World Journal of Information Technology
Print ISSN: 2959-9903
Online ISSN: 2959-9911
DOI: https://doi.org/10.61784/wjit3083

THE CONSTRUCTION, EVALUATION, AND APPLICATION OF AN INTELLIGENT ADMISSIONS Q&A SYSTEM BASED ON RETRIEVAL-AUGMENTED GENERATION (RAG)

JiaXin Wang[1#], ZiHan Qi[1#], JingJing Li[1], XuanEr Chen[2], Yuan Lin[1*]
[1]*Business School, Dalian University of Technology, Panjin 124221, Liaoning, China.*
[2]*School of Chemical Engineering, Ocean and Life Sciences, Dalian University of Technology, Panjin 124221, Liaoning, China.*
[#]*JiaXin Wang and ZiHan Qi are both the first authors.*
[*]*Corresponding Author: Yuan Lin*

Abstract: This study addresses the pain points in university admissions consultation, such as "human response bottlenecks" and "information overload and fragmentation", by proposing an intelligent admissions consultation system solution based on Retrieval-Augmented Generation (RAG) technology. The system adopts the LangChain framework, integrates Large Language Models (LLMs) with RAG technology, and constructs a complete workflow including knowledge base construction, retrieval enhancement, and answer generation. Experiments show that the system significantly outperforms the control groups in key metrics, including overall accuracy (95.6%), security compliance rate (96%), and dynamic personality matching rate (92.3%). This system not only effectively improves the response efficiency and answer quality of admissions consultation but also provides a replicable and scalable practical reference for the digital and intelligent upgrading of university admissions management services.
Keywords: Retrieval-Augmented Generation; Large language model; Intelligent question-answering system; University admissions counseling

1 INTRODUCTION

Driven by the national strategy of artificial intelligence empowering educational reform, the rapid rise of artificial intelligence has transformed multiple fields, with education emerging as a primary area of impact [1]. Generative artificial intelligence, represented by models such as DeepSeek and ChatGPT, has gradually deepened its applications in higher education sectors like university admissions publicity [2,3], leveraging its powerful natural language processing capabilities, multimodal interaction abilities, and chain-of-thought reasoning technologies. The state has clearly defined the development goal of building artificial intelligence education large models. Meanwhile, as China's higher education moves towards the popularization stage, local universities, as important pillars of the higher education system, shoulder the responsibility of cultivating a large number of high-quality talents for regional economic and social development [4]. With the continuous expansion of university enrollment scales, the number of applicants and admissions information have grown exponentially. Traditional admissions consultation methods relying on manual work, including information matching and student source analysis, have become increasingly inadequate in coping with massive data and personalized needs, facing four core pain points: "human response bottlenecks", "information overload and fragmentation", "information barriers between disciplines", and "difficulties in sensitive information security management".

Taking the admissions practice of Dalian University of Technology as an example, although universities have carried out admissions publicity through multiple channels, during the critical window period from the release of college entrance examination scores to the deadline for application submission, admissions consultation still faces prominent pain points: consultation demands surge intensively during this window and the questions are highly repetitive, making manual answering time-consuming and labor-intensive; staff have professional limitations, and are prone to deviations in interpreting interdisciplinary questions and admissions policies; information update and transmission efficiency is low, making it difficult to achieve timely synchronization; some internal information is not publicly available, resulting in applicants and parents being unable to obtain accurate responses. The above problems may cause potential qualified applicants with matching scores and strong willingness to apply to give up their applications due to ineffective answers, leading to the loss of high-quality student sources. To stand out in this highly competitive environment, universities urgently need to innovate and transform their admissions publicity strategies [5], and artificial intelligence has inevitably become an essential tool for providing better services, improving efficiency, and reducing errors by supporting quality management processes [6].

From the perspective of intelligent educational technology (EdTech), this study, through the practical application of RAG technology, can directly improve the response efficiency, answer quality, and consistency of policy interpretation in the university's admissions consultation services, and significantly reduce the administrative labor costs caused by manual repetitive answering and multiple information synchronizations; at the same time, it can effectively optimize the consultation experience of applicants and parents, solve the problem of untimely and inaccurate information acquisition

during the critical window period, reduce the loss of high-quality matching student sources, and help enhance the university's admissions service image and student source quality; in addition, the implementation plan and experience of RAG technology in university admissions consultation in the EdTech field formed by this study not only improve the scientificity of admissions work but also provide strong support for admissions decision-making [7], and can also provide a replicable and scalable practical reference for other administrative service scenarios in the university and admissions work of similar universities, promoting the digital and intelligent upgrading of university admissions management services.

RAG is a collaborative framework combining retrieval and generation. Its core principle is as follows: first, fragment the knowledge base text and establish indexes using counting methods such as word frequency and TF-IDF; after receiving user queries, calculate keyword matching degrees to retrieve relevant text fragments; then input the fragments as context into the generation model to generate accurate responses supported by facts, fundamentally avoiding the "hallucination" problem of pure generation models. It is suitable for scenarios requiring high reliability such as university admissions consultation and government affairs Q&A, where current scenarios suffer from defects such as low consultation efficiency, poor response consistency, delayed information updates, and professional information barriers.

This study covers the entire implementation process: first, clarify consultation pain points and functional demands through requirement analysis, select the LangChain framework, realize multi-source data vector conversion and deep retrieval with RAG technology as the core, and then complete system architecture design, development and deployment, and effect evaluation.

The study adopts the literature research method to sort out the technical context, the case design method to customize the system scheme suitable for admissions consultation, the controlled experiment method to compare the efficiency and accuracy between the system and manual consultation, and the questionnaire survey method to collect user experience feedback.

