BUILDING AN INTELLIGENT UNIVERSITY ADMISSION COUNSELING SYSTEM USING LARGE LANGUAGE MODELS AND KNOWLEDGE GRAPHS

Tong Gia Bao[1], Tieu Xuan Hoang[2], Le Hoan[1,*]

DOI: https://doi.org/10.57001/huih5804.2026.076

ABSTRACT

University admission counseling is a critical process that involves processing complex, interconnected data regarding academic majors, admission criteria, tuition fees, and career prospects. Traditional consulting systems at universities in Vietnam, including Electric Power University (EPU), primarily rely on keyword-based search or manual consultation, leading to information fragmentation and a lack of personalized guidance. This paper presents an experimental admission counseling support system that integrates Large Language Models (LLMs) with a domain-specific Knowledge Graph (KG). We propose a Single Agent architecture where an LLM functions as a central reasoning unit to orchestrate queries over a Neo4j-based Knowledge Graph constructed from EPU's admission data. This approach enables the system to perform semantic reasoning, handle multi-hop queries (e.g., matching aptitude scores with career goals), and provide accurate, context-aware advice. Initial experimental designs suggest that this neuro-symbolic approach significantly improves the relevance and depth of counseling responses compared to traditional SQL-based methods.

***Keywords*:** *LLMs (Large Language Models), University Admission System, Knowledge Graphs, Retrieval-Augmented Generation (RAG), Educational Decision Support Systems, Intelligent Admission Counseling.*

[1]Faculty of Information Technology, Electric Power University, Vietnam
[2]Faculty of New Energy, University of Electric Power
[*]Email: hoanle @epu.edu.vn

Received: 02/01/2026
Revised: 10/3/2026
Accepted: 30/3/2026

1. INTRODUCTION

Effective university admission counseling is a cornerstone of higher education systems worldwide, guiding prospective students through complex decisions about programs, majors, admission criteria, and career pathways. Traditional counseling methods whether in person or via simple FAQ systems are limited by scalability, a lack of personalized interaction, and rigid query patterns that do not align with the way students naturally express concerns or preferences. With rising application volumes and increasing demand for real-time assistance, there is an urgent need for intelligent, scalable, and context-aware admissions support systems.

Existing university admission counseling systems still exhibit several critical limitations, particularly in the context of increasingly diverse and personalized student inquiries. Most university admission portals in Vietnam rely primarily on static FAQ pages, keyword-based search engines, or manually maintained consultation workflows. Such systems are often unable to support natural conversational interaction, contextual follow-up questions, or personalized reasoning based on student profiles and career preferences. Traditional relational database or SQL-based retrieval approaches are effective for exact information lookup but perform poorly when handling semantically complex or multi-hop advisory queries. For example, questions such as "Which engineering major is suitable for a student interested in renewable energy and strong in mathematics?" require reasoning across multiple interconnected concepts including majors, admission methods, subject combinations, and career outcomes.

Furthermore, existing retrieval-based chatbots typically operate on isolated text chunks without explicitly modeling semantic relationships between educational entities. This limitation often leads to fragmented responses, weak contextual coherence, and difficulties in maintaining conversational continuity across multiple dialogue turns. Another important limitation is the lack of explainability and grounding in

many generative AI-based counseling systems. Large language models may generate plausible but factually incorrect information, particularly for numerical admission data such as tuition fees, quotas, and admission scores. In high-stakes educational decision-making scenarios, such hallucinations can significantly reduce user trust and system reliability.

Recent advances in large language models (LLMs) such as GPT-4 and other generative pretrained architectures have demonstrated remarkable capabilities in natural language understanding and generation, enabling conversational interfaces that can engage users in human-like dialogue and handle unstructured user input more effectively than rule-based systems. LLMs have been explored for a variety of educational applications, from tutoring to automated guidance, highlighting their potential to reduce counsellors' workload and enhance access to information for students and families alike [1]. However, LLMs alone suffer from well-documented limitations, including hallucination (generation of plausible but incorrect information) and a lack of grounded, verifiable knowledge for domain-specific queries. While LLMs excel in language fluency, their internal parametric knowledge is not explicitly tied to structured datasets, which can undermine factual accuracy in high-stakes domains such as university admissions. This constraint poses risks when the system is relied upon for critical decisions about educational pathways.

To address these issues, recent research has turned to knowledge graphs (KGs) structured representations of domain knowledge that encode entities and their relationships explicitly. Knowledge graphs provide reliable context and verifiable information that can anchor generative models, improving both accuracy and traceability of responses [2]. Hybrid approaches combining LLMs with KGs are emerging as a promising paradigm, leveraging the linguistic and dialogic strengths of LLMs with the precision and structure of KGs to facilitate more trustworthy and context-aware interactions. Such neuro-symbolic integrations have been shown to enhance explainability, personalization, and overall system performance in educational settings, including tutoring and learning-path recommendations [3].

In the specific context of university admissions, systems that integrate structured admissions data such as program requirements, historical admission scores, student profiles, and career outcomes with conversational AI can provide richer, more personalized guidance than traditional tools. Recent work has demonstrated that KG-guided LLM frameworks can significantly improve the relevance and accuracy of university and major recommendations compared to standalone LLM or rule-based methods [4] .

