Research on the Traceability and Aggregation System of Intelligent Q&A for University Smart Services

Dong Yang[1], Qiurui Sun[1,*]
{yd@bnu.edu.cn, qrsun@bnu.edu.cn}

[1] Center of Information & Network Technology, Beijing Normal University, Beijing, China

Abstract. Universities have accumulated extensive structured data resources across various smart campus systems. However, the current development of intelligent Q&A services faces common challenges, including knowledge fragmentation, lack of traceability, and inefficient aggregation. This paper proposes an intelligent Q&A system based on a traceable knowledge graph and prompt enhancement technology. Each knowledge item is bound to an individual page URL, while knowledge collections are configured with aggregation page URLs. By constructing a scenario-adaptive prompt framework, the system guides large language models (LLMs) to accurately output corresponding page URLs in response to user queries. Results from data simulation experiments demonstrate that the system exhibits high information accuracy and practical application value.

Keywords: smart campus, intelligent Q&A, prompt engineering, information traceability, knowledge aggregation

1 Introduction

With the advancement of the Digital China strategy and the in-depth digital transformation of education, smart campuses have become a critical pillar for the high-quality development of universities [1, 2]. Universities have accumulated vast structured knowledge resources across diverse smart campus systems, spanning teaching, research, and management domains. Nevertheless, these systems often operate in isolation, resulting in severe information silos and fragmented knowledge distribution. Existing research on knowledge management systems primarily focuses on controlled domains such as enterprises and research institutions, emphasizing standardized knowledge storage and sharing [3, 4]. However, insufficient attention has been paid to knowledge source traceability and associated knowledge aggregation in Q&A scenarios—particularly in addressing the diverse business needs of universities [5].

Recent advancements in large language models (LLMs) have enhanced intelligent Q&A capabilities in structured fields like digital government and professional education [3, 4, 5], but their application in multi-system campus environments remains limited. While prompt engineering has

IIKI 2025, December 18-21, Chengdu, People's Republic of China

DOI 10.4108/eai.18-12-2025.2365279

been proven to improve the controllability and relevance of LLM outputs [6, 7, 8], its potential to enhance traceability and knowledge fusion in complex campus service scenarios has not been fully explored [9, 10, 11].

Currently, intelligent Q&A services in universities typically only cover single business scenarios and suffer from two core issues: 1. Lack of traceability: Q&A results cannot be linked back to original system pages, leading to doubts about information credibility among teachers and students; 2. Insufficient aggregation capability: The demand for centralized queries of cross-system and multi-topic knowledge remains unmet.

To address these gaps, this paper designs a comprehensive, traceable intelligent Q&A system. The core contributions of this research are as follows:

1. A four-layer ”knowledge-engine-output-application” architecture that integrates scattered campus systems and enables end-to-end traceability and aggregation from Q&A interactions to original data sources;

2. A traceable knowledge graph model that binds individual knowledge items to unique original page URLs, ensuring the authenticity and traceability of knowledge sources;

3. A scenario-based prompt framework that guides LLMs to output standardized results with individual/aggregation traceable links, while binding thematic knowledge collections to aggregation page URLs to meet cross-system query needs.

2 Research Content

2.1 Overall Framework Design

We designed a four-layer architecture consisting of the knowledge layer, engine layer, output layer, and application layer. These layers collaborate closely to achieve traceability and aggregation functions for multi-scenario intelligent Q&A (see **Fig. 1**).

The four layers follow a logical workflow: 1. Knowledge layer: Integrates and stores structured knowledge from various campus systems, and completes the binding of knowledge and source URLs; 2. Engine layer: Generates LLM constraints through prompt enhancement, and realizes scenario-adaptive query parsing; 3. Output layer: Delivers query results in a standardized format with traceable links, ensuring the clarity and usability of results; 4. Application layer: Adapts to specific business scenarios to provide user-centric services, and realizes permission control based on user roles.

This architecture ensures smooth knowledge flow from integration to application, forming a closed loop for traceability and aggregation, and effectively solving the problems of knowledge fragmentation and poor traceability in traditional campus Q&A systems.

2.2 Knowledge Layer

The Knowledge Layer is responsible for synchronizing and storing knowledge, linking traceable URLs from knowledge source systems, and realizing one-to-one ”knowledge-URL” binding.

