VNUHCM Journal of Economics, Business and Law 2026, 10(2): 6508-6514

Research article

Open Access Full Text Article

Building an Information Retrieval System for University Documents Based on Generative AI Technologies

Hoang-Lam Vo [1,2], Quang-Phuc Nguyen [1,2], Duy Thanh Tran [1,2,*]

Use your smartphone to scan this QR code and download this article

ABSTRACT

This study will entertain the design and implementation of a generative AI-based information retrieval system to enhance university documentation access. On the contrary, other search engines that look for keywords and provide irrelevant or lacking results, the system hence uses semantic search and natural language interaction with LangChain, Facebook AI Similarity Search (FAISS), and the OpenAI API integrated. Document embedding is used to place documents in high-dimensional vector spaces, while Retrieval-Augmented Generation (RAG) helps provide responses that are context-aware, accurate, and in natural language. The architecture comprises Django in the back-end and Angular for the front-end, designed to scale and remain modular to handle the processing of complex, poorly indexed, and unstructured academic documents with ease. University documents are processed through OCR-enabled pipelines, chunked, and stored in FAISS for fast similarity-based matching. LangChain then connects the retrieval results to the generative model, guaranteeing that the response is well-grounded on actual documents, thereby circumventing hallucinations and providing human-sounding fluency. While the evaluation provided ample evidence about the efficacy and accuracy of the approach, an overall accuracy of 89% was recorded, while, for specific questions, it was as high as 92%. The average time taken to retrieve was 1.33 seconds, and generation took around 3.3 seconds, all of which had user ratings above average for categories like relevance, ease of use, and satisfaction. It is, therefore, clear that vector-based semantic retrieval coupled with generative AI offers a promising alternative to circumvent the shortcomings of traditional search methods. Emerging problems, however, include embedding and inference costs, dependency on third-party APIs, and difficulties in text extraction from scanned or locally formatted, very untidy PDFs. Future work includes performance improvements such as caching and incremental indexing, better OCR, and better multilingual and adaptive learning support. With these characteristics, the system can pursue aggressively with the implementation of its functions in the academic environment, thereby bridging the gap at the two ends of keyword search and actual context-based information access.

Key words: Generative AI, Information Retrieval, FAISS, LangChain, Retrieval-Augmented Generation, Semantic Search, University Documentation

[1]*University of Economics and Law, Ho Chi Minh City, Vietnam*

[2]*Vietnam National University, Ho Chi Minh City, Vietnam*

Correspondence

Duy Thanh Tran , University of Economics and Law, Ho Chi Minh City, Vietnam

Vietnam National University, Ho Chi Minh City, Vietnam

Email: thanhtd@uel.edu.vn.

History

- Received: 21/03/2025
- Revised: 10/10/2025
- Accepted: 09/04/2026
- Published Online: 05/05/2026

DOI:https://doi.org/10.32508/vnuhcmjebl.v10i2.1597

Check for updates

VNUHCM PRESS

INTRODUCTION

Generative AI and NLP are the way to go since they will now improve accessibility and efficiency in document retrieval through semantic search. Traditional keyword-based search systems have proven futile in achieving accurate and contextually relevant search results in the vast documentation generated by universities on policies, regulatory regime and institutions on research. Typical examples of document retrieval systems with applications outside education include health care and legal research, where search precision and context become extremely important in the former. For instance, AI-assisted legal research greatly enhances the efficiency of document retrieval by analyzing vast legal banks and yielding relevant cases and precedents[1] . Similarly, AI-driven decision support systems assist doctors by providing rapid access to clinical guidelines, research papers, and patient records, improving diagnostic accuracy and treatment planning[2] . In education, for example, students and faculty from universities now have to cope with and find their way around relevant documents with very poor indexing or vastly inadequate keywording. A generative AI-based solution can be put in place, retrieving images on responses which may be precise and in natural query language using RAG[1] . This study develops an AI-based retrieval system integrating FAISS for vector search and LangChain for structured query processing. The technique seeks to embed documents inside high-dimensional vector spaces thus improving the accuracy of search, even when the query is complex, and dealing with performance optimization, depen-