Despite this emerging interest, there remains a gap in research on comprehensive, real-world implementations of LLM-KG hybrid systems tailored for university admission counseling, particularly those evaluated in operational settings with actual student users. This paper aims to fill that gap by proposing a hybrid architecture that combines generative LLMs with curated admission knowledge graphs to create an intelligent counseling system. We further evaluate this system in the practical context of Electric Power University, demonstrating improvements in accuracy, personalization, and user satisfaction over baseline approaches.

The novelty of this work does not merely lie in the integration of LLMs and Knowledge Graphs, which has been explored in previous studies, but rather in the design of a domain-aware GraphRAG framework specifically optimized for conversational university admission counseling.

The main methodological contributions of this paper are summarized as follows:

- We propose a lightweight single-agent GraphRAG architecture that unifies intent understanding, Cypher query generation, graph reasoning, and response synthesis within a coherent reasoning pipeline, reducing orchestration complexity compared to multi-agent approaches.

- We introduce a schema-guided Cypher generation mechanism that constrains LLM outputs using domain ontology and conversational context, improving query reliability and reducing hallucination during graph retrieval.

- We integrate both short-term and long-term conversational memory into the GraphRAG pipeline, enabling contextual follow-up reasoning across multiple admission counseling turns.

- We design a domain-specific university admission ontology for Vietnamese higher education, supporting multi-hop reasoning over majors, admission methods, tuition, subject groups, and career opportunities.

- We implement and experimentally evaluate the proposed framework in a real-world deployment scenario at Electric Power University.

2. RELATED WORK

In this section, we review prior research that is directly relevant to the key components of our proposed system: LLM-based educational systems, hybrid LLM + Knowledge Graph frameworks, and intelligent counseling/advising systems in educational contexts. The review highlights specific limitations in each area that motivate the need for integrated, hybrid approaches.

Large language models have been widely studied for their potential in educational applications. A recent comprehensive survey by Liu et al. discusses the opportunities and challenges of applying LLMs across educational domains, including student and teacher support, adaptive learning, and automated assistance. However, the survey emphasizes that LLMs alone lack systematic grounding in structured domain knowledge, which can lead to unreliable guidance in complex educational scenarios such as admissions counselling [5]. Although LLMs can interpret natural language effectively and facilitate dialogue, their internally stored representations are static and prone to generate hallucinations plausible but incorrect responses particularly when handling technical or domain-specific queries that require precise, structured knowledge.

The synergistic integration of knowledge graphs with LLMs has attracted increasing research attention as a solution to mitigate the limitations of standalone generative models. Knowledge graphs provide structured, semantically rich representations that can ground language generation, support reasoning, and reduce hallucinations.

Leng et al. [4] recently proposed a hybrid university and major recommendation framework that constructs a structured knowledge graph of admission data and uses it to constrain and guide LLM-based recommendations. Their work demonstrated improved precision and relevance in recommendations compared to purely LLM-based systems but was limited to university-major selection contexts and did not address conversational interaction or personalized counseling flows extensively.

Beyond domain-specific applications, broader research investigates LLM+KG synergies for question answering. For example, Chuangtao et al. [6] surveyed multiple methods by which knowledge graphs can serve as background knowledge, reasoning guides, and refiners for LLMs in QA tasks. While this work highlights the potential of KG augmentation to enhance correctness and explainability, it focuses primarily on QA benchmarks rather than personalized guidance or dialogue systems tailored to admissions queries.

Similarly, methodological studies such as Li et al.'s investigation into Retrieval-Augmented Generation (RAG) highlight how external knowledge sources can enhance the factual basis of LLM outputs. Nevertheless, these approaches often rely on text retrieval rather than rich, domain-structured graphs, limiting their ability to represent complex relations inherent in admissions decision frameworks [7].

Knowledge graphs on their own have been explored in educational contexts for structuring curricular components and supporting personalized learning guidance. Wang et al. developed an intelligent tutoring model that integrates course knowledge graphs with LLMs to improve domain-specific question answering and student-centered learning outcomes. This work shows that structured educational knowledge can enhance LLM accuracy and domain alignment, but it does not target admission counseling scenarios where the knowledge domain and reasoning requirements differ substantially [8].

Abu-Rasheed et al. [9] proposed the use of knowledge graphs as context sources for LLM-based explanations of learning recommendations. Their approach reduced hallucination risk by grounding LLM prompts in verified knowledge graph contexts, improving precision and relevance of guidance compared to LLM-only baselines. However, the focus was on learning recommendation explanations, not higher-level decision support for university admissions.

Studies on AI-driven academic advising systems suggest that automation can improve access to guidance and reduce dependency on human advisors. Abdelhamid et al. reviewed the application of AI and LLMs in academic advising and highlighted their potential in personalized support, but also pointed out challenges related to engagement and authenticity of advice when systems rely solely on generative models without structured reasoning or factual grounding [10]. Moreover, hybrid conversational systems such as URAG combine rule-based and retrieval approaches to enhance educational chatbots, demonstrating competitive performance against commercial models on real admission question datasets. Nevertheless, these systems still lack explicit domain knowledge structures capable of supporting multi-step reasoning over admission criteria, program attributes, and student profiles [11].

