that combine its information retrieval and generation capabilities. Choi Jongmyung et al. (2024) developed a KTAS(Korean Triage and Acuity Scale) determination system using LangChain and GPT-4[6]. This study demonstrated an effective use case of Retrieval-Augmented Generation (RAG) technology by integrating voice and text data to automatically classify patient severity in emergency medical settings. LangChain was utilized to search external data and integrate with GPT-4, delivering rapid and reliable emergency care guidelines.

So Hoon et al. (2024) conducted a study to improve the interaction with NPCs in the game system by combining LangChain and Unreal Engine[7]. The system was designed to provide real-time, contextual responses to user questions, and the combination of the vector database and LangChain enabled NPCs to have more natural and personalized conversations.

Additionally, Namhyun Kim et al. (2024) developed a system to process and cluster tourism research data using LangChain[8]. This study clustered semantically similar data to analyze research trends and explore new possibilities in the tourism industry. LangChain's search and summarization capabilities contributed to effectively managing large datasets and improving the quality of the analysis.

These studies have shown that LangChain and RAG technologies can be applied in each specific domain to provide appropriate responses to domain-specific requirements. However, university school regulations and administrative information have different structures and contents for each school, and user needs are also diverse, so research that reflects this is still needed.

Currently, Daegu University does not provide a chatbot service to search for school regulations and administrative information, and this information is provided in the form of several documents through the administrative service on the Daegu University website. Each document contains different regulations and detailed information, making it challenging for users to quickly find the information they need. To solve this problem, this study utilized LangChain-based RAG technology and implemented it as an app-based interface so that users can search for information that is most relevant to school regulations and administration-related questions and quickly check the necessary information. Through this, we would like to propose a system that increases the efficiency of information access and reduces the time and effort required to search for information.

### 3. System design and configuration

Figure 1. is a diagram illustrating the overall process of the RAG -based Q&A system, which is designed to manage Daegu University's school regulations and administrative information and provide reliable responses. The system includes the following steps from data processing to user response generation.

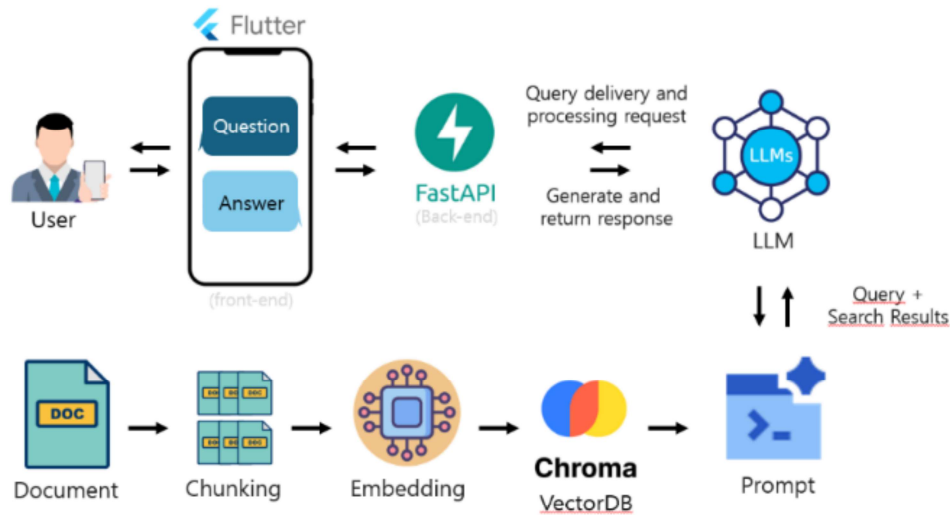


Figure 1. RAG-based Q&amp;A system structure diagram

### 3.1 Data collection and preprocessing

Daegu University Academic Regulations and Administrative Data is available in the Regulations section of the Daegu University website and contains various documents related to university administration and academic operations, such as “Tuition Regulations,” “Academic Grade Evaluation Regulations,” “Leave of Absence and Reinstatement Procedures,” etc. The materials are primarily provided in Hangul document format (HWP) and consist of a mixture of tabular data and unstructured text. In order for a large-scale language model (LLM) to process it efficiently, preprocessing to reconstruct and clean the data was essential.