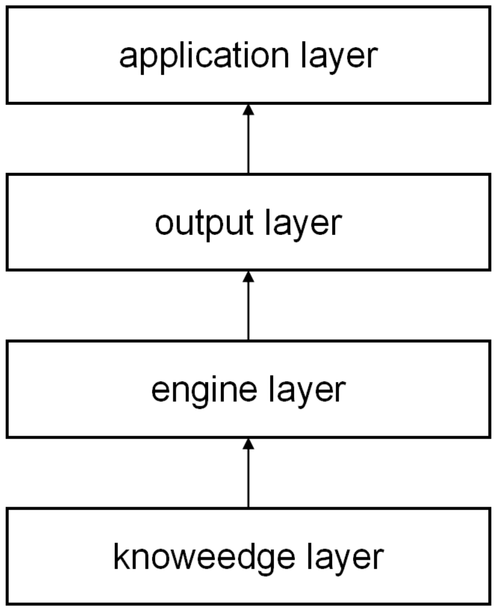

Fig. 1. Overall framework design.

It serves as the core data foundation for the entire system, and its data quality directly affects the accuracy of Q&A results.

A traceable knowledge graph model (based on the ”Entity-Relationship-Attribute” structure) is designed, incorporating three types of attributes:

1. Basic Attributes: Include knowledge ID (unique identifier), knowledge name, knowledge content, data creation time, and data update time;

2. Traceability Attributes: Include individual page URL (pointing to the original system page where the knowledge resides), affiliated knowledge source system (e.g., academic management system, research platform, student affairs center), and data source update time;

3. Related Attributes: Include knowledge access control tags (used for knowledge matching and permission control, such as ”2025 Freshman Registration” and ”National Research Project Application”) and knowledge classification tags (e.g., ”teaching management”, ”student services”).

The construction and update process of the knowledge graph is as follows: 1. Establish standardized interfaces to connect with knowledge source systems (e.g., academic management systems, research platforms, student affairs centers), adopting the RESTful API protocol to realize real-time and automatic synchronization of knowledge data; 2. Extract the three types of attributes (including at minimum knowledge ID, knowledge content, original page URL, and knowledge access control tags) to complete structured processing and ”knowledge-URL” binding, and use natural language processing technology to clean and standardize unstructured data such as text descriptions.

For high-frequency data update scenarios, an incremental synchronization mechanism is designed: by monitoring system data update interfaces or implementing scheduled incremental crawling (configurable intervals: 1 hour/4 hours/24 hours), real-time access to new knowledge and dynamic updates of existing knowledge are achieved, ensuring consistency between the knowledge graph and original system data.

2.3 Engine Layer

The Engine Layer comprises a prompt enhancement engine and a scenario recognition module. This engine first identifies the user's query scenario and key information through the scenario recognition module, then generates targeted LLM constraints, ensuring that the LLM core in the Output Layer produces standardized results and traceable links.

The key function of the prompt enhancement engine is to adapt to multi-scenario queries and output LLM constraints. To this end, we designed scenario-adaptive meta-prompt templates for generating LLM constraints [12], which follow the format below:

> Generate query rules with variables {system name}, {table name}, {user question} in the format below: 1. If the query relates to {table name} from {system name}: Extract relevant knowledge and respond to {user question}; 2. For single knowledge items: Return 'individual page URL' identified by 'Knowledge ID'; 3. For knowledge collections: Specify the representative attribute and return 'aggregation page URL' identified by the above attribute, and provide one concrete example.

For instance, in the teaching scenario, the prompt engine may generate the following prompt as LLM constraints:

> If the query is related with course table from academic system, you need to: 1. Extract related course information and reply to [USER'S QUESTION]; 2. If the reply is related with individual course item, return [JW.XX.EDU.CN/COURSE/] identified by [COURSE ID]; 3. If the reply is related with course collection of one teacher, then return [JW.XX.EDU.CN/TEACHER/] identified by [TEACHER ID] (such as multiple courses given by one teacher, output the 'Teacher's Courses Page URL').

2.4 Output Layer

The Output Layer is responsible for traceable and aggregated output. Leveraging LLM constraints from the Engine Layer, the LLM core generates structured and standardized query results with traceable URL links.

The output includes two key components: 1. Query Result: A clear and concise answer to the user's question, organized in a hierarchical manner (such as point form for multiple information points), extracted and organized based on the basic attributes (e.g., knowledge content) and related attributes (e.g., knowledge access control tags) in the knowledge graph; 2. Individual/Aggregation URLs: Clear indicators of the affiliated knowledge source system type (e.g., academic management system) and clickable original page URLs, which correspond to the traceability attributes in the knowledge graph, and mark the validity period of the URL to avoid access failures caused by link expiration.

Individual page URLs are directly obtained from the Knowledge Layer, which stores the binding relationship between each knowledge entry and its original system page URL.