Traditional Retrieval-Augmented Generation (RAG) systems primarily rely on vector similarity search over unstructured textual chunks. While effective for general document retrieval, such approaches often struggle with multi-hop reasoning and relational queries commonly found in university admission counseling scenarios.

Recent GraphRAG frameworks improve retrieval by leveraging graph-structured knowledge representations. However, many existing GraphRAG systems focus mainly on generic question answering tasks and employ either static retrieval pipelines or complex multi-agent orchestration architectures.

In contrast, the proposed framework introduces several domain-specific enhancements:

- A schema-guided Cypher generation mechanism for reliable graph querying,

- Integrated conversational memory for follow-up admission counseling,

- Lightweight single-agent orchestration for reduced architectural complexity,

- Ontology-aware multi-hop reasoning tailored to university admission decision support.

Unlike conventional RAG systems that retrieve isolated text chunks, our approach performs explicit relational reasoning over structured university admission entities and relationships stored in Neo4j.

Although previous studies have explored the integration of LLMs and Knowledge Graphs for question answering and educational recommendation systems, the proposed work differs from existing approaches in several important aspects. First, unlike traditional Retrieval-Augmented Generation (RAG) systems that primarily rely on vector similarity search over unstructured text chunks, the proposed framework performs explicit graph-based reasoning over structured admission entities and semantic relationships stored in Neo4j. Second, many existing GraphRAG systems focus mainly on generic question answering tasks or recommendation problems, whereas our work specifically targets conversational university admission counseling, which requires contextual reasoning, personalized advising, and multi-turn interaction management. Third, the proposed architecture adopts a lightweight single-agent reasoning framework that unifies intent understanding, Cypher query generation, graph retrieval, and answer synthesis within a coherent pipeline. Compared to multi-agent orchestration architectures, this design reduces system complexity and improves deployment efficiency. Fourth, the system integrates both short-term and long-term conversational memory mechanisms to preserve dialogue context and support follow-up counseling queries, which are often overlooked in existing educational GraphRAG systems. Finally, this work provides a real-world implementation and evaluation using admission data from Electric Power University, demonstrating the practical feasibility of domain-specific GraphRAG systems in higher education environments.

In summary, existing research demonstrates that:

- LLMs offer strong natural language capabilities but are insufficient for domain-specific precision without external grounding.

- Knowledge graphs improve structure and correctness but, when used alone, do not address conversational complexity.

- Hybrid LLM + KG systems show promise, yet current work is either limited to recommendation tasks or QA benchmarks, with little work on integrated counseling systems tailored to university admissions.

These gaps motivate our research on a KG-guided LLM counseling system that combines dialogue flexibility, structured domain knowledge, and personalized decision support, rigorously evaluated in a real university context.

3. SYSTEM OVERVIEW AND ARCHITECTURE

3.1. Overall architecture

Our proposed system operates on a Retrieval-Augmented Generation (RAG) paradigm, specifically adapted for graph databases (GraphRAG), as shown in Figure 1. The core architecture consists of three main pillars: (1) Knowledge Graph Construction Pipeline, (2) Intelligent Query Routing Mechanism, and (3) Single Agent Reasoning Framework.

The system is designed as a neuro-symbolic framework where a Large Language Model (LLM) functions as the central reasoning unit. It orchestrates queries over a Neo4j-based Knowledge Graph constructed from Electric Power University (EPU)'s admission data. This architecture enables the system to perform semantic reasoning, handle multi-hop queries (e.g., matching aptitude scores with career goals), and provide accurate, context-aware advice, significantly improving relevance compared to traditional SQL-based methods.