For example, in the Daegu University school regulations document, tabular data including “Transitional measures following college integration and department (major) name change” contained clear relationships between columns, such as “Before integration” and “After integration.”. However, when such table data was simply converted to text, the relationships between columns and structural context were lost, making it difficult to understand the meaning of the information. Figure 2. shows this original table data. For example, if a relationship indicating that the “Urban Planning Engineering major in the Department of Urban Planning and Landscape” was changed to “Architecture Department” was converted to a text simply listed, the relationship between the columns was not clearly revealed, and it was possible that the language model would not understand it correctly.



Before Integration		After Integration	
College of Business and Economics	Department of Trade	College of Business	College of BusinessCollege of Business
College of Business and Economics	Department of Business Administration		
College of Applied Life Sciences	Division of Data Science, Major in Mathematics and Industrial Mathematics	College of Applied Life Sciences	Department of Big Data
College of Applied Life Sciences	Division of Mathematical and Big Data Sciences, Statistics and Big Data Major		
College of Engineering	Department of Urban and Landscape Architecture, Urban Planning Engineering Major	College of Engineering	Department of Architectural Engineering
College of Engineering	Department of Convergence Industrial Engineering	College of Engineering	Department of Mechanical Engineering (Major in Mechanical Engineering or Mechanical Design Engineering)

Document Content: Before Integration After Integration  
 College of Business and Economics Department of Trade College of Business College of BusinessCollege of Business College of Business and Economics Department of Business Administration  
 College of Applied Life Sciences  
 Division of Data Science, Major in Mathematics and Industrial Mathematics College of Applied Life Sciences  
 Department of Big Data College of Applied Life Sciences  
 Division of Mathematical and Big Data Sciences, Statistics and Big Data Major  
 College of Engineering Department of Urban and Landscape Architecture, Urban Planning Engineering Major College of Engineering Department of Architectural Engineering  
 College of Engineering Department of Convergence Industrial Engineering College of Engineering Department of Mechanical Engineering (Major in Mechanical Engineering or Mechanical Design Engineering)

Figure 2. Some output in a structured format, such as a table

To solve this problem, the table data was manually reconstructed and converted into a form that maintains the relationship and context between columns. Figure 3. shows the reconstructed text data and is designed to maintain the relationship between "before integration" and "after integration". This work was an important step in preserving the consistency of the data and helping the language model to handle it. In addition, in addition to the table format data, the unstructured text of school regulations and administrative data was classified based on the title, main item, and keyword of the regulation to increase the relevance. Unnecessary information was removed, and the structure and meaning of the document were succinctly organized.

Document 1 Content: < Major Changes Before and After Department Integration >  
 1. College of Business and Economics  
 Before Integration: Department of Trade, Department of Business Administration  
 After Integration: College of Business College of BusinessCollege of Business  
 2. College of Science and Life Convergence  
 Before Integration: Department of Mathematics and Big Data, Mathematics/Industrial Mathematics Major, Department of Mathematics and Big Data, Measurement/Big Data Major  
 After Integration: Integration into the Department of Science and Life Sciences Convergence Big University Data  
 3. College of Engineering  
 Before Integration: Department of Urban and Landscape Architecture, Urban Planning Engineering Major,  
 After Integration: Department of Architectural Engineering  
 Before Integration: Department of Convergence Industrial Engineering  
 After Integration: Department of Mechanical Engineering (Major in Mechanical Engineering or Mechanical Design Engineering)

Figure 3. Convert structured data to text format

Then, the organized data was divided into approximately 1000 character units using Recursive Character Text Splitter. Each text chunk maintained the continuity of the context by adding a superposition of 300 characters. This allowed the language model to better understand the back-and-forth relationship of the text and to generate responses suitable for user queries.

### 3. 2 Build embedding and search algorithms

The preprocessed Daegu University academic regulations and administrative data were converted into high-dimensional vectors using OpenAI's GPT-4 embedding model, and the converted data was stored in a vector database called ChromaDB. ChromaDB is a database designed to search for the most appropriate data by calculating the similarity between user queries and data.

