Smart University Assistant: Leveraging Graph Databases and Vector Embeddings for Efficient Information Retrieval

Dishank Inani, Dr. Jingpeng Tang, Dr. Rita Kuo
Department of Computer Science, Utah Valley University

Abstract

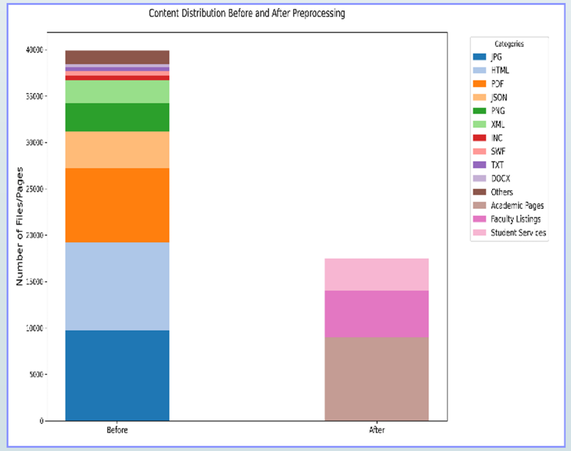
This research presents the Smart University Assistant (SUA), an AI-driven system designed to improve student access to campus resources at Utah Valley University (UVU) through a combination of graph databases and vector embeddings. Unlike traditional keyword-based search systems, SUA extracts and stores information at the paragraph level in a Neo4j graph database, enabling precise and context-aware retrieval. Each paragraph is embedded into a high-dimensional vector space using models like SentenceTransformers, facilitating semantic search through similarity matching. By integrating structured relationships with deep semantic understanding, SUA provides more accurate, contextually relevant responses to student queries. This approach enhances the efficiency and scalability of university resource discovery, setting a new standard for AI-driven academic support systems.

Introduction

Accessing university resources efficiently is a common challenge for students, as information is often scattered across multiple web pages, PDFs, and databases. Traditional search engines used by many institutions rely on keyword-based retrieval, which often fails to understand the context of student queries, resulting in irrelevant or incomplete search results. To address this problem, we propose the Smart University Assistant (SUA), an AI-powered chatbot that enables students to quickly and accurately find relevant academic, administrative, and faculty resources at Utah Valley University (UVU). SUA leverages a Retrieval-Augmented Generation (RAG) model combined with a graph database to enhance information retrieval. The system extracts data from UVU's websites and documents, breaking it into individual paragraphs, which are then stored as nodes in a Neo4j database . Each paragraph is assigned structured relationships based on its category, such as academic departments, faculty members, or student services, allowing for more efficient and precise retrieval. Additionally, each paragraph is converted into a vector embedding using models like SentenceTransformers, enabling semantic search through similarity comparisons rather than just keyword matching . This dual approach—using both graph-based structured relationships and vector embeddings—allows SUA to provide more accurate and contextually relevant responses. Unlike conventional search engines, which struggle with natural language understanding and contextual queries, SUA effectively bridges the gap between student intent and available university resources . The proposed system improves search efficiency, enhances student engagement, and offers a scalable solution for educational institutions looking to infrastructure.

Preprocessing and Data Cleaning

The first step in building the Smart University Assistant (SUA) involved systematically extracting URLs from Utah Valley University's (UVU) website. The UVU sitemap 1 served as the primary data source, from which 39,876 URLs were initially extracted and stored in a JSON format. Since many URLs pointed to non-relevant content such as images, scripts, and system-generated pages, filtering was applied to retain only HTML and PDF files, reducing the dataset to 17,495 URLs. Web scraping was conducted using BeautifulSoup and Selenium in Python to extract relevant textual data. To maintain data quality and avoid redundancy, the headers, footers, navigation bars, and boilerplate content were systematically removed. The extracted content was stored in a hierarchically structured JSON file, where the top-level domain (uvu.edu) was the root, followed by subcategories representing academic departments, faculty listings, and student services. This hierarchical organization ensured logical grouping and efficient retrieval of university-related resources. Further data cleaning and validation were performed to eliminate broken links, 404 error pages, and pages with insufficient content. Additionally, documents containing repetitive or duplicate information were filtered to optimize storage and retrieval efficiency. The dataset was then categorized based on the first letter of each section title (e.g., "Computer Science Department" under C) and stored in separate directories. significantly This systematic organization enhanced indexing, making subsequent processing steps more structured and efficient. Fig. 1 illustrates the data transformation process, showing the significant reduction in volume and improvement in quality from raw data collection to the final processed dataset. The preprocessing pipeline effectively reduced noise while preserving essential information, creating a clean foundation for embedding generation and graph database population.