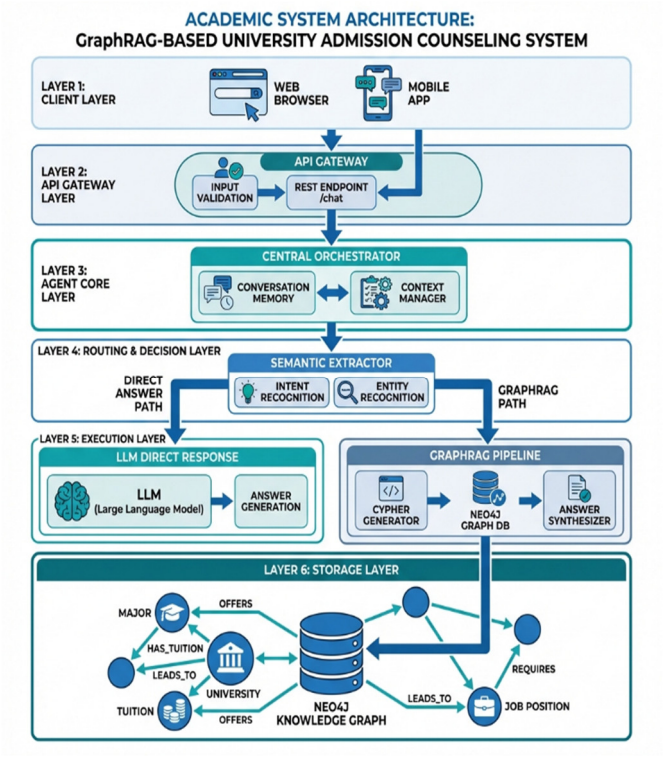

Figure 1. Overview of GraphRAG-based admission counseling system

In this implementation, the LLM component is deployed through an API-based configuration rather than local model hosting. The system uses GPT-4o-mini as the primary language model for intent understanding, Cypher query generation, entity-relation extraction, and answer synthesis. The model was selected due to its balance between response quality, latency, and deployment cost. The LLM is accessed through a RESTful API service, and all prompts are constructed using predefined templates that include the graph schema, task instruction, conversation context, and retrieved graph facts. The temperature parameter is set to 0.2 for Cypher generation and factual answer synthesis to reduce randomness, while a slightly higher temperature of 0.4 is used for natural conversational responses. No fine-tuning was conducted in this study; instead, the system relies on schema-guided prompting and few-shot examples to constrain model behavior.

3.2. Knowledge graph construction

To transition from unstructured university documents (PDFs, HTML) to structured knowledge, we employ a semi-automated pipeline consisting of Schema Definition, Extraction, and Resolution.

3.2.1. Graph schema definition

We define the ontology schema S with the set of entity types T_V and relation types T_E:

$$\mathcal{T}_V = \begin{Bmatrix} University, Major, AdmissionMethod, \\ SubjectGroup, Course, JobPosition, Tuition \end{Bmatrix}$$

where each entity type represents a core concept in the university admission ecosystem:

- $University$: An educational institution offering undergraduate programs.

- *Major*: An academic program or field of study provided by a university.

- *AdmissionMethod*: A specific admission pathway, such as exam-based admission or transcript-based admission.

- *SubjectGroup*: A group of subjects used for admission evaluation (e.g., A00, D01).

- *Course*: A compulsory or elective subject associated with a major.

- *JobPosition*: A potential career outcome linked to a major.

- *Tuition*: Tuition fee information associated with a major or university.

The set of relation types T_E captures the semantic relationships between entities, enabling structured reasoning and multi-hop retrieval over the knowledge graph. Typical relations include associations between universities and majors, majors and courses, majors and job positions, as well as admission methods and subject groups.

This ontology scheme serves as the foundation for knowledge graph construction and ensures that all downstream reasoning and query generation processes are grounded in a consistent and well-defined domain structure, as illustrated in Figure 2.

The end-to-end processing workflow of the proposed GraphRAG-based counseling system is illustrated in Figure 2 and consists of six main stages. First, the user submits a natural language admission query through the conversational interface. The query, together with the recent conversation history stored in Short-Term Memory (STM), is forwarded to the LLM-based reasoning module. Second, the reasoning module performs intent understanding and entity recognition to identify the semantic requirements of the query, such as target majors, admission methods, tuition information, or career-related concepts. Third, based on the predefined ontology schema and the conversational context, the LLM generates a Cypher query for retrieving relevant information from the Neo4j Knowledge Graph. The

schema-guided prompting mechanism constrains the query generation process to ensure structural correctness and reduce hallucination. Fourth, the generated Cypher query is executed against the Knowledge Graph database. The graph retrieval module returns a structured set of nodes, relationships, and attributes associated with the user query. Fifth, the retrieved graph information is combined with the original user query and the conversational context to construct a grounded synthesis prompt. The LLM then generates a context-aware response that integrates both symbolic graph knowledge and natural language reasoning.

Finally, the generated response is returned to the user and simultaneously stored in both Short-Term Memory and Long-Term Memory components to support future conversational continuity and analytics.