The technical route is presented in the form of diagrams and texts: first, integrate materials such as admissions policies and major introductions to build a structured knowledge base; use LangChain to split long documents into semantic fragments and convert them into high-dimensional vectors for storage in a dedicated database; then develop a RAG engine, and realize functions such as agent active question prediction, permission hierarchical management and control, and consultation period opening relying on "chain" programming; finally complete system integration testing. At present, front-end construction, back-end path verification, and simulated knowledge base access testing have been realized, and the full-function operation can be achieved by replacing real information. Work Flow Chart of Agent Based on Large Language Model is shown in figure 1.

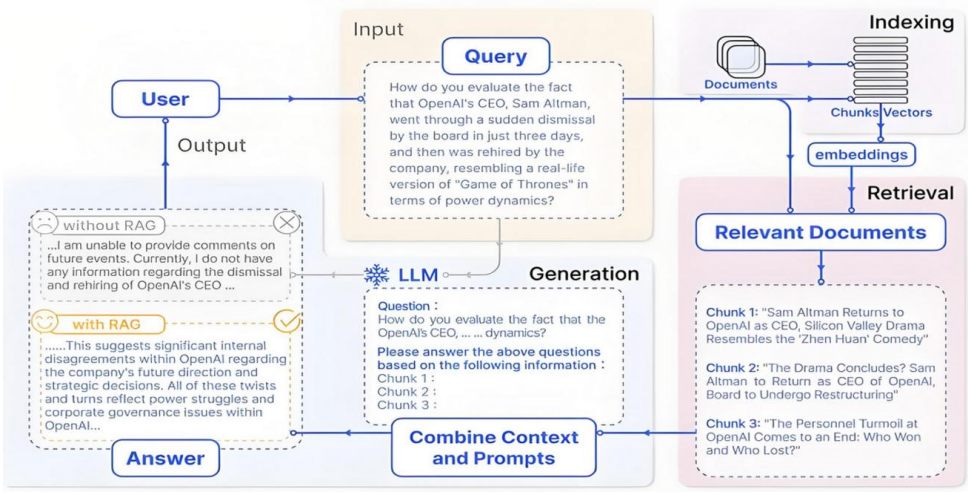

Figure 1 Work Flow Chart of Agent Based on Large Language Model

2 RELATED THEORIES AND TECHNICAL FOUNDATIONS

2.1 Overview of Intelligent Question Answering Systems

2.1.1 Intelligent question answering systems and their typical architectures

With the development of the times, humanity has entered an era where artificial intelligence technology deeply integrates into daily life, and the way of obtaining information has gradually transformed from simple keyword retrieval to a more efficient and accurate intelligent question-answering interaction mode. An Intelligent Question Answering System (IQAS) is an artificial intelligence application based on natural language processing technology, which provides users with accurate information services through interactive question-answering. Its core architecture includes three modules: question understanding, knowledge retrieval, and answer generation. It uses technologies such as knowledge graphs and machine learning to convert unordered corpus into a structured knowledge base, realizing automated information processing and personalized recommendation (as shown in Figure 2). As one of the core applications in the

field of Natural Language Processing (NLP), it can automatically understand semantics based on natural language questions input by users and return accurate and concise answers, greatly reducing the threshold and cost of information acquisition.

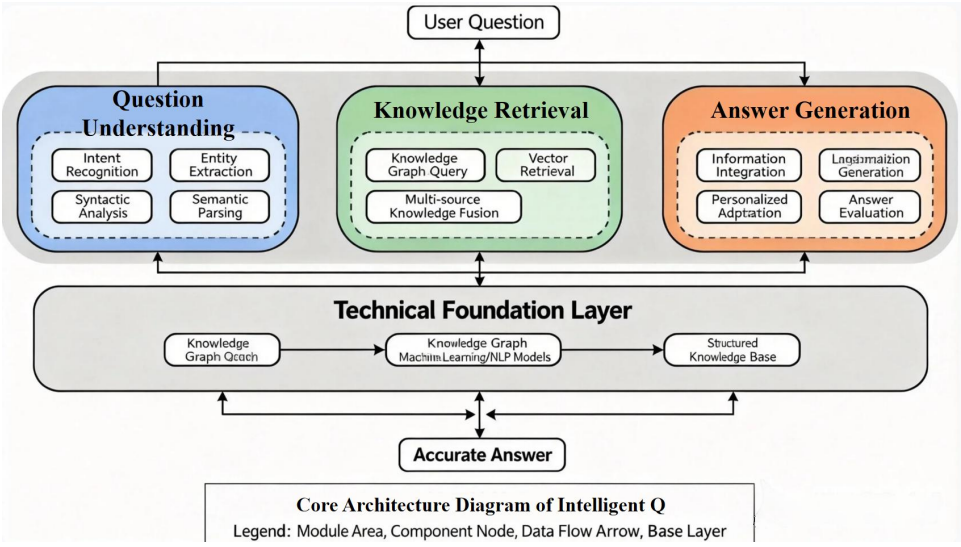

Figure 2 Core Architecture of Intelligent Question Answering System

2.1.2 Challenges faced by typical architectures

Although intelligent question answering systems have become an indispensable part of modern life, they still have certain limitations. In terms of question understanding, existing intelligent question answering systems have problems such as poor adaptability to ambiguous questions, poor domain adaptability, and insufficient question understanding. In terms of knowledge retrieval, there are issues of single and outdated retrieval methods, which cannot comprehensively answer users' questions. In addition, existing models also have the problem of "making up information", and the authenticity, traceability, and accuracy of retrieved data remain to be verified. With the development of the times, people's requirements for intelligent question answering systems will be higher and higher, and they will pay more attention to the connectivity of context. However, existing intelligent question answering systems are deficient in this regard, often facing urgent problems such as incoherent context connection, inaccurate answers, and slow response speed.