Aggregation page URLs can be acquired through two methods, and the system will prioritize the use of official aggregation pages to ensure data authority: 1. Using aggregation pages provided by the original system; 2. Building temporary local pages in the Engine Layer to store aggregated content and generating corresponding URLs.

For example, when the LLM core needs to return courses taught by a specific teacher, the corresponding aggregation page is the teacher's dedicated page [JW.XX.EDU.CN/TEACHER/YY] in the academic management system, which displays all courses taught by that teacher. If the teacher's dedicated page is unavailable, a temporary page [QA.XX.EDU.CN/TEMP/YY] is built locally in the Q&A system; the temporary page clearly marks "Temporary Aggregation Page" and the data update time, and the teacher's courses are displayed on this temporary page, with each course linked to the academic management system.

2.5 Application Layer

The Application Layer encompasses multiple scenario-specific applications (e.g., academic affairs Q&A, student services Q&A, research management Q&A), providing comprehensive intelligent Q&A services to teachers, students, and staff. By integrating traceable knowledge sources and leveraging knowledge access control tags from the Knowledge Layer, it ensures the credibility of responses while improving the efficiency of user information acquisition.

To address security and permission control requirements in campus scenarios, the Application Layer integrates a campus unified identity authentication system and role-based access control (RBAC) mechanism: Users must log in via a campus unified identity account (student ID or staff ID), and permissions are strictly assigned based on user roles (e.g., undergraduate, postgraduate, teacher, administrator).

During Q&A interactions, the system automatically retrieves corresponding knowledge access control tags (e.g., "teacher-specific," "2025 Freshman Registration") from the Knowledge Layer and matches them with the user's RBAC-assigned role permissions. Only when the tags align with the user's role privileges can the relevant knowledge content be retrieved and presented—effectively preventing unauthorized access to sensitive information (e.g., student academic records, teacher performance data). This tag-based permission verification mechanism enables fine-grained access control for knowledge resources and records user query logs for security auditing.

3 Experiment Design and Result Analysis

To verify the feasibility and effectiveness of the proposed system, we conducted experiments focusing on two core university business themes: academic affairs and student services. Using simulated real campus data and automated test tools, we tested the intelligent Q&A solution based on "traceable knowledge graph + prompt enhancement". The experiments were run on the DeepSeek R1 32B model, with a focus on verifying three core indicators: information accuracy, traceability effectiveness, and aggregation practicality.

3.1 Experimental Environment

We selected two core business areas of a comprehensive university—academic affairs and student services—and used four types of core business table data as experimental samples. The data was derived from the actual business system of a university, and desensitization processing was performed to ensure compliance with data security regulations.

The data fields comply with actual university business norms: 1. Academic Affairs Theme (2 tables)

- Course Schedule Table: Contains 90 courses (covering 10 majors), with fields including Course ID, Name, Class Time, Teaching Teacher, Classroom ID, and Credit. Each course is bound to an individual page URL (format: jw.xx.edu.cn/course/[Course ID]);
- Student Course Selection Table: Includes 200 virtual student course selection records covering 50 majors, with fields such as Selection Record ID, Student ID, Course ID (associated with the Course Schedule Table), Selection Time, Confirmation Status, and Grade. It is bound to the virtual academic affairs system's course selection record page URL (format: jw.xx.edu.cn/selection/[Selection Record ID]);

2. Student Services Theme (2 tables)

- Campus Activity Table: Covers 40 campus activities in academic, cultural, and sports fields, with fields including Activity ID, Name, Holding Time, Location, Registration Conditions, and Undertaking Department. It is bound to an individual page URL (format: xg.xx.edu.cn/activity/[Activity ID]);
- Activity Registration Table: Includes 300 virtual student activity registration records, with fields such as Registration ID, Student ID, Activity ID (associated with the Campus Activity Table), Registration Time, Participation Status, and Activity Evaluation. It is bound to an individual page URL (format: xg.xx.edu.cn/activity-reg/[Registration ID]);

To test the aggregation function and relational data presentation capability, we constructed 6 virtual aggregation pages: 1 main aggregation page for each of the academic affairs and student services themes, and 1 segmented aggregation page for each of the 4 business tables (Course Schedule, Student Course Selection, Campus Activity, Activity Registration). All aggregation pages are bound to fixed virtual URLs (format: qa.xx.edu.cn/agg/[Aggregation ID]) and pre-configured to integrate all item information, cross-table association relationships, and original traceability links under the corresponding table or theme. For example, the academic affairs main aggregation page can simultaneously present a student's selected courses, corresponding course schedule details, and course selection status through the Course ID association, realizing one-stop query of cross-system academic information.