During the search process, the MMR (Maximal Marginal Relevance) algorithm was applied to maintain relevance to user queries while minimizing redundant information, thereby producing highly reliable search results. For example, as shown in Figure 4, when the query “tuition fee” is entered, the MMR algorithm searches for the most relevant information within the Daegu University academic policy data, such as “tuition payment standards,” “international student tuition regulations,” and “installment payment procedures.” Returns information as search results. Through this search process, users can clearly understand key information related to the scope of minor completion and quickly check the school policy information most appropriate for your inquiry.

```
question = "I am interested in pursuing a minor. Could you provide an explanation of the relevant procedures and requirements?"
k = 3
answer = ask_question_with_history_and_retrieve_k(question, k)
print(f"Answer: {answer}")
✓ 22.6s
```

Displaying 3 relevant documents:

Document 1:  
Source: C:\Users\lge\RAG\Daegu University Minor Completion Regulations (September 1, 2017)  
Content (Snippet): Daegu University Minor Completion Regulations

Article 1 (Purpose): These regulations are established in accordance with Article 39 and Article 40 of the university academic rules to stipulate the detailed matters regarding the completion of a minor.

Article 2 (Eligibility for Completion): The eligibility for minor completion covers all colleges and majors.

① Students in the following departments are not allowed to pursue a minor: Department of Physical Therapy, Department of Occupational Therapy, Department of Nursing, and Department of Health Administration.

② Students in convergence majors are not eligible to pursue a minor.

Answer: If you wish to pursue a minor at Daegu University, you must follow the procedures outlined below.  
First, although all colleges and majors are generally eligible to apply for a minor, students in the Department of Physical Therapy, Department of Occupational Therapy, Department of Nursing, and Department of Health Administration are not allowed to do so.  
There are no restrictions on eligibility to apply for a minor in general. However, students admitted through special admission programs—such as athletes or those in the special Health Administration track—are not permitted to apply for a minor.

Figure 4. Example of MMR Similarity Based Search

Through this search process, we go beyond simply listing data and selectively provide key information most relevant to user queries, allowing users to search for data that suits their needs.

### 3.3 Server integration and application implementation

The built system is designed in a structure that interfaces FastAPI and ChromaDB to process user questions, and the GPT-4 model generates the final response. FastAPI is a lightweight web framework that provides high performance and fast request handling through support for asynchronous and concurrent processing [7]. This study leveraged these advantages of FastAPI to design a query-answering system for Daegu University's academic regulations.

In particular, we optimized the system's scale-up performance by using Hypercorn, an Application Server Gateway Interface (ASGI) server [8]. Hypercorn is an asynchronous ASGI server that supports HTTP/1.1, HTTP/2, and WebSockets, enabling fast data transfer and efficient handling of multiple requests simultaneously. Through this, we built a system that processes large-scale school policy data and provides real-time responses.

Figure 5. shows a question-answering test performed using Postman, an API testing tool, to verify the FastAPI question-answering system [9]. When a user enters the question "What do I need to do to get a student ID card?", the question is sent to the server through an HTTP request. The request data is in JSON format, and after processing it, the server searches ChromaDB for documents related to student ID processing and generates and returns a response through the GPT-4 model. As a result of the test through Postman, it was confirmed that the FastAPI server returned an accurate response to the input student ID issuance method.

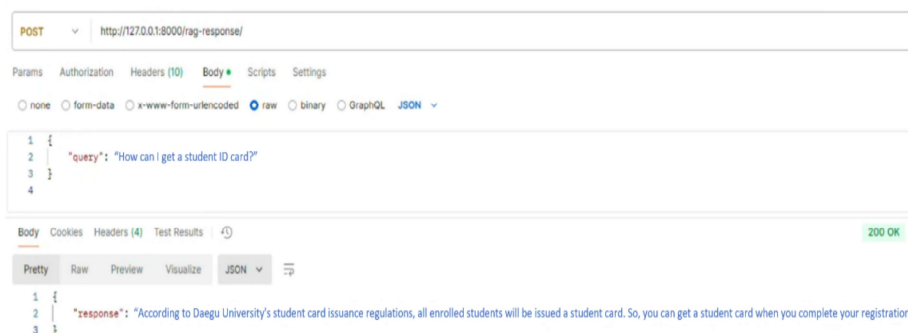


Figure 5. API testing using Postman

A Flutter-based mobile application was developed for the front-end implementation [10]. The application is designed to provide an intuitive user interface

(UI) so that users can effectively utilize the question-and-answer system related to school rules.