2.2 Integration of Retrieval-Augmented Generation (RAG) Technology

In view of the shortcomings of existing intelligent question answering systems mentioned above, the author introduces RAG technology. Retrieval-Augmented Generation (RAG) technology improves the reliability and accuracy of generation models by integrating knowledge from external data sources [8]. This technology combines traditional language models with external knowledge bases, and through dynamic information retrieval before text generation, it not only enriches the generated content but also ensures the relevance and timeliness of the content. Combining RAG with intelligent question answering systems can effectively solve the above-mentioned pain points of outdated data, lack of authenticity to a certain extent, and slow retrieval speed.

The so-called RAG technology enables machines to quickly and accurately locate corresponding information and provide answers. Its main workflow involves four parts as shown in Figure 3: "Indexing", "Retrieval", "Augmentation", and "Generation". Next, the author will explain the involved processes and related technologies in simple language: when an agent receives a user's question, it first uses a text splitter to split the sentence into small segments and then labels each segment through an embedding model, completing the first step of "Indexing". Using the indexed information, it parses through a vector database to find corresponding content according to the meaning, even if the question expression is different, it can find accurate content. The subsequent retrieval algorithm is like a content filter, which selects some keywords from the content to obtain the most accurate information, completing the "Augmentation" process. Finally, with the help of a large language model, which acts as an assistant to translate into human-understandable language, it organizes the found scattered content into coherent and easy-to-understand answers, ultimately generating accurate answers. This is the core idea and workflow of RAG technology.

With the rapid development of large language model technology, RAG enhanced in the reasoning stage has gradually become mainstream, and its development architecture has also made significant progress, especially frameworks such as LangChain, LlamaIndex, and AutoGen. These frameworks provide a series of modular functions, covering data segmentation, import, vectorized storage, retrieval optimization, and answer generation. The integrated functional design greatly simplifies the construction process of the RAG system pipeline and improves development efficiency and system flexibility [9]. The two strategies of retrieval re-ranking and multi-path retrieval used in RAG technology can effectively improve the retrieval accuracy in the question-answering system. These applications have once again greatly improved the answering efficiency of intelligent question answering systems.

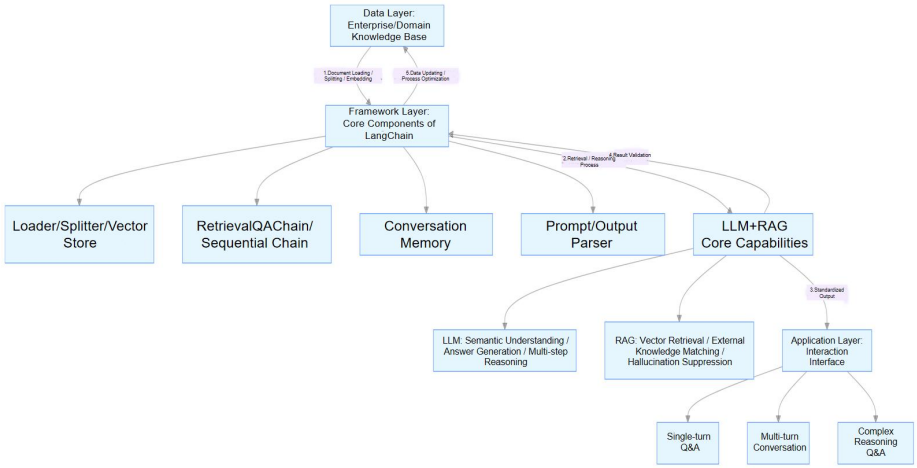

Figure 3 RAG Technology Workflow

2.3 Large Language Models (LLM) and Agent Frameworks

2.3.1 LLM basics and prompt engineering for RAG scenarios

A Large Language Model (LLM) refers to a deep learning model trained using a large amount of text data, which can generate natural language text or understand the meaning of language text. Simply put, integrating LLMs into intelligent question answering systems brings significant improvements compared to traditional intelligent question answering systems in three core dimensions: capability boundary, interactive experience, and adaptation efficiency.

Based on Transformer and self-attention mechanism, LLMs break through the semantic understanding bottleneck of traditional systems and improve the ability to handle complex questions. At the same time, LLMs have context understanding capabilities, can handle multi-round follow-up questions from users, and provide coherent answers with good context connection. The above functions rely on the architectural foundation of LLMs, namely Transformer and its self-attention mechanism. The following is an analysis from the perspective of its capability foundation: LLMs support generative answers and quickly adapt to different fields through the pre-training + fine-tuning mode, improving their generalization ability and enriching built-in general knowledge through massive corpus pre-training. Even when retrieval results are limited, they can give basic answers based on their own knowledge. Finally, it is understood that pure LLMs still have certain limitations, such as poor timeliness and knowledge hallucinations. Then, integrating RAG technology with them again: the process of first enhancing the retrieval of external knowledge bases through RAG technology and then inputting the retrieval results as context into LLMs can not only retain the advantages of generative answers in intelligent question answering systems but also solve the hallucination problem, ensuring the accuracy and reliability of the question answering system.

2.3.2 Establishment of the LangChain framework

LangChain is a development framework specially designed for building Large Language Model (LLM) applications. It is a set of modular toolchains that enable LLMs to efficiently connect external resources, implement complex logic orchestration, and build end-to-end question-answering links in intelligent question answering systems. Applying LangChain to intelligent question answering systems and establishing a core framework for this purpose mainly follows the principles of modular splitting, componentized calling, and process-oriented orchestration. Its typical framework is mainly divided into model layer, data connection layer, chain layer, memory layer, and application layer. Through these layers, rapid data integration is ensured, the scalability and flexibility of the question answering system are improved, and the operation difficulty is greatly reduced.