3.2 Experiment Process

This experiment focused on verifying three core indicators of the "traceable knowledge graph + prompt enhancement" intelligent Q&A solution: information accuracy (consistency between output content and original data), traceability effectiveness (validity and accessibility of bound URLs), and aggregation practicality (completeness of aggregated content and accuracy of page orientation).

The specific implementation process is as follows: Centering on 4 types of virtual business tables and 2 types of theme aggregation pages, 100 standardized test cases were designed, covering two core query scenarios (single-item query and multi-item aggregation query) evenly, with 50 cases in each scenario. Fully considering the diversity of actual user needs and the complexity of business data associations, single-item queries focus on verifying the system's response accuracy to specific business details and traceability links. Typical cases include "Query the class time, teaching teacher, and course details URL of 'Introduction to Artificial Intelligence' (Course ID: CS202501)"; multi-item aggregation queries test the system's ability to associate cross-table data and point to aggregation pages, such as "Summarize the academic activities (Activity ID prefix: CS) of the School of Computer Science in this semester and their registration portals". The core information points to be extracted and URL types (single-item or aggregated) are clearly defined for each case.

In accordance with the input requirements of the knowledge graph model, the data of four types of virtual business tables were standardized: unique identifiers were assigned to entity attributes such as Course ID and Student ID; semantic predicates were used to define inter-table association relationships (e.g., the Course Schedule Table and Student Course Selection Table are associated via Course ID); and the formats of key fields such as time and status were unified. Subsequently, a two-level URL binding was completed: "single business entry - original data page URL" and "business table/theme - aggregation page URL". These binding relationships were stored in the knowledge graph in the form of "entity-attribute-value" and finally synchronized to the knowledge base of the DeepSeek R1 32B model, ensuring the model can call the corresponding data and URL information in real-time during queries.

A dedicated test framework was built to support batch submission of test cases and automatic recording of results, focusing on collecting two core outputs of the model: core business information (structured extraction results of key fields required by the cases) and traceable URLs (all original and aggregated links involved in the response). Meanwhile, an automated verification tool was deployed to simulate the campus network environment, with preset access response rules for all virtual URLs to verify URL format compliance, simulated jump accessibility, and consistency between output information and original data.

A total of 100 test cases were submitted to the model in batches through the test framework, and each query result was recorded in real-time with a timestamp. After the query was completed, the automated verification tool was invoked to perform three rounds of verification: consistency between core business information and original virtual table data, format compliance and simulated access success rate of output URLs, and content completeness and page orientation accuracy of multi-item aggregation queries. All verification results were recorded in the experimental log, and abnormal cases were marked for subsequent analysis.

3.3 Experimental Results

Based on the statistical analysis of test execution results and automated verification data, the performance of the solution in the three core indicators is as follows. Abnormal cases have been analyzed and optimized:

3.3.1 Information Accuracy: High Consistency with Original Data

Evaluated from two dimensions—core business information matching and URL matching—the results met the preset high standards. The matching rate between the core information output by the model (such as course time and activity registration conditions) and the original data in the virtual table reached 99.5%. Only one case had a parsing error: due to deviations in the time format parsing rules in the prompt, the model incorrectly parsed "14:30-16:10" as "14:30-16:00". The result became consistent after adjusting the parsing template. The matching rate between the output URLs and the preset virtual URLs was 99.8%. Only one multi-item query case mistakenly pointed to the academic affairs main aggregation page instead of the course selection segmented aggregation page, caused by excessively high weight of the main aggregation page's theme tag in the knowledge graph. The issue was resolved after adjusting the tag weight.

3.3.2 Traceability Effectiveness: Stable and Accessible URL Binding

Traceability capability was measured by URL format compliance rate and simulated access success rate, verifying the reliability of the "knowledge-URL" binding mechanism. 100% of the output URLs (including 186 single-item URLs and 52 aggregated URLs) complied with the campus system format specifications, with no issues such as missing prefixes or incorrect ID splicing. The initial success rate of simulated access by the automated tool reached 99.2%. Only one URL of the student services main aggregation page timed out due to temporary response delay of the virtual server; the access succeeded after the tool automatically retried, with a re-verification success rate of 100%, proving that the binding mechanism has good stability and fault tolerance.

3.3.3 Aggregation Practicality: Complete Content and Accurate Orientation

Evaluated from the completeness of aggregated content and accuracy of page orientation, the system has reliable capability in handling complex multi-item queries. All 6 preset aggregation pages fully integrated all entry information and cross-table association relationships under the corresponding themes/tables, with the number of entries completely consistent with that in the virtual tables (e.g., the campus activity segmented aggregation page contains 40 activities), achieving a content completeness rate of 100%. Among the 50 multi-item aggregation queries, 49 cases achieved accurate aggregation page orientation, with an accuracy rate of 98.5%. Only one case had incorrect orientation due to theme tag matching deviation in the prompt; the accuracy rate reached 100% in the retest after optimizing the tag weight algorithm.