Cite this article : H V, Q N, D T T. **Building an Information Retrieval System for University Documents Based on Generative AI Technologies**.VNUHCM J. Econ. Bus. Law 2026; 10(2):6508-6514.

dency management, and secure authentication. The study is meant to underpin access to university documentation with plans for enhanced performance and scaling capabilities in multilingualism and adaptation learning mechanisms.

RELATIVE WORKS

Historically, keyword searching would yield syntactically correct yet semantically irrelevant results, especially in unstructured academic repositories. Rarely being perfect, in recent years, LLMs have been integrated into workflows, yet problems such as hallucinations and computationally expensive inferences remain[3] . Vector-based retrieval, especially FAISS, is proving helpful in trying to resolve this semantic mismatch between e-commerce and academics[4–7] , with some effort paid to addressing scalability issues. LangChain further links LLMs to vector databases for Retrieval-Augmented Generation (RAG), so that the LLM can base answers on retrieved source material[8 9] , but has observed limitations related to memory and query reformulation.

Prior academic approaches rely on metadata indexing or rule- or thesaurus-based expansions. Such methods work well for standardi ed resources but offer no hope in addressing voluminous and diverse internal university documents. Similarly, search engines like Google Scholar fare well with published works but remain a novice when dealing with internal institutional materials. At the same time, standalone models such as ChatGPT would generate fluent but ungrounded answers.

Briefly, this work sets itself apart by combining FAISS-based semantic search with LangChain-orchestrated GPT responses, a tailored preprocessing for complex document formats, and a modularized Django backend. The hybrid design trades off precision in retrieval against fluency in generation and, in so doing, scales well to internal academic documentation.

Table 1 provides a feature-level comparison among the proposed RAG system, Google Scholar, and standalone ChatGPT. Google Scholar works wonderfully in retrieving published works in the domain of scholarly literature, but it stands weak when it comes to keyword-based searches of semantically related internal university documents. Standalone ChatGPT showed proficient natural language generation abilities but is not domain-specifically grounded and prone to hallucinations. On the other hand, the proposed system stitches FAISS-based semantic retrieval with LangChain-orchestrated GPT generation to provide contextually correct answers grounded in the institution's document corpus. Thus, it balances retrieval precision, generative fluency, and adaptability on the corpus.

Table 1: ComparativeAnalysis with Google Scholar and ChatGPT. ***Source: Authors***

Feature / Capability	Proposed System (RAG)	Google Scholar	Chat-GPT (Stan-dalone)
Covers Internal University Docs	Yes	No	No
Semantic Search (Vector-based)	Yes (FAISS)	Limited	No
Retrieval-Augmented Generation	Yes (LangChain + GPT)	No	No
Source Citation in Answers	Yes	Yes (docu-ment links)	No (unless prompted)
Up-to-date and Cus-tomizable Corpus	Yes	Limited	No
Risk of Hal-lucination	Low (grounded in retrieved docs)	Low (returns raw docu-ments)	High without re-trieval

THEORETICAL BACKGROUND:

Modern IR has shifted from keyword search methods to meaning-based retrieval via NLP and generative AI. However, keyword-based systems are simple but often produce syntax-wise correct results yet semantically irrelevant. This led to another paradigm where queries and documents are embedded into high-dimensional vector spaces, enabling similarity to be calculated semantically, rather than by exact term matching.

Vector retrieval systems such as Facebook AI Similarity Search (FAISS) enable rapid nearest-neighbours searches over these dense embeddings, further refining the focus on ambiguous or complex queries. The generative AI models, mainly the GPT series by OpenAI, lend contextual fluency to the responses by generating them in natural language. Under the Retrieval-Augmented Generation (RAG) framework, LLMs use the retrieved context for generating more grounded responses and hence, fewer hallucinations.