Take the university intelligent question-answering system as an example: import the university's historical admissions data through the Langchain document loader, split the data into small fragments, embed the model to convert them into vectors, and store them in the vector database. Vector databases focus on the storage, analysis, and retrieval of vector data, converting text data into vector form through vectorization models, thereby achieving efficient data management [10]. To enable comparison, search, and analysis between professional knowledge texts in the admissions knowledge base, the Tongyi text vector model is called to embed the segmented admissions knowledge text into high-quality vectors. These vectors capture the deep semantic features of the admissions knowledge text and are stored in the PostgreSQL vector database [11]. Then, when users input complex questions, the data chain, namely LLM and RAG, will collaboratively complete the retrieval to achieve end-to-end answering from understanding to retrieval and generation. With the assistance of Langchain, the rapid construction of a RAG-enhanced question answering system is ensured. When the underlying data is updated, only the model parameters need to be adjusted without modifying the overall process, which greatly enhances the accuracy of knowledge and the flexibility of the system.

2.4 Construction of an Intelligent Question Answering System Based on Deep Retrieval-Augmented Generation

Guu et al. proposed the REALM model and introduced a knowledge retriever, thereby explicitly utilizing text corpus knowledge during the pre-training phase [12]. Borgeaud et al. designed the RETRO model, which retrieves and encodes

document fragments in the corpus, significantly reducing the number of parameters [13]. The technical path of large model RAG question-answering for standard documents is mainly divided into two functional modules: the first functional module mainly conducts intention analysis on users' questions, adopts regular and model algorithm methods for keyword extraction, and further maps them into question intentions; the second functional module is document question-answering based on LLM. In this step, the most critical aspects are document chunking, hierarchical title extraction, and table-to-sentence conversion [14].

In summary, constructing an intelligent question answering system based on retrieval-augmented generation (hereinafter referred to as "RAG question answering system") involves three key links: data index construction, document retrieval, and text generation [15]. Its core goal is to build a question answering system that can efficiently utilize external knowledge bases, significantly reduce model hallucinations, and provide accurate and reliable answers. As shown in Figure 4, a framework with clear responsibilities and high decoupling is established through the construction logic of data standardization process - componentized reasoning - intelligent application - scenario-based implementation. Under this framework, each link operates like an "intelligent question-answering factory". First, Langchain conducts an offline preparation in the system, reads various documents through loading tools, then splits them, and finally stores them to form an intelligent knowledge warehouse. When a user asks a question, the process officially starts. Langchain calls its own model interaction module to send the user's question to the large language model. After the large language model understands the question and disassembles the core needs, it dispatches the retrieval module with the assistance of Langchain. At the same time, RAG technology comes into play. It accurately finds the content of documents most relevant to the question through vector similarity matching and brings these key materials back to Langchain. At this point, Langchain reorganizes the user's question and the found key materials into a format understandable by the LLM and uploads it. Then the LLM plays its role in organizing language and ensuring content authenticity. If the user has multi-round interaction needs, Langchain's memory module will directly retrieve the above background and provide it to the LLM. The LLM combines new materials to quickly give answers, realizing coherent dialogue. It is worth mentioning that in this process, the system has a reverse verification link. After the LLM generates an answer, it will be fed back to Langchain. If it is found that the materials are insufficient or the answer needs to be verified, Langchain will re-dispatch the retrieval module for supplementary search to ensure the accuracy of the answer.

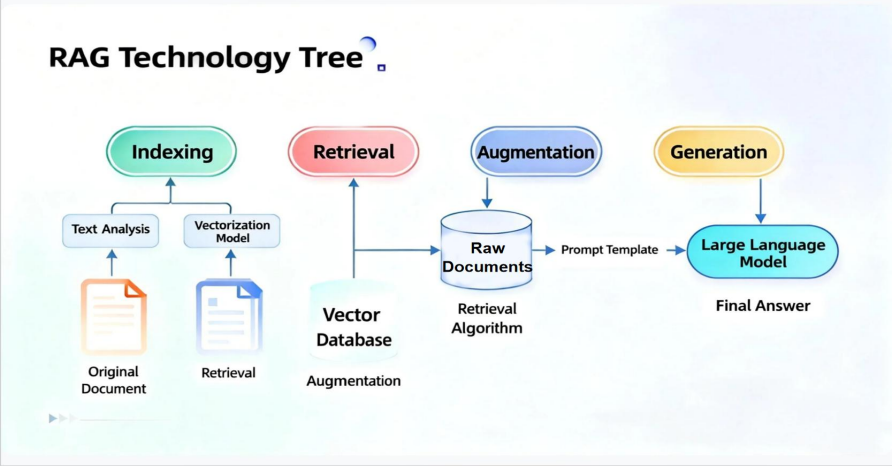

Figure 4 Framework of Intelligent Question Answering System Based on Deep Retrieval-Augmented Generation (RAG)

This framework fundamentally solves the pain points of traditional intelligent question answering systems, such as weak semantic understanding, narrow knowledge coverage, rigid interaction, and high development costs. It upgrades the single-round memoryless answering method to a multi-round coherent interactive answering method, which not only retains the naturalness of generative answers but also ensures the accuracy of answers through RAG technology. At the same time, it lowers the threshold for implementation with the help of Langchain, greatly improving the practicality of intelligent question answering systems.

3 SYSTEM DESIGN: RAG AGENT FOR ADMISSIONS CONSULTATION

3.1 System Requirement Analysis

3.1.1 Functional requirements

Functional requirements clarify the core services and operational capabilities that the agent should achieve, covering five dimensions: question-answering interaction, information retrieval, knowledge base management, security control, and statistical analysis, ensuring that the system can efficiently and credibly serve the admissions consultation scenario:

(1) Basic Question-Answering Requirements

The system should support multi-modal question interaction, have context-aware multi-round dialogue capabilities, automatically identify user intentions and predict associated follow-up questions, and maintain dialogue coherence and

service efficiency. Key information must be attached with traceability identifiers, such as citing the "2025 Grade XX Major Talent Training Program", to enhance the credibility and authority of answers.