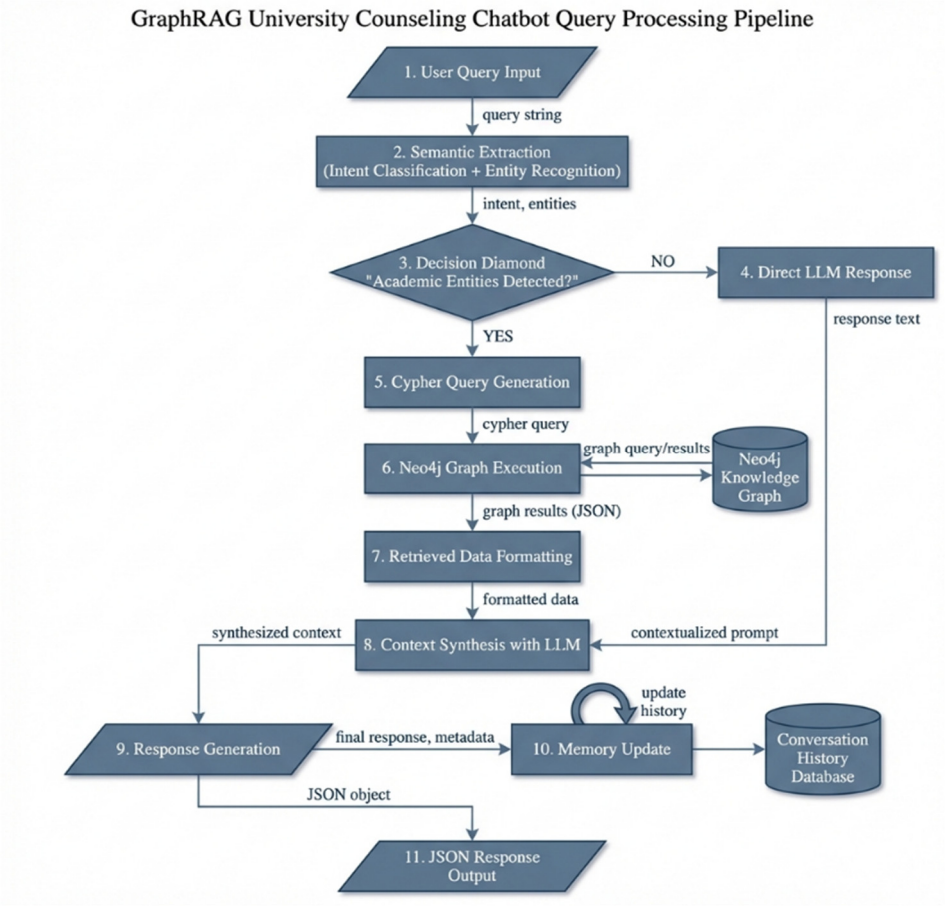

Figure 2. Query processing pipeline of GraphRAG

3.2.2. Semantic chunking & extraction

Raw text documents T(d) are segmented into semantic chunks using a sliding window approach with a boundary scoring function β(p) to preserve context:

$$C_d = \text{Chunk}\,(T_d, w_{max}) = \{c_1, c_2, \ldots, c_n\} \quad (1)$$

where:

- w_{max} the maximum chunk size (in tokens),
- ω is the overlap ratio between consecutive chunks,
- c_i denotes the i-th semantic chunk.

In the above equations, d denotes an input admission document, T(d) represents the textual content extracted from document d, and ci denotes the i-th semantic chunk. The parameter Wmax defines the maximum allowed chunk length measured in tokens, while ω denotes the overlap ratio between two consecutive chunks.

To preserve semantic coherence, chunk boundaries are selected using a boundary scoring function β(p), which prioritizes meaningful structural positions in the document.

$$\beta(p) = \lambda_1 \cdot I_{paragraph}(p) + \lambda_2 \cdot I_{sentence}(p) + \lambda_3 \cdot I_{table}(p) \quad (2)$$

In this formulation:

- p is a candidate boundary position,
- $I_{type}(p)$ is an indicator function returning 1 if position p matches the given structural type, otherwise 0,
- $\lambda_1 > \lambda_2 > \lambda_3$ enforce higher priority for paragraph boundaries over sentence and table boundaries.

The function β(p) is a boundary scoring function used to determine whether position p is a suitable segmentation point. Paragraph, sentence, and table boundaries are assigned different weights through λ_1, λ_2, and λ_3, respectively, where $\lambda_1 > \lambda_2 > \lambda_3$ indicates that paragraph boundaries are preferred over sentence and table boundaries.

Each resulting chunk maintains sufficient contextual information while remaining suitable for downstream LLM-based extraction.

3.3. LLM-based entity and relation extraction

The entity and relation extraction module does not rely on a supervised training dataset. Instead, we adopt a prompt-based extraction strategy using the predefined

admission ontology as a constraint. For each semantic chunk, the LLM is instructed to extract only entities and relations that belong to the predefined schema. To improve consistency, we designed a set of few-shot extraction examples based on manually selected admission documents from Electric Power University. These examples include typical entity types such as Major, AdmissionMethod, SubjectGroup, Tuition, Course, and JobPosition, as well as relations such as offers Major, *usesAdmissionMethod*, *requiresSubjectGroup*, *hasTuition*, *includesCourse*, and *leadsToJobPosition*. The extraction output is required to follow a structured JSON format. Each extracted entity contains its name, type, source chunk, and confidence score, while each relation is represented as a triple consisting of source entity, relation type, and target entity. Invalid entities and relations that do not match the ontology schema are filtered out before being inserted into Neo4j.

For each semantic chunk c_i, a Large Language Model is employed to perform entity extraction and relation extraction under a predefined schema.

Entity extraction

The entity extraction process maps a chunk to a set of typed entities:

$$E_i = LLM_{NER}\,(c_i, T_V) \tag{3}$$

where E_i is the set of extracted entities from chunk c_i and each entity is assigned a type from T_V.

Relation extraction

After entities are identified, semantic relations between them are inferred:

$$R_i = LLM_{RE}\,(c_i, E_i, T_E) \tag{4}$$

Each relation is represented as a triple (es, r, et), where:

- es is the source entity,
- et is the target entity,
- r in is the relation type.

This two-stage extraction allows the system to populate both nodes and edges of the Knowledge Graph in a structured manner.

3.4. Entity resolution and graph assembly

Since the same real-world entity may appear multiple times across different documents or chunks, an entity resolution step is required to merge duplicates.

We compute a similarity score between two entities e_1 and e_2 using a weighted combination of lexical and semantic similarity:

$$sim(e_1, e_2) = \alpha \cdot sim_{lex}(e_1, e_2) + (1-\alpha) \cdot sim_{sem}(e_1, e_2) \tag{5}$$

In the entity resolution formula, sim(e_i, e_j) denotes the similarity score between two candidate entities e_i and e_j. The function sim_{lex} measures lexical similarity based on entity names, while simsem measures semantic similarity using vector embeddings. The parameters α and 1 - α control the relative importance of lexical and semantic similarity. Two entities are merged if their similarity score is greater than or equal to the threshold τ.