LangChain supports this process by tying LLMs to vector databases, providing memory, prompt templates, and orchestrating retrieval and generation. Therefore, FAISS, GPT, and LangChain lay down the pathway for building contextual, semantic, and user-friendly retrieval systems. The focus of the present study is on employing these technologies to deliver a scalable intelligent platform for accessing university documents.

PROPOSE FRAMEWORK

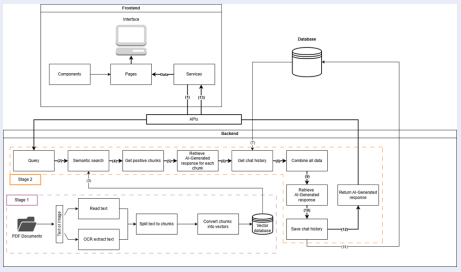

Figure 1: Architecture of an AI-Powered Information Retrieval System - *Source: Authors*

Figure 1 is an architectural diagram showing the Generative-AI-based Information Retrieval System. The system has three layers: front-end, back-end, and database. The frontend acts as the user interface, allowing user authentication, query submission, and visualization of fetched results. The frontend and backend communicate with each other through secure RESTful APIs. The backend constitutes the main computing layer, which performs query processing, retrieval, and generation of responses with the integration of LangChain and OpenAI's GPT API. The Secure Token Service provides authentication, authorization, and session management for users, while the user interactions history is maintained to provide context continuity. At the database level, a FAISS vector store is integrated for semantic search with a relational database to manage structured metadata, user information, and session logs. This layered architecture provides a highly scalable and context-aware retrieval framework as depicted in Figure 1.

Initialization, shown in Figure 2, details the institutional-type documents prepared for retrieval. This involves preprocessing PDFs uploaded with OCR and cleaning steps for textual readability. Then, documents are chunked into semantically coherent chunks so that downstream retrieval is fine-grained. Each chunk is converted into a giant-dimensional embedding using models such as BERT or GPT,

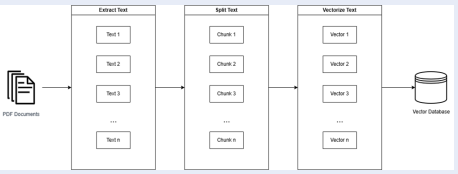

Figure 2: Vector Database Initialization Flow - *Source: Authors*

which embed relations on a semantic level and not just on mere keywords. Subsequently, the embeddings get indexed and stored in FAISS since it allows fast nearest-neighbor searching across large repositories. Figure 2 depicts the framework for how documents in their raw, unstructured state get converted into a searchable vectorised corpus.

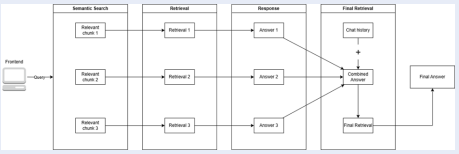

Figure 3: Retrieval System and Response Flow - *Source: Authors*

Figure 3 depicts the retrieval and response pipeline, showing how queries are taken, processed, and answered. The user query gets embedded first in the same higher-dimensional space as the document vectors. FAISS identifies and retrieves the most pertinent chunks based on semantic similarity. These retrieved chunks are given to LangChain, which does the prompt engineering and presents context to OpenAI's GPT for grounded response generation. The results are well-structured, natural language answers firmly based on institutional documents, thus placing less risk on hallucinations. The retrieval and response flow shown in Figure 3 confirm that the results are precise and contextually relevant, justifying their worth above the traditional keyword-based search.

IMPLEMENTATION AND DISCUSSION:

The UEL Generative AI Retrieval System is designed to offer the shortest administrative way to access institutional academic documents through a multi-stage pipeline, etc., utilizing state-of-the-art technologies. Semantic search and natural language question-answering are available through FAISS vector indexing and large language models of OpenAI. Each mod-

ule may intake documents, generate indexes, build retrievers, or produce intelligent responses-an architecture suited for highly diverse and complex academic formats.