(2) RAG Retrieval Requirements

The system needs to implement multi-dimensional accurate retrieval, including keyword retrieval, semantic retrieval, and fuzzy retrieval, and can parse the core query intentions behind users' colloquial expressions. Retrieval results should be dynamically sorted based on relevance, timeliness, and importance, prioritizing the presentation of the latest admissions policies and high-frequency consultation content to ensure the accuracy and timeliness of generated answers.

(3) Knowledge Base Management Requirements

Support structured entry of admissions-related information, including admissions regulations, major introductions, admission rules, historical scores, application guides, etc., compatible with formats such as text, PDF, and Word. The knowledge base should have real-time update and version control capabilities to adapt to annual policy adjustments and major setting changes, ensuring timely information synchronization and historical traceability, and avoiding misoperation of key data.

(4) Security Management Requirements

Implement role-based access control, and assign differentiated permissions to university staff. For example, administrators have full system permissions, operation personnel are limited to knowledge base maintenance and statistical queries, and customer service personnel can only view dialogues and transfer to manual services, ensuring the minimum permission access to sensitive information.

(5) Statistical Analysis Requirements

The system should automatically collect and visualize consultation data, including key indicators such as total consultation volume, time period distribution, high-frequency questions, unresolved queries, and manual transfer rate. By analyzing student source geographical distribution, major attention, and score segment characteristics, it provides data support for formulating admissions strategies and optimizing publicity resources. At the same time, it supports real-time push of consultation hotspots, helping the university to update the knowledge base or adjust publicity materials in a timely manner.

3.1.2 Non-functional requirements

Non-functional requirements focus on the performance, reliability, usability, and other quality characteristics of the system, ensuring that the system can operate stably and efficiently during the peak period of university admissions consultation, while taking into account the convenience of university operation and maintenance and user use. All requirements are in line with the professional attributes and scenario characteristics of admissions work.

(1) Performance Requirements

The system response time should meet the standard that the average text question-answering time is ≤ 1 s, the overall voice question-answering response time is ≤ 3 s, and there is no significant delay when the concurrent consultation volume surges. The system should support at least 500 simultaneous online consultations and have elastic expansion capabilities to cope with high concurrent access during the admissions season and improve service throughput and user experience.

(2) Maintainability Requirements

Adopt a modular architecture design to achieve low coupling between functional modules such as question-answering, retrieval, knowledge base, and statistics, ensuring that local faults do not affect the overall operation of the system. Provide a visual operation and maintenance interface, support real-time monitoring of operating status such as server load, response delay, and concurrency, and realize abnormal early warning to reduce system operation and maintenance costs and risks.

3.2 Overall System Architecture Design

Retrieval-augmented generation technology is an advanced artificial intelligence method that combines external information retrieval and large language model text generation technology, which can effectively solve problems such as knowledge lag, generation hallucinations, insufficient professional domain knowledge, and inability to utilize sensitive data [16]. This system is built based on retrieval-augmented generation technology, and by combining external knowledge retrieval with large language model generation capabilities, it effectively alleviates problems such as delayed knowledge updates, generation hallucinations, insufficient domain knowledge, and restricted utilization of sensitive data. The system adopts a layered architecture, which is composed of the infrastructure layer, data layer, RAG engine layer, application service layer, and user interaction layer from bottom to top. Each layer has clear responsibilities and cooperates with each other to support the stable service of the system in diverse terminal environments.

Among them, the user interaction layer provides user interfaces for multiple terminals such as user Web, APP, and business application programming interfaces (APIs). Its main purpose is to realize the interaction between the system and users, provide a consistent and smooth user experience, and ensure that the system can adapt to service requirements in different terminals and environments.

The application service layer consists of a question-answering module and a management module. The question-answering module includes several sub-modules such as knowledge retrieval, re-ranking, prompt words, and answer generation, whose purpose is to ensure the coherence of multi-round dialogues and provide technical support for keyword retrieval, semantic retrieval, and fuzzy retrieval [17]; the management module is responsible for collecting and

organizing user information, further processing and summarizing it to form visual charts reflecting system work efficiency and potential student source characteristics, supporting operational decision-making and resource optimization.

The RAG engine layer is the core architecture of the system. This layer implements three key links: index construction, retrieval, and generation under the LangChain framework. The indexing stage completes the vectorization and storage of knowledge texts; the retrieval stage integrates dense retrieval and sparse retrieval strategies to achieve accurate recall of multi-source information; the generation stage generates accurate and coherent consultation replies based on retrieval results and large language models. This layer integrates components such as prompt engineering and Agent tool calling to enhance the system's semantic understanding and task execution capabilities [18].

The data layer integrates the university's internal structured and unstructured admissions data to build a localized and authoritative admissions knowledge base. Structured data includes historical enrollment plans, admission scores, admission results, etc.; unstructured data covers admissions regulations, college and major introductions, policy documents, etc. Through large models, intention analysis and QA pair construction are carried out on hot consultation questions, semantic-preserving segmentation processing is performed on documents, and they are converted into high-quality vector representations, which are stored in the vector database to support upper-layer retrieval.

The infrastructure layer realizes security permission management requirements and is responsible for computing resource scheduling, network communication, storage management, and security permission control. Through the implementation of fine-grained identity authentication and access authorization mechanisms, the system data security and operational compliance are ensured.

3.3 Detailed Design of Core Modules

3.3.1 Construction of the knowledge base module

Knowledge base construction is the foundation for the system to realize knowledge-driven intelligent question answering, and its process covers three links: data source governance, text preprocessing, vectorization, and index construction.

Data Source Governance: The system connects to multi-source heterogeneous data such as the university's admissions information network, academic affairs system, and policy document database to form a structured admissions knowledge system. Knowledge sources should include admissions regulations, major introductions, historical admission data, frequently asked questions compilations, policy notices, etc., and establish metadata specifications and update mechanisms to ensure the authority and timeliness of information sources.