In Figure 6, the application consists of a splash screen and a question and answer screen. The splash screen is the first thing that appears when the application is launched, and it conveys the purpose and identity of the service to the user through the Daegu University logo and the chatbot title. The question and answer screen displays the title “Academic Rules Response System” at the top, and a question input field and a send button are placed at the bottom. The input field defaults to the prompt “How can I help you?” to prompt the user to type in a question, which can be sent via the Send button (arrow icon) in the bottom right corner.



Figure 6. Application implemented in Flutter

#### 4. Experiment results

The figures below show examples of responses provided by the system to various questions entered by the user in the Flutter-based application.

Figure 7. shows how the system searches for relevant information from Daegu University's school regulations data and guides the documents necessary for issuing student cards and application procedures when the user enters the question, "How do I issue a student card?". In addition, in response to the question, "How do I cancel my

leave of absence?" the system explained the procedure for applying for cancellation of leave of absence and precautions step by step. This allows users to easily understand and execute the necessary administrative procedures.

In Figure 8, for the question "What are the criteria for issuing an employment recommendation letter?", the system retrieved the relevant academic policy data to show the grade point average requirement (3.5 or higher) and the application process for obtaining an employment recommendation letter. Meanwhile, Figure 9. illustrates how the system responds to queries that are not included in the dataset, informing users that the requested information could not be found.

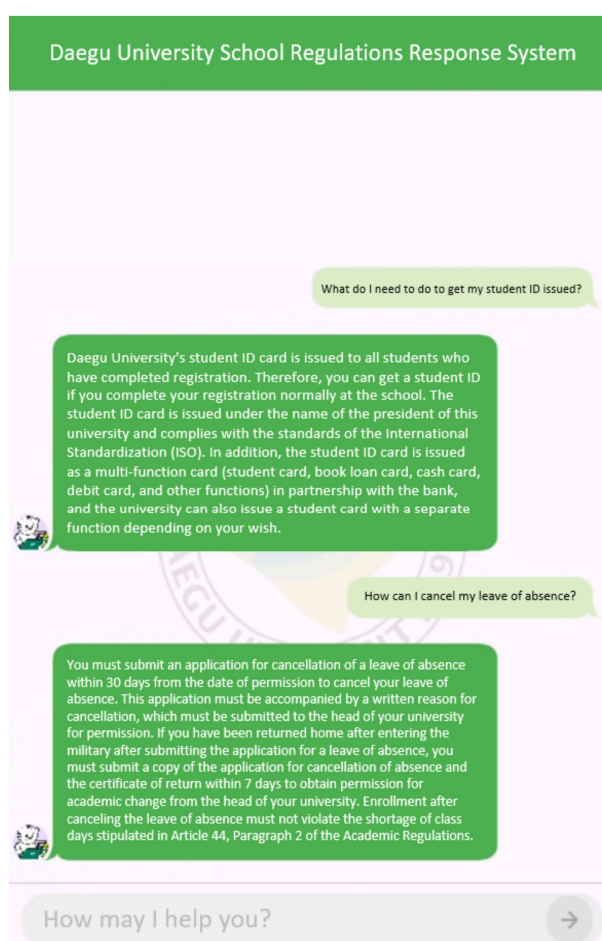


Figure 7. Q&A with Implemented Applications (1)