Input:

- *folder_path:* Path to a folder containing .pdf files.

Output:

- *documents:* A list of LangChain Document objects containing parsed text and metadata.

Input:

- *documents:* Output from the PDF processing step.
- *folder_path:* Path to store the FAISS vector index.

Output:

- A saved FAISS index with embedded vectors of the document chunks.

Section 1: PDF Processing

```
procedure LoadPDFs
input ← folder_path
documents ← []
for file in folder_path do
if file.endswith('.pdf') then
for page in pdf.pages do
text ← extract_text(page)
if text is empty then
text ← OCR(page)
end if
documents.append(Document(text, metadata))
end for
end if
end for
return documents
end procedure
```

Section 2: Vector Database Creation

```
procedure CreateVectorDatabase
input ← documents
chunks ← TextSplitter.split(documents)
vectorstore ← FAISS.from_documents(chunks, embeddings)
vectorstore.save_local(path)
return vectorstore
end procedure
```

In this stage, textual documents are split into chunks of manageable size and are then converted into dense vectors, with vectorization done with OpenAI embeddings. These vectors will be stored in FAISS indexes, a retrieval mechanism that provides high-performance similarity search during search.

Input:

- *VECTOR_DATABASE_FOLDER:* Path where the FAISS index is or will be stored.

Output:

- retriever: A configured retriever object for semantic search using FAISS and OpenAI embeddings.

Section 3: System Initialization

```
procedure InitializeRetriever
if vectorstore exists then
vectorstore ← FAISS.load_local(path)
else
vectorstore ← CreateVectorDatabase(LoadPDFs(folder))
end if
retriever ← vectorstore.as_retriever(k=3)
return retriever
end procedure
```

It loads an existing index into FAISS or creates one if missing. It prepares the retriever interface which allows user queries to be matched with semantically similar chunks of documents so that the retrieval engine is always ready and scalable.

Input:

- *query:* User's question.
- *chat_history:* List of prior messages (used for context).
- *retriever:* A FAISS-based document retriever.
- *rag_chain:* LangChain-based RAG pipeline combining retriever + GPT.

Output: JSON response

- *response:* Answer generated by OpenAI based on context.
- *sources:* List of document sources supporting the answer.

This last phase answers user queries using RAG. Queries are contextualized with prior discussion history; documents are retrieved from the FAISS index to provide context for grounded answers in GPT. The

```
Section 4: AI Answer Generation
1: procedure GenerateAnswer
2: input ← request.body["Message"]
3: chatId ← request.body["ChatId"]
4: if chat_history is empty then
5: messages ← DB_Message.find(chatId).limit(4)
6: for msg in messages do
7: chat_history.append(HumanMessage(msg.User))
8: chat_history.append(SystemMessage(msg.System))
9: end for
10: end if
11: chat_history ← EnforceTokenLimit(chat_history)
12: result ← rag_chain.invoke({input, chat_history})
13: answer ← result["answer"]
14: sources ← extract_sources(retriever.invoke(input))
15: DB_Message.save(chatId, input, answer)
16: chat_history.append(HumanMessage(input))
17: chat_history.append(SystemMessage(answer))
18: return JsonResponse({response=answer,
sources=sources})
19: end procedure
```

answers are recorded and personalized, continuing from the same session.

For a replication of the implementation discussed in the section and to be able to cross-check the results obtained using the UEL Generative AI Retrieval System, one is advised to go through the publicly available source code repositories. Appendix Reference [1] hosts the entire backend codebase that consists of modules for PDF ingestion, vector indexing, and AI response generation. The frontend interface for user interaction with the retrieval system in question may be found at Appendix Reference [2]. Each repository also contains detailed setup instructions, dependencies, and configuration files for local deployment and testing. Further information on datasets and application contexts can be found in the data availability statements in the Appendix.