Preprocessing Pipeline: Perform cleaning, formatting, and intelligent segmentation of original texts. The cleaning stage removes irrelevant characters and corrects encoding errors; the formatting stage unifies text styles and punctuation; intelligent segmentation adopts sentence and paragraph segmentation strategies based on large models according to the principle of semantic integrity to avoid semantic fragmentation and provide high-quality input for subsequent vectorization.

Vectorization and Indexing: Select an embedding model suitable for Chinese semantics to perform vector representation of segmented texts and store them in the vector database. Existing studies have shown (Zhao Bang, Cao Shujin, 2025; Ruan Kun et al., 2025) that combining efficient vector indexes such as FAISS and Milvus can significantly improve retrieval speed and accuracy. On this basis, the system designs a dynamic index update mechanism to support real-time synchronization and version management of knowledge base content.

3.3.2 Construction of the retrieval and enhancement module

The retrieval and enhancement module is responsible for accurately locating relevant information from the knowledge base and synthesizing it into context understandable by large language models, which is the key to ensuring the relevance and accuracy of answers.

Hybrid Retrieval Strategy: Combine dense retrieval and sparse retrieval methods to improve information recall rate. Dense retrieval is based on semantic vector similarity matching, suitable for generalized queries [19]; sparse retrieval relies on keyword matching technologies such as BM25, suitable for specific queries. By weighted fusion of the results of the two retrieval methods, multi-dimensional and high-coverage document recall is achieved.

Re-ranking Mechanism: Design a dynamic context assembly strategy to deduplicate, filter, and logically organize multiple retrieved document fragments to construct a structured prompt context with clear structure and high information density. Combined with task-specific prompt templates, guide large language models to generate accurate, professional, and context-compliant admissions replies, while controlling output format and style to ensure the standardization and readability of information transmission.

Context Construction and Prompt Engineering: Multiple retrieved fragments need to be deduplicated, filtered, and logically integrated to construct a structured prompt context. Combined with prompt engineering methods (such as role setting, chain-of-thought prompting, etc.), guide large language models to generate admissions context-compliant and format-standardized replies. Relevant literature points out that reasonable prompt design can significantly improve the performance of models in vertical domain tasks and enhance the interpretability and traceability of answers [18].

4 SYSTEM IMPLEMENTATION, DEPLOYMENT, AND EMPIRICAL EVALUATION

4.1 System Implementation Environment and Key Implementations

4.1.1 Development environment and technology stack

To realize the system design, we built the following technology stack: Python 3.9 is used as the main development language, and the core process is quickly orchestrated and integrated using the LLM reasoning framework. Its efficient local deployment feature greatly improves development efficiency. Considering data security, deployment costs, and customizability, we selected the open-source and commercially available Qwen3-Guard-4B as the security check model, Qwen3-4B-Instruct as the label classification model, and Qwen-Plus/GPT-4 as the main large model generation base. It is deployed on a local GPU server through INT4 quantization technology to ensure independent and controllable services. At the data storage and retrieval level, Milvus is used as the vector database for storing and managing embedded vectors of knowledge fragments; the text embedding model selects BAAI/bge-large-zh-v1.5, which has outstanding performance in Chinese semantic matching evaluations, to convert text into 768-dimensional vectors. To further improve retrieval accuracy, the BAAI/bge-reranker-large cross-encoder model is introduced after vector retrieval to re-rank candidate documents. The system back-end uses FastAPI to build RESTful API services, the front-end uses Vue.js to build interactive interfaces, and Docker containerization technology is used for deployment on the university's cloud computing platform to achieve elastic resource scaling and high availability. Back-end Hierarchical Large Model Calling Technology on the Development Side is shown in figure 5.

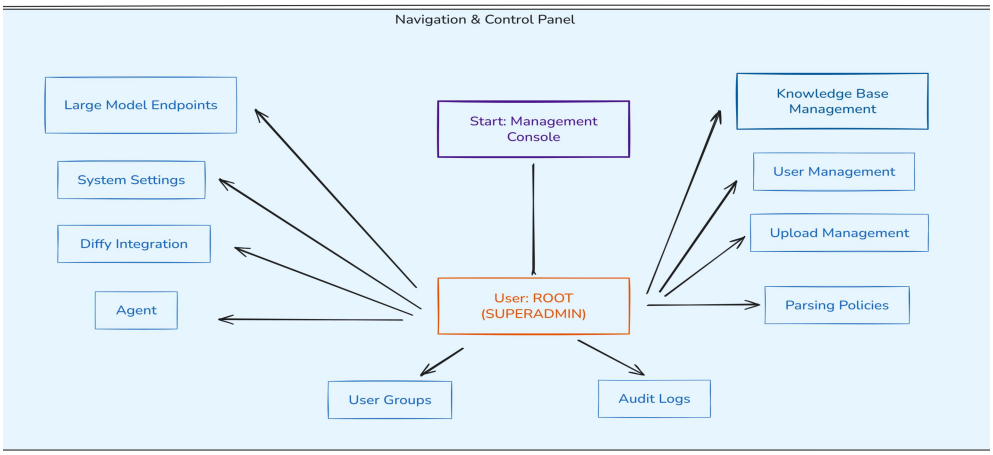

Figure 5 Back-end Hierarchical Large Model Calling Technology on the Development Side