Entities are merged if the similarity score exceeds a predefined threshold:

$$sim(e_1, e_2) > \theta_{merge} \tag{6}$$

After resolution, all validated entities and relations are assembled into the final Knowledge Graph G, which is then persisted in a Neo4j database for efficient querying and retrieval.

3.5. GraphRAG execution pipeline

When the routing module selects the $graphRag$ execution path, the system activates a three-stage $graphRag$ pipeline, including:

$(1) NaturalLanguagetoCypherTranslation,$

$(2)\ Graph\ Query\ Execution,$

$(3) Context - AwareAnswerSynthesis.$

3.5.1. Natural language to cypher translation

The query generation module translates a natural language query into Cypher, the declarative query language of Neo4j.

This translation is performed by a Large Language Model with schema-aware prompting and conversational context.

$$Cypher(q) = LLM_{query(q,S,Examples,M_t)} \tag{7}$$

where:

- q is the user query,
- S is the graph schema,
- $Examples$ are few-shot Cypher examples,
- M_t represents the conversation context at time step t, which provides disambiguation cues for follow-up questions.

The prompt template explicitly includes the graph schema and example mappings from questions to Cypher queries to constrain the model output.

3.5.2. Graph query execution

The generated Cypher query is executed against the Neo4j database using a connection pooling mechanism to ensure efficiency and scalability.

$$D_G = Neo4j.Execute\big(Cypher(q)\big) = \{r_1, r_2, \dots, r_n\} \quad (8)$$

where:

- D_G denotes the graph query result set,

- Each record r_i contains matched node properties and relationship attributes.

To prevent resource exhaustion, the execution process enforces both timeout and result-size constraints.

$$\mathrm{D}_{\mathrm{G}_{\mathrm{safe}}} = \mathrm{Truncate}(D_G, \mathrm{n}_{\max})\ subject\ to\ \mathrm{t}_{\mathrm{exec}} \leq t_{max} \quad (9)$$

In our implementation:

- maximum execution time t_{max}= 10s,

- maximum number of returned records $\mathrm{n}_{\max}$= 100.

3.5.3. Context-aware answer synthesis

The answer synthesis stage integrates the retrieved graph data with the original query and the conversation context to generate a coherent and informative response.

$$a = LLM_{synth}\big(q, D_G, M_t, P_{synth}\big) \quad (10)$$

where P_{synth} is the synthesis prompt that controls grounding behavior.

To handle cases where the graph query returns no results, a fallback mechanism is applied.

$$P_{synth} = \big\{P_{grounded}\ if\ |D_G| > 0\ P_{fallback}\ if\ |D_G| = 0\big\} \quad (11)$$

- $P_{grounded}$ enforces strict grounding on retrieved graph facts.

- $\mathrm{P}_{\mathrm{fallback}}$ allows general domain knowledge with explicit disclaimers.

3.6. Conversation memory management

To maintain contextual understanding across multiple conversation turns, the system incorporates both Short-Term Memory (STM) and Long-Term Memory (LTM).

3.6.1. Short-Term Memory (STM)

STM is implemented as a fixed-size sliding window buffer that stores the most recent k user–system interactions.

$$M_{STM(t)} = \big\{\big(q_{\{t-k+1\}}, a_{\{t-k+1\}}\big), \dots, (q_t, a_t)\big\} \quad (12)$$

In our system, the window size is set to k = 10. The STM content is formatted into a linear context string before being passed to the LLM.

3.6.2. Long-Term memory (LTM)

Long-Term Memory persists historical conversation data into the Neo4j graph, enabling semantic retrieval and analytics across sessions.

Each conversation turn is stored as a node with the following structure:

$$v_{conv} = \begin{pmatrix} session_{id}, timestamp, query, \\ response, intent, entities \end{pmatrix} \quad (13)$$

This design supports:

- session reply,

- keyword-based retrieval of past interactions,

- statistical analysis of intent and entity distributions

3.7. System integration and API design

The complete system is exposed through a RESTful API with a single endpoint:

$$POST_{chat}: (query, session_{id}) \rightarrow (answer, cypher, timing, metadata) \quad (14)$$

The API response includes detailed timing metrics for transparency and performance monitoring.

The total end-to-end latency is bounded by:

$$T_{total} \leq T_{route} + \max\begin{pmatrix} T_{direct}, T_{cypher} \\ +T_{neo4j} + T_{synth} \end{pmatrix} \quad (15)$$

Empirical evaluation shows that:

$$T_{total} < 8\,s\ \ for\ 70\%\ of\ queries \quad (16)$$

4. EXPERIMENTAL TESBED

4.1. Dataset construction

To evaluate the system's performance comprehensively, we constructed a diverse dataset consisting of 100 test queries (denoted as test_dataset_100).