According to Table 2, the system underwent an evaluation for 172 interactions, yielding an average retrieval time of 1.33 seconds and a system response generation time of 3.3 seconds (SD = 0.23). The accuracy was 89%, up to 92% with specific queries and 85% with general ones. Users gave highly positive feedback- the satisfaction rating was 4.5/5, ease of use was rated at 4.3/5, and relevance was given a rank of 4.6/5.

This dictates that the proposed RAG-based system provides an efficient and friendly user experience with its overall performance level, reliability, and accuracy. Among the prominent strong points are grounded responses and better search precision, though computationally intensive, somewhat limiting due to dataset scope, and reliant on APIs. Overall, the findings belie the system's future potential in academic information retrieval.

Table 2: Quantitative evaluation result. ***Source: Authors***

Evaluation Metric	Value	Description
Number of Evaluation Samples	172	Total number of system interactions used for evaluation.
Average Document Retrieval Time	1.33 seconds	Time taken by the system to retrieve relevant documents.
Average Response Generation Time	3.3 seconds	Average time for the system to generate an AI-based response.
Standard Deviation (Response Time)	0.23 seconds	Indicates consistency in system performance across different queries.
Overall Accuracy	89%	Percentage of correct responses compared to ground truth.
Accuracy for Specific Queries	92%	Accuracy in answering precise and well-defined questions.
Accuracy for General Queries	85%	Accuracy in answering general or ambiguous queries.
Average User Satisfaction Rating	4.5 / 5	Overall user satisfaction with the system's functionality.
Ease of Use Score	4.3 / 5	User rating of how easy the system is to use.
Relevance Score	4.6 / 5	User rating of how relevant the responses were to their queries.

CONCLUSIONS

Implementing a Generative AI-based Information Retrieval system showed high scores for good retrieval accuracy, speed, and user satisfaction. By pairing FAISS semantic vector search with LangChain under the RAG framework, the system generated relevant contextual answers superior to keyword-based ones. Initial evaluation results showed that university documents were being handled in a balanced approach regarding speed and precision.

Some challenges still exist, especially regarding complex PDF handling, computational resources, and reliance on external APIs. Future development shall, therefore, consider the improvement of preprocessing, caching, and incremental indexing, along with multilingual document support. With all these improvements, the system may be a scalable AI-enabled approach for academic information retrieval.

AUTHORS' CONTRIBUTIONS

Hoang-Lam Vo: Conceptuali ation, Methodology design, Literature review, Experimental setup, Model development, Backend implementation of the Generative AI framework, Data preprocessing, Formal analysis, and Writing – original draft preparation.

Quang-Phuc Nguyen: Software engineering, Frontend development and system integration of the Generative AI platform, Data visualization, Experimental validation, Performance evaluation, and Writing – review & editing.

Duy Thanh Tran: Project supervision, Research design, Methodology design, Literature review, Theoretical analysis, Critical revision of experimental methodology, Backend implementation of the Generative AI framework, Frontend development, Results interpretation, comparison result, and Final review & editing of the manuscript.

LIST OF ABBREVIATIONS:

- AI: Artificial Intelligence
- FAISS: Facebook AI Similarity Search
- RAG: Retrieval-Augmented Generation
- NLP: Natural Language Processing
- API: Application Programming Interface

COMPETING INTERESTS

The author declares that there are no financial or non-financial competing interests in relation to this study.

ACKNOWLEDGEMENT:

This research is funded by University of Economics and Law, Vietnam National University Ho Chi Minh City, Vietnam.

APPENDIX:

All source code and resources associated with this research are publicly available for academic and non-commercial use. For additional information or specific data requests, please contact the corresponding author.