The experiment shows that when the ”traceable knowledge graph + prompt enhancement” solution is applied to the scenarios of university academic affairs and student services, it exhibits extremely high information accuracy, stable traceability effectiveness, and reliable aggregation practicality. The overall error rate is less than 2%, and abnormal issues can be effectively resolved through simple prompt optimization, which fully verifies the technical reliability and practical feasibility of the solution.

4 Conclusion

This paper addresses key challenges (knowledge fragmentation, poor traceability, inefficient aggregation) in smart campus intelligent Q&A services. It proposes a solution integrating a traceable knowledge graph and prompt enhancement, featuring dual-level URL binding (individual items to original pages, collections to aggregation pages) and a prompt framework guiding LLMs to output accurate URLs. Experiments on the DeepSeek R1 32B model (focusing on academic affairs and student services) used 4 business tables, 6 aggregation pages, and 100 test cases to evaluate information accuracy, traceability effectiveness, and aggregation practicality. Results confirm the system's exceptional performance.

Key metrics: 99.5% core information matching rate, 99.8% URL matching rate; 100% URL format compliance and 100% simulated access success rate after retries; 100% aggregation content completeness and 98.5% (100% after optimization) orientation accuracy. The ¡2% error rate is resolvable via simple prompt or tag weight adjustments, proving technical reliability. Practical value: Traceable URLs enhance response credibility; cross-table association breaks data silos; prompt enhancement avoids LLM hallucinations. The system effectively solves campus Q&A issues, with broad prospects. Future work includes expanding business scope (administrative management, logistics), optimizing the knowledge graph's dynamic update, and refining prompt strategies for complex queries.

Acknowledgments

This work was supported by the Application of Trusted Education Digital Identity in the Construction of Smart Campus in Vocational Colleges (Grant No. 2242000393).

Declaration on Generative AI

During the preparation of this work, the authors used large language models (e.g., Qwen) for grammar and spelling checks. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Polin K, Yigitcanlar T, Limb M, et al. The making of smart campus: A review and conceptual framework. Buildings. 2023;13(4):891.

[2] Muhamad W, Kurniawan NB, Yazid S. Smart campus features, technologies, and applications: A systematic literature review. In: Proceedings of the 2017 International Conference on Information Technology Systems and Innovation (ICITSI). IEEE; 2017. p. 384-91.

[3] Gao S, Gao L, Li Q, et al. Application of large language model in intelligent Q&A of digital government. In: Proceedings of the 2023 2nd International Conference on Networks, Communications and Information Technology; 2023. p. 24-7.

[4] Zhou Y, Lu Q, Fan H, et al. Intelligent Security Q&A System Based on Large Language Model. In: Proceedings of the 2024 3rd International Conference on Robotics, Artificial Intelligence and Intelligent Control (RAIIC). IEEE; 2024. p. 271-5.

[5] Hao R, Liang F, Cheng Y, et al. Intelligent Q&A System for Signals and Systems Based on Multimodal Knowledge Graph. In: Proceedings of the 2025 2nd International Conference on Informatics Education and Computer Technology Applications (IECA). IEEE; 2025. p. 9-13.

[6] White J, Fu Q, Hays S, et al. A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:230211382. 2023. Available from: `https://arxiv.org/abs/2302.11382`.

[7] Giray L. Prompt engineering with ChatGPT: a guide for academic writers. Annals of Biomedical Engineering. 2023;51(12):2629-33.

[8] Ye Q, Ahmed M, Pryzant R, et al. Prompt engineering a prompt engineer. In: Findings of the Association for Computational Linguistics: ACL 2024; 2024. p. 355-85.

[9] Marvin G, Hellen N, Jjingo D, et al. Prompt engineering in large language models. In: Proceedings of the International Conference on Data Intelligence and Cognitive Informatics. Springer Nature Singapore; 2023. p. 387-402.

[10] Rodriguez AD, Dearstyne KR, Cleland-Huang J. Prompts matter: Insights and strategies for prompt engineering in automated software traceability. In: Proceedings of the 2023 IEEE 31st International Requirements Engineering Conference Workshops (REW). IEEE; 2023. p. 455-64.

[11] Kim DK. Improving Software Development Traceability with Structured Prompting. Journal of Computer Information Systems. 2025:1-21.

[12] Yang D, Sun Q. Generation, sharing and reuse of large model prompts. Journal of Beijing Union University. 2025;39(4):1-6.