4.1.2 Core process implementation and data flow

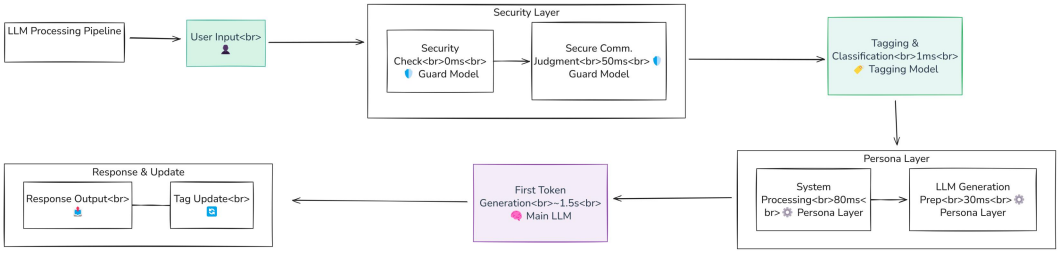

Figure 6 Core Technology Implementation Flow Chart

The core of the system is a personalized question-answering process combining dual-layer small models, dynamic personality, and RAPTOR. Taking the typical question "I want to know the admission score line of the Computer Science major?" as an example, we elaborate on the entire process from user question input to answer generation. As shown in the sequence diagram in Figure 6, the data flow of the process is fully revealed: the user's question first reaches the application service layer, and after being checked by the security check layer (Guard-4B), the request is sent to the label classification layer (Qwen3-4B). The label classification layer performs multi-dimensional label classification on the user's question, outputting intention, emotion, personality matching, and Top5 prompt words. Then, the system processing layer parses the labels, selects dynamic personalities (for example, selects the personality: warm - caring according to the emotion label "anxious"), retrieves the knowledge base, splices prompt words, and sends the complete prompt words to the main large model (Qwen-Plus/GPT-4). The main large model generates personalized answers based on dynamically spliced prompt words. The answers pass through the post-processing module, automatically attach citation sources and format them, and finally return to the user through the application service layer. The key steps of the process are implemented through code based on llama.cpp/vLLM. Its core lies in building a prompt template including security guidance, personality shaping, user portraits, knowledge base content, and current questions, and integrating the chain of label classification, knowledge base retrieval, and dynamic personality selection to ensure the accuracy and personalization of answers.

4.2 Experimental Design and Evaluation Methods

4.2.1 Evaluation objectives

This experiment aims to verify whether the intelligent question answering system constructed in this study (experimental group) has significant improvements in core dimensions such as answer accuracy, response efficiency, security compliance, dynamic personality matching rate, and user satisfaction compared with existing solutions and general models.

4.2.2 Experimental objects

For effective and practical comparison, three groups of experimental objects are set up: Group A is the intelligent admissions consultation robot officially purchased and deployed by Dalian University of Technology. According to the university's relevant procurement announcements, this robot is based on large language models and has intelligent recommendation and question-answering functions, representing the advanced intelligent level applied in current university admissions consultation. Group B is a pure generative large language model, which directly calls the ChatGPT-3.5-turbo API (knowledge cutoff date is early 2022) without providing any external knowledge base, used to represent the performance of general AI models when separated from the latest and proprietary knowledge. Group C is the intelligent admissions question answering system designed and implemented in this study, based on the architecture of Qwen3-Guard-4B security check, Qwen3-4B-Instruct label classification, and Qwen-Plus/GPT-4 main model, and retrieves and enhances from Dalian University of Technology's latest structured admissions knowledge base from 2021 to 2023. In this experiment, Group C is the experimental group, and Groups A and B are the control groups, among which Group A represents the existing official consultation service, and Group B represents the automatic thinking and generation capabilities of AI large models.

4.2.3 Evaluation dataset

We constructed a test set containing 320 questions to ensure the comprehensiveness and challenging nature of the evaluation. Its composition includes 150 historical common questions extracted and desensitized from the university admissions office's online consultation platform logs in the past two years, representing the high-frequency needs of real users; 120 expert questions designed by 3 admissions office staff around policy difficulties, interdisciplinary comparisons, data interpretation, etc., used to test the system's deep understanding and reasoning capabilities; and 50 adversarial test questions including ambiguous questions, false premises, or involving sensitive information, specially used to test the system's boundary processing capabilities and security. All questions are evenly classified into four types: concept definition, policy query, data calculation, and complex reasoning.

4.2.4 Evaluation metrics

We adopt the following multi-dimensional metrics for evaluation: answer accuracy, independently reviewed by admissions experts to determine whether the answers are factually correct, informationally complete, and non-misleading; security compliance rate/rejection rate, evaluating whether the system correctly rejects or guides for adversarial test sets and all unanswerable questions; average response time, counting the average time-consuming from the system receiving the question to returning the final answer; citation accuracy, specifically for the Group C system, evaluating whether the answer sources provided by it truly support the answer content; dynamic personality matching rate, evaluating whether the system can automatically select an appropriate dialogue personality according to the user's emotional tendency; and user satisfaction, recruiting 30 simulated users to conduct interactions and then score from multiple dimensions using a Likert 5-point scale.

4.3 Result Analysis and Discussion

The performance comparison of the three groups of systems in various indicators is shown in Table 1. In terms of overall accuracy, the experimental group C reaches 95.6%, showing a significant advantage; the accuracy of Group A is 68.5%, especially the accuracy of data calculation questions (such as specific admission score queries) and complex reasoning questions (such as major comparison analysis) is less than 60%, reflecting its processing limitations in scenarios requiring precise matching and multi-source information integration; the accuracy of Group B is the lowest, only 70.3%, and the high accuracy shown in complex reasoning questions is obviously deceptive, and the answers often lack credibility due to factual "hallucinations".