[1] Hoang-Lam Vo, Quang-Phuc Nguyen, Duy Thanh Tran*. *Backend source code repository*. Available at: https://link.uel.edu.vn/be-source-code

[2] Hoang-Lam Vo, Quang-Phuc Nguyen, Duy Thanh Tran*. Frontend source code repository. Available at: https://link.uel.edu.vn/fe-source-code-gen-ai

[3] Hoang-Lam Vo, Quang-Phuc Nguyen, Duy Thanh Tran*. *University document dataset and related resources. Accessible via: https://link.uel.edu.vn/gen-ai-dataset*

[4] Hoang-Lam Vo, Quang-Phuc Nguyen, Duy Thanh Tran*. https://link.uel.edu.vn/result-comparison

REFERENCES

1. Biresaw SM, Saste AU. The Impacts of Artificial Intelligence on Research in the Legal Profession. International Journal of Law and Society. 2022;5(1):53. Available from: https://10.11648/j.ijls.20220501.17.
2. Sopruchi AD, Rashid A, undefined Ariowachukwu Divine Sopruchi. The Integration of AI-Driven Decision Support Systems in Healthcare: Enhancements, Challenges, and Future Directions. Idosr Journal of Computer and Applied Sciences. 2024;9(2):17–25. Available from: https://10.59298/jcas/2024/92.1725.
3. Corchado JM, López F S, Núñez V JM, et al. Generative Artificial Intelligence: Fundamentals. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal. 2023;12(1). Available from: https://10.14201/adcaij.31704.
4. Johnson J, et al.. FAISS: A library for efficient similarity search and clustering of dense vectors.; 2017.
5. Zhang Y, et al. Semantic Search in E-commerce: A Review. J Retailing Consum Serv. 2018;45:145–57.
6. Borgman CL. Scholarship in the Digital Age. The MIT Press eBooks. The MIT Press; 2007. Available from: https://10.7551/mitpress/7434.001.0001.
7. Bard A, et al. Building context-aware applications with LangChain. J Artif Intell Res. 2022;75:1–15.
8. Manning CD, Raghavan P, Schütze H. Introduction to Information Retrieval.; 2008. Available from: https://10.1017/CBO9780511809071.
9. Yu Y, Wang J, Zhang Y, Zhang L, Yang Y, Sakai T. EALM: Introducing Multidimensional Ethical Alignment in Conversational Information Retrieval. . 2023;Available from: https://10.1145/3624918.3625327.

VNUHCM Journal of Economics, Business and Law 2026,10(2): 6508-6514

Bài báo nghiên cứu

Open Access Full Text Article

Xây dựng Hệ thống truy xuất thông tin cho tài liệu Đại học dựa trên công nghệ Trí tuệ nhân tạo tạo sinh

Hoang-Lam Vo [1,2], Quang-Phuc Nguyen [1,2], Duy Thanh Tran [1,2,*]