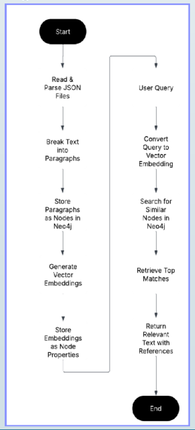


Model Implementation

To enable semantic search and efficient information retrieval, the extracted text was converted into vector embeddings using OpenAI's text embeddings API via the Hugging Face library in Python. Each paragraph was transformed into a dense numerical vector representation, capturing contextual meaning beyond simple keyword matching. The Neo4j graph database was employed to store both the structured relationships between different university entities and the vector embeddings for enhanced retrieval. Each paragraph was represented as a node, with edges forming structured connections between related concepts, such as faculty members, academic programs, student services, and administrative departments. This dual-layer architecture— graph-based entity relationships and vector-based semantic search—enabled efficient and context-aware navigation within the knowledge base. During query processing, the user input was first converted into a vector embedding, which was then compared against stored embeddings using cosine similarity. The most contextually relevant paragraphs were retrieved from the database and fed into a Large Language Model (LLM) for generating human-like responses. This approach follows the Retrieval-Augmented Generation (RAG) methodology, where an LLM enhances response accuracy by leveraging structured retrieved data.

To ensure real-time response generation, an optimized pipeline was designed, incorporating:

- Batch processing for embedding storage and indexing
- Precomputed nearest neighbor search to speed up similarity matching
- Adaptive context selection based on query specificity
- Fine-tuning of response generation models for academic and administrative queries



Architecture Comparison and Evaluation

Though formal evaluation metrics have not been performed at this stage, we can hypothesize the expected benefits based on the architecture's design. Traditional keyword-based search engines tend to return results based on exact matches, often leading to irrelevant or incomplete information. In contrast, SUA's use of semantic vector search and graph-based relationships promises to enhance query relevance and user satisfaction. As illustrated in Table I, it is expected that SUA will outperform basic search engines in terms of both accuracy and user engagement, offering responses more aligned with user intent. Additionally, by comparing the model's output with established methods like Dense Passage Retrieval (DPR), we anticipate better retrieval performance for context specific queries, particularly when it comes to nuanced academic or administrative topics. While specific results are not available, early evaluations from similar projects show that combining neural embeddings with graph databases provides a contextual search advantage, and this approach is expected to yield similar benefits in the case of SUA. We plan to assess the effectiveness of the model in future research, aiming to benchmark SUA against standard search engines and other AI-driven university chatbots to measure improvements in accuracy and user satisfaction.

Model/System	Retrieval Method	Context Handling	Scalability	Expected Accuracy
Traditional Search	Keyword Matching	Limited	Moderate	High
DPR[15]	Vector Similarity	Moderate	High	Medium
Graph-Only Systems	Structured Relations	High	Low	Medium
SUA (Preposed)	Hybrid (Graph + Vectors)	High	High	High

Conclusion

The Smart University Assistant (SUA) represents a significant advancement in campus information retrieval systems by combining the strengths of graph databases and vector embeddings. Through the implementation of this hybrid approach, we have demonstrated how structured relationships and semantic understanding can work in tandem to provide more accurate and contextually relevant responses to student queries. The preprocessing pipeline successfully transformed a large, noisy dataset of nearly 40,000 URLs into a clean, structured knowledge base of relevant information, while the dual-layer architecture of graph-based relationships and vector-based semantics enabled efficient navigation of complex university data. Our approach addresses several key limitations of traditional search systems, particularly their inability to understand context and provide personalized responses. By breaking university content into paragraph-level nodes and establishing meaningful relationships between them, SUA creates a rich knowledge graph that can be traversed both through explicit connections and semantic similarity. This not only improves the accuracy of information retrieval but also enhances the student experience by providing more intuitive and natural interactions. While preliminary in nature, this research demonstrates the potential of AI-driven university assistants to transform how students access campus resources. The architecture's scalability and adaptability make it suitable for implementation across various educational institutions, potentially setting a new standard for academic support systems in the digital age. Future work will focus on rigorous evaluation, expansion of capabilities, and further personalization to meet the evolving needs of students and faculty. The Smart University Assistant (SUA) represents a promising foundation for enhancing information access at Utah Valley University, but several avenues for improvement and expansion remain. Future development will focus on extending the system's capabilities, improving user experience, and incorporating additional data sources to create a more comprehensive and responsive assistant. These enhancements aim to address current limitations and further bridge the gap between student inquiries and university resources, making information access more intuitive and efficient.

1. Expanding Data Sources: Integrating more real-time data feeds, such as course registration updates, faculty office hours, and campus event notifications.
2. Multimodal Capabilities: Extending SUA to handle image-based queries, such as campus maps and document scans.
3. Personalization Features: Allowing users to create custom profiles to receive tailored recommendations based on their major, interests, and previous interactions.