In terms of security compliance, Group A achieves 100% compliance rate with its closed architecture and clear security strategy; the compliance rate of Group C reaches 96%, fully verifying the effective risk control ability of the "retrieval enhancement + strict instruction" framework; the compliance rate of Group B is only 32%, highlighting the potential risks of direct application of general large models in serious government affairs scenarios. In terms of efficiency, Group A has the fastest response, with an average time-consuming of less than 200 milliseconds; the average response time of the experimental group is 1.5 seconds, slightly better than Group B's 1.8 seconds, and all three meet the real-time interaction requirements. In the characteristic indicator of dynamic personality matching rate, the experimental group reaches 92.3%, which can intelligently match dialogue personalities such as "warm - caring" and "professional - rigorous" according to the user's emotional tendency. User satisfaction surveys show that the experimental group ranks significantly ahead with a comprehensive score of 4.4 points, and user feedback highly recognizes its answer characteristics of "detailed and evidence-based" and "warm and caring"; Group A only gets 3.1 points, and users generally point out that its answers are "correct in content but general in expression" and "lack personalized warmth", showing a templated tendency. Survey Scale of Various Indicators for the Consultation Platform Experiment is shown in table 1.

Table 1 Survey Scale of Various Indicators for the Consultation Platform Experiment

Evaluation Metrics	Group A	Group B	Group C
Overall Accuracy	68.5%	70.3%	95.6%
Concept Definition	82%	78%	98%
Policy Query	75%	65%	97%
Data Calculation	55%	52%	96%
Complex Reasoning	58%	88%	91%
Security Compliance Rate	100%	32%	96%
Average Response Time	< 200 ms	1.8 s	1.5 s
Citation Accuracy	90.4%	87.5%	99.2%
Dynamic Personality Matching Rate	None	None	92.3%
User Satisfaction	3.1	3.5	4.4

5 CONCLUSION

This study successfully designed and implemented an intelligent admissions Q&A system based on Retrieval-Augmented Generation (RAG) technology, addressing critical challenges such as information fragmentation, response bottlenecks, and security risks in university admissions consultation. Through a systematic framework integrating LangChain, multi-source knowledge bases, and layered model orchestration, the system achieved outstanding performance, with an overall accuracy of 95.6%, a security compliance rate of 96%, and a dynamic personality matching rate of 92.3%. Compared with existing solutions and general generative models, the proposed system significantly improved answer reliability, response efficiency, and user satisfaction. These results validate the effectiveness of the RAG-based approach in enhancing digital and intelligent services for university admissions management, providing a scalable and replicable reference for similar educational and administrative scenarios.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Karakatsoulis, Dimosthenis, Adam, et al. ChatGPT in Education: A Review of Recent Advances and Applications. INTERNATIONAL JOURNAL ON ARTIFICIAL INTELLIGENCE TOOLS, 2025(1).
[2] Li Ying. Path Exploration of Embedding Generative Artificial Intelligence Technology into Vocational College Admissions Promotional Videos. Journal of News Research, 2024, 15(8): 146-148.
[3] Shen Lin, Zhai Siyu. Research on Strategies of Empowering University Admissions with Generative Artificial Intelligence. Scientific Consult, 2025(17): 5-8.
[4] Tian Miaohua. Research on the Breakthrough Path of Local University Admissions Dilemma under the New Situation. Journal of Changchun University of Science and Technology (Social Sciences Edition), 2025, 38(05): 141-146.
[5] Du Mingyue. Research on Optimization of University Admissions Publicity Strategies from the Perspective of Digital Transformation. Journal of Jilin Radio and TV University, 2024(03): 128-130.
[6] Abdulkadir Atalan CA. The ChatGPT Application on Quality Management: A Comprehensive Review. Journal of Management Analytics, 2025, Vol.12(2): 229-259.
[7] Zhang Liping. Application of Decision Tree Algorithm in Data Visualization of University Admissions. New Technology & New Products of China, 2025, (20): 43-46.
[8] Lewis P, Perez E, Piktus A, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. 2020.
[9] Liu Xiao. Design and Implementation of Retrieval-Augmented Generation System Based on Hybrid Search and Large Model Reasoning. Huazhong University of Science and Technology, 2024.
[10] Zhou Aoying. A Beneficial Exploration of Distributed Relational Databases in the Internet Era. Journal of Computer Research and Development, 2024, 61(03): 539.
[11] Ruan Kun, Yang Jingxuan, Yin Xu, et al. Design and Implementation of AI Admissions Consultation Assistant Based on Large Language Models. Research and Exploration in Laboratory, 2025, 44(02): 110-116.
[12] Guu K, Lee K, Tung Z, et al. REALM: Retrieval-Augmented Language Model Pre-Training. 2020.
[13] Borgeaud S, Mensch A, Hoffmann J, et al. Improving Language Models by Retrieving from Trillions of Tokens. arXiv e-prints, 2021.
[14] Cheng Yun, Lv Shuang, Chen Guoxiang. Research on Intelligent Question Answering Technology for Standard Documents Based on Large Models. Information Technology & Standardization, 2024, (08): 38-43.
[15] Chen Donglei, Qin Wei. Research on Key Technologies of Intelligent Question Answering System Based on Retrieval-Augmented Generation. Office Informatization, 2024, 29(19): 82-86.

[16] Ruan Kun, Yang Jingxuan, Yin Xu, et al. Design and Implementation of AI Admissions Consultation Assistant Based on Large Language Models. Research and Exploration in Laboratory, 2025, 44(02): 110-116.
[17] Jia Chunyan, Fang Weijie, Xie Yuwei, et al. Research on Campus Question Answering System Supported by Retrieval-Augmented Generation Technology. Journal on Communications, 2024, 45(S2): 248-254.
[18] Zhao Bang, Cao Shujin. An Analysis of Knowledge Mining by Combining Knowledge Bases and AI Agents with Generative AI Large Models. Documentation, Information & Knowledge, 2025, 42(04): 88-101.
[19] Wei Hongcheng, Yang Jianlin. Key Technologies of Large Language Model + Retrieval-Augmented Method and Their Application Processes in Intelligence Tasks. Information Studies: Theory & Application, 2025, 48(03): 178-188+206.