The evaluation dataset was manually constructed to simulate realistic university admission counseling scenarios encountered by prospective students and parents at Electric Power University. The dataset generation process consisted of three main stages:

First, admission-related information was collected from official EPU sources, including admission brochures, university websites, tuition announcements, program descriptions, admission regulations, and career orientation materials. Second, candidate queries were designed based on frequently asked admission questions

collected from university counseling activities, online admission forums, and social media interactions during recent admission seasons. The questions were formulated to reflect natural conversational language rather than template-based search queries. Third, the generated queries were manually reviewed and categorized by two domain experts with experience in university admission counseling and educational information systems. The reviewers verified that each query was semantically meaningful, grammatically valid, and relevant to real-world counseling scenarios.

The difficulty levels were determined according to the reasoning complexity required to answer each query. The queries were categorized by difficulty level and user intent as follows:

- Easy (40%): Easy queries involve single-hop factual retrieval tasks requiring direct lookup of one entity or attribute from the knowledge graph.

Factoid questions requiring single-hop retrieval, for example: "*What is the tuition fee for IT?*"

- Medium (40%): Medium queries require multi-attribute retrieval, filtering, or comparison across related entities.

Queries involving specific admission criteria or multi-fact lookup, for example: "*Admission score for Marketing in 2023 versus 2024.*"

- Hard (20%): Hard queries involve multi-hop semantic reasoning, contextual interpretation, or personalized advisory generation that combines multiple graph entities and conversational constraints.

Examples of hard queries include career-oriented recommendations, comparative major analysis, and follow-up conversational questions requiring contextual memory.

Queries requiring complex reasoning, comparison, or advisory capabilities, for example:*"Compare job opportunities between Software Engineering and Computer Science for a student who likes mathematics."*

This dataset was designed to reflect realistic student inquiries while covering a broad spectrum of reasoning complexity.

For evaluation purposes, each query was associated with an expected reference answer derived from official university admission documents. The correctness of system responses was manually assessed by two independent evaluators based on factual accuracy, contextual relevance, and completeness. Disagreements between evaluators were resolved through discussion to ensure consistency in the final assessment.

4.2. Evaluation metrics

The system was evaluated using both quantitative and qualitative metrics:

- Success Rate (SR): The percentage of queries that returned a valid, non-error response (*HTTP status code 200*).

- Average Latency (AL): The average time required to process a query and generate a response.

- Intent Accuracy: A measure of whether the system correctly identified the user's intent category, such as Admission, Major Information, or Comparison. Moreover, Intent Accuracy measures the ability of the system to correctly identify the semantic intention of a user query before graph retrieval and response generation.

For evaluation purposes, each query in the dataset was manually assigned a ground-truth intent label by two domain experts. The intent taxonomy includes categories such as Admission Information, Tuition Inquiry, Major Information, Career Orientation, Comparison Query, and Personalized Counseling. Given a query qi, the intent prediction is considered correct if the predicted intent category matches the manually assigned ground-truth label.

The intent accuracy is computed as:

Intent Accuracy = Number of correctly predicted intents / Total number of queries

The final intent accuracy score is reported as the percentage of correctly classified queries over the entire evaluation dataset.

4.3. Performance overview

We conducted an automated benchmark using the 100-query evaluation dataset as illustrated in Table 1. The system achieved an overall accuracy of 70%, with an average latency of 7.15 seconds per query.

Table 1. System performance by query difficulty (N = 100)

Difficulty	Count	Average Latency (s)	Success Rate (%)
Easy	40	6.85	75.0
Medium	40	7.20	67.5
Hard	20	7.85	65.0
Overall	100	7.15	70.0

While the system demonstrated strong performance for general and explanatory queries, it showed noticeable limitations when handling precise numerical information.

4.4. Analysis and error modes

The experimental results reveal a clear dichotomy in system performance.

Strengths

The single-agent architecture performs particularly well on definitional and procedural queries, such as:

- "What is the admission process?"
- "List the core subjects of the IT major."

In these cases, the semantic reasoning capability of the large language model enables it to generate well-structured and coherent responses. Qualitative human evaluation frequently rated such answers between 9/10 and 10/10.

Weaknesses

The primary source of failure, accounting for approximately 30% of errors, is numerical hallucination. When responding to queries requiring exact scalar values (for example, "Tuition fee for 2025" or "Admission quota"), the model occasionally produced outdated or approximate values instead of the correct fixed numbers. This behavior suggests that although the Knowledge Graph schema is well-designed, retrieval precision for numerical attributes requires further improvement. One promising direction is to enforce stricter constraints during Cypher query generation to ensure exact value matching.

The most common error category is numerical hallucination, where the LLM generates approximate or outdated numerical values despite the existence of correct graph data. This issue mainly occurs during the response synthesis stage when the model attempts to produce fluent natural language responses instead of strictly copying retrieved scalar values. Another significant source of errors is incorrect Cypher query generation. In some complex multi-hop advisory queries, the generated Cypher statements failed to fully capture the semantic constraints of the user query, leading to incomplete or irrelevant graph retrieval results.

The system also exhibited limitations when handling ambiguous or underspecified user queries. For example, questions such as "Which major is better?" require additional contextual information regarding student preferences, academic strengths, or career goals. In such cases, the absence of sufficient contextual constraints may reduce recommendation accuracy. In several failure cases, missing or incomplete entities in the Knowledge Graph also affected system performance. Since the current graph is constructed from a limited set of university admission documents, certain specialized relationships or updated admission attributes may not yet be represented.