Use your smartphone to scan this QR code and download this article

TÓM TẮT

Nghiên cứu này tập trung vào việc thiết kế và triển khai một hệ thống truy xuất thông tin dựa trên trí tuệ nhân tạo tạo sinh (Generative AI) nhằm nâng cao khả năng tiếp cận tài liệu của các Trường Đại Học. Khác với các công cụ tìm kiếm truyền thống vốn dựa vào từ khóa và thường cung cấp kết quả thiếu chính xác hoặc không liên quan, hệ thống được đề xuất sử dụng tìm kiếm ngữ nghĩa (semantic search) và tương tác ngôn ngữ tự nhiên, kết hợp giữa LangChain, Facebook AI Similarity Search (FAISS) và OpenAI API. Quá trình nhúng tài liệu (document embedding) được áp dụng để biểu diễn tài liệu trong không gian vector có số chiều lớn, trong khi mô hình Retrieval-Augmented Generation (RAG) giúp tạo ra các phản hồi có tính ngữ cảnh, chính xác và tự nhiên. Chúng tôi đề xuất kiến trúc hệ thống gồm Django ở tầng back-end và Angular ở tầng front-end, được thiết kế theo hướng mô-đun hóa và có khả năng mở rộng nhằm xử lý hiệu quả các tài liệu học thuật phức tạp có chỉ mục kém hoặc phi cấu trúc. Các tài liệu của trường đại học được xử lý qua quy trình OCR, chia nhỏ (chunking) và lưu trữ trong FAISS để so khớp dựa trên độ tương đồng một cách nhanh chóng. LangChain sau đó kết nối kết quả truy xuất với mô hình tạo sinh nhằm đảm bảo phản hồi được tạo ra dựa trên nguồn tài liệu thực tế, qua đó hạn chế hiện tượng ``ảo tưởng'' (hallucination) và nâng cao tính tự nhiên trong ngôn ngữ phản hồi. Kết quả đánh giá cho thấy phương pháp đề xuất đạt hiệu quả và độ chính xác cao, với độ chính xác tổng thể đạt 89%, và đối với một số loại truy vấn cụ thể có thể lên đến 92%. Thời gian trung bình để truy xuất tài liệu là 1,33 giây, trong khi quá trình sinh phản hồi mất khoảng 3,3 giây. Hệ thống cũng đạt điểm đánh giá cao của người dùng về các tiêu chí như mức độ liên quan, độ dễ sử dụng và mức độ hài lòng. Do đó, có thể khẳng định rằng việc kết hợp giữa truy xuất ngữ nghĩa dựa trên vector và trí tuệ nhân tạo tạo sinh là một hướng tiếp cận đầy hứa hẹn nhằm khắc phục các hạn chế của phương pháp tìm kiếm truyền thống. Tuy nhiên, hệ thống vẫn đối mặt với một số thách thức như chi phí tính toán cho quá trình nhúng và suy luận, chi phí thực thi các API, cũng như khó khăn trong việc trích xuất văn bản từ các tệp PDF được quét hoặc định dạng phức tạp. Trong tương lai, nghiên cứu hướng đến việc cải thiện hiệu năng thông qua cơ chế lưu đệm (caching), lập chỉ mục gia tăng (incremental indexing), nâng cao khả năng OCR, và hỗ trợ tốt hơn cho tài liệu đa ngôn ngữ cũng như học thích ứng. Với những đặc tính này, hệ thống có tiềm năng trở thành một giải pháp khả thi và tiên tiến cho việc truy xuất thông tin học thuật trong môi trường Đại Học, góp phần thu hẹp khoảng cách giữa tìm kiếm dựa trên từ khóa và truy xuất thông tin dựa trên ngữ cảnh thực tế.

Từ khoá: Trí tuệ Nhân tạo Tạo sinh, Truy xuất Thông tin, FAISS, LangChain, Retrieval-Augmented Generation, Tìm kiếm Ngữ nghĩa, Tài liệu Đại học

[1]*University of Economics and Law, Ho Chi Minh City, Vietnam*

[2]*Vietnam National University, Ho Chi Minh City, Vietnam*

Liên hệ

Duy Thanh Tran , University of Economics and Law, Ho Chi Minh City, Vietnam

Vietnam National University, Ho Chi Minh City, Vietnam

Email: thanhtd@uel.edu.vn.

Lịch sử

- Ngày nhận: 21/03/2025
- Ngày sửa đổi: 10/10/2025
- Ngày chấp nhận: 09/04/2025
- Ngày đăng: 05/05/2026

DOI : https://doi.org/10.32508/vnuhcmjebl.v10i2.1597

Check for updates

VNUHCM PRESS

Trích dẫn bài báo này: H V, Q N, D T T. **Xây dựng Hệ thống truy xuất thông tin cho tài liệu Đại học dựa trên công nghệ Trí tuệ nhân tạo tạo sinh**. VNUHCM J. Econ. Bus. Law 2026; 10(2):6508-6514.