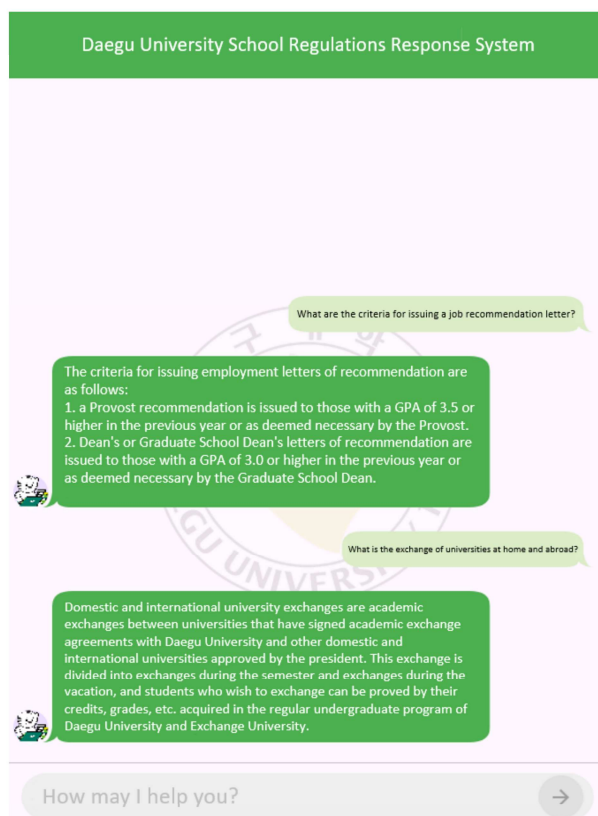


Figure 8. Q&amp;A with Implemented Applications (2)



Figure 9. Responding to questions that do not exist in the dataset

## 5. Conclusion

In this study, we developed a system that provides academic and administrative information of Daegu University in the form of questions and answers by utilizing the RAG(Retrieval-Augmented Generation) technique. Users were able to obtain information more quickly about school regulations-related questions such as tuition regulations and graduation requirements, which greatly improved the efficiency of accessing academic and administrative information.

However, since structured data such as tabular format must be preprocessed manually so that LLM can accurately process it, there is a limitation in that additional time and cost are required. To address these issues, future research will improve the system's processing efficiency by applying technologies that can integrally process images and text using multimodal data.

## References

- [1] Lanchain, [https://python.langchain.com/v0.1/docs/modules/model\\_io/lms/](https://python.langchain.com/v0.1/docs/modules/model_io/lms/). Accessed: December 12, 2024.
- [2] OpenAI, "GPT-4 Technical Report," Available: <https://openai.com/research/gpt-4>, arXiv:2303.08774 (2024).
- [3] OpenAI, <https://openai.com/>. Accessed: December 12, 2024.
- [4] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions," arXiv, vol. 2311.05232, pp. 1-2, Nov. 2023.
- [5] Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, Chun Jason Xue, "Retrieval-Augmented Generation for Natural Language Processing: A Survey," arXiv, 2024.
- [6] Choi Jongmyung, Lee Youngho, and Kim Sun Kyung, "Design of a KTAS Decision System using LangChain," (Jour. of KoCon.a) The Journal of the Korea Contents Association, pp. 9-10, 2024.
- [7] Hoon So, Seongjun Kang, Chanwoo Jeong, and Hyunjun Jung, "Unreal Engine's specific NPC chatbot system using Langchains," Proceedings of KIIT Conference, pp. 394-398, 2024.
- [8] Namhyun Kim, Sehyoung Kim, and Juyoung Kang, "Analysis of National Research and Development Trends Using LangChain, and Clustering of Research Outcomes: Focused on the NTIS Tourism Industry," The Journal of Society for e-Business Studies, Vol. 29, No. 2, pp. 93-115, 2024, DOI: 10.7838/jsebs.2024.29.2.093
- [9] ChromaDB, <https://www.trychroma.com/>. Accessed: December 12, 2024.

- [10] FastAPI, <https://fastapi.tiangolo.com/ko/>. Accessed: December 12, 2024.
- [11] Hypercorn, <https://hypercorn.readthedocs.io/en/latest/>. Accessed: December 14, 2024.
- [12] Postman, <https://www.postman.com/>. Accessed: December 14, 2024.
- [13] Flutter, <https://flutter.dev/>. Accessed: December 12, 2024.