Finally, a small number of errors were associated with multi-turn conversational reasoning. Although the Short-Term Memory mechanism preserves recent interactions, long conversational chains occasionally caused context dilution, resulting in incorrect follow-up interpretation.

5. CONCLUSIONS

This paper presented a GraphRAG-based conversational question answering system designed to support university admission counseling at Electric Power University. By integrating a structured Knowledge Graph with large language model reasoning, the proposed system addresses the limitations of purely generative approaches, particularly in terms of consistency, explainability, and domain grounding.

The experimental evaluation on a curated dataset of 100 admission-related queries demonstrated that the system achieves a 70% overall success rate, with acceptable response latency and strong performance on definitional, procedural, and multi-hop informational queries. These results confirm that combining symbolic graph retrieval with neural language understanding provides a practical balance between flexibility and reliability in real-world academic advising scenarios.

One of the key contributions of this work is the adoption of a single-agent architecture that unifies intent detection, graph querying, and answer synthesis within a coherent reasoning pipeline. This design simplifies system orchestration, reduces engineering overhead, and enables efficient scaling compared to multi-agent or rule-heavy systems. In addition, the integration of short-term and long-term conversation memory allows the system to handle follow-up questions and maintain contextual coherence across multiple dialogue turns.

Despite these strengths, the evaluation also revealed notable limitations. The most significant source of error arises from numerical hallucination, particularly when responding to queries requiring exact scalar values such as tuition fees, admission quotas, or year-specific scores. This limitation highlights the need for stricter constraints

on graph query generation and more deterministic handling of numerical attributes during response synthesis.

Overall, the proposed system demonstrates that GraphRAG is a viable and effective paradigm for domain-specific conversational AI in higher education settings. By grounding language model outputs in a structured knowledge graph, the system improves response consistency, transparency, and user trust compared to traditional manual counseling processes.

ACKOWLEDGMENT

The author would like to thank the referees for giving precious comments and suggestions. This work was done under the full support of the Science and Technology Foundation of Electric Power University under grant number ĐTKHCN.09/2025.

REFERENCES

[1]. H. Xu, W. Gan, Z. Qi, J. Wu, P. S. Yu, "Large Language Models for Education: A Survey," *arXiv*: arXiv:2405.13001, 2024. doi: 10.48550/arXiv.2405.13001.

[2]. N. E. H. Ben Chaabene, H. Hammami, "Neuro-symbolic synergy in education: a survey of LLM-knowledge graph integration for explainable reasoning and emotion-aware student support," *Smart Learn. Environ.*, 13, 1, p. 6, 2026. doi: 10.1186/s40561-025-00423-z.

[3]. H. Abu-Rasheed, M. H. Abdulsalam, C. Weber, M. Fathi, "Supporting Student Decisions on Learning Recommendations: An LLM-Based Chatbot with Knowledge Graph Contextualization for Conversational Explainability and Mentoring," *arXiv*: arXiv:2401.08517, 2024. doi: 10.48550/arXiv.2401.08517.

[4]. Y. Leng, "A Major Recommendation System for University Admissions Based on Knowledge Graphs and Large Language Models," *Appl. Comput. Eng.*, 184, 154-164, 2025. doi: 10.54254/2755-2721/2025.LD27290.

[5]. S. Wang, et al., "Large Language Models for Education: A Survey and Outlook," *arXiv*: arXiv:2403.18105, 2024. doi: 10.48550/arXiv.2403.18105.

[6]. C. Ma, Y. Chen, T. Wu, A. Khan, H. Wang, "Large Language Models Meet Knowledge Graphs for Question Answering: Synthesis and Opportunities," *arXiv*: arXiv:2505.20099, 2025. doi: 10.48550/arXiv.2505.20099.

[7]. Z. Li, Z. Wang, W. Wang, K. Hung, H. Xie, F. L. Wang, "Retrieval-augmented generation for educational application: A systematic survey," *Comput. Educ. Artif. Intell.*, 8, p. 100417, 2025. doi: 10.1016/j.caeai.2025.100417.

[8]. G. Wang, Z. Zhan, S. Qin, "Synergizing Knowledge Graphs and LLMs: An Intelligent Tutoring Model for Self-Directed Learning," *Educ. Sci.*, 15, 9, p. 1102, 2025. doi: 10.3390/educsci15091102.

[9]. H. Abu-Rasheed, C. Weber, M. Fathi, "Knowledge Graphs as Context Sources for LLM-Based Explanations of Learning Recommendations," in *2024 IEEE Global Engineering Education Conference (EDUCON)*, 1-5, 2024. doi: 10.1109/EDUCON60312.2024.10578654.

[10]. S. Abdelhamid, J. Bangura, S. Shah, "Pixel International Conferences," *New Perspect. Sci. Educ.*, 2025, Accessed: Jan. 30, 2026. [Online]. Available: https://conference.pixel-online.net/library_scheda.php?id_abs=7058

[11]. L. Nguye, T. Quan, "URAG: Implementing a Unified Hybrid RAG for Precise Answers in University Admission Chatbots - A Case Study at HCMUT," *arXiv*: arXiv:2501.16276, 2025. doi: 10.48550/arXiv.2501.16276.