Phát triển trợ lý ảo nâng cao hỗ trợ sinh viên tại Trường Đại học Công nghệ Thông tin, ĐHQG TPHCM

Trình Trọng Tín[1*], Đào Gia Hải[2], Đoàn Văn Anh Hiển[3]
Trường Đại học Công nghệ Thông tin – Đại học Quốc gia Thành phố Hồ Chí Minh
tintt@uit.edu.vn

Tóm tắt

Trong bối cảnh giáo dục đại học ngày nay, nhu cầu được hỗ trợ nhanh chóng và chính xác về thông tin học vụ, quy chế, và các dịch vụ sinh viên ngày càng tăng. Trước thực trạng này, nghiên cứu đề xuất và triển khai hệ thống trợ lý ảo UITWiki, ứng dụng mô hình Truy xuất tăng cường và Tạo sinh (Retrieval-Augmented Generation) nhằm tự động giải đáp các thắc mắc liên quan đến quy định và hoạt động tại Trường Đại học Công nghệ Thông tin. Dữ liệu được thu thập và chuẩn hóa từ các tài liệu chính thức của các phòng ban, tạo thành cơ sở tri thức phục vụ cho quá trình truy xuất thông tin. Nhóm nghiên cứu đồng thời phân tích các điểm hạn chế của mô hình Naive RAG và đề xuất cải tiến theo ba giai đoạn: tiền xử lý dữ liệu, mô hình hóa truy xuất ngữ nghĩa (semantic retrieval) và sinh câu trả lời. Hệ thống được đánh giá qua thực nghiệm sử dụng nhiều mô hình embedding như Google, Cohere và một mô hình mã nguồn mở dành riêng cho tiếng Việt. Kết quả cho thấy phiên bản Advanced RAG sau cải tiến đạt độ chính xác và mức độ hài lòng người dùng vượt trội so với mô hình cơ bản. Nghiên cứu mở ra tiềm năng ứng dụng rộng rãi trong việc hỗ trợ sinh viên tại các cơ sở giáo dục đại học.

Nhận 05/06/2025
Được duyệt 15/07/2025
Công bố 30/10/2025

Từ khóa
Trợ lý ảo; Chatbot tư vấn; RAG; LLM; Hỗ trợ sinh viên

® 2025 Journal of Science and Technology - NTTU

1. Đặt vấn đề

Trong môi trường giáo dục bậc đại học, sinh viên ngày càng có nhu cầu được hỗ trợ toàn diện và kịp thời ở nhiều khía cạnh như quản lý học tập, giải đáp thắc mắc và truy cập thông tin. Nhằm đáp ứng nhu cầu này, các trường đại học đã đầu tư đáng kể thời gian và nguồn lực vào việc xây dựng các website cung cấp thông tin quan trọng. Tuy vậy, với số lượng sinh viên lớn và quy mô đào tạo ngày càng mở rộng, các nguồn lực hỗ trợ truyền thống như phòng ban chức năng, đội ngũ tư vấn hoặc giảng viên thường rơi vào tình trạng quá tải, đặc biệt trong các giai đoạn cao điểm như nhập học, đăng ký học phần, thi cử hay tuyển sinh.

Trước bối cảnh này, trí tuệ nhân tạo (AI) nổi lên như một giải pháp tiềm năng nhằm cải thiện chất lượng dịch vụ và nâng cao trải nghiệm người dùng. Các ứng dụng AI, điển hình là chatbot, với khả năng hoạt động liên tục 24/7 và cung cấp câu trả lời nhanh chóng, chính xác, đã chứng minh được giá trị trong việc hỗ trợ nhiều lĩnh vực khác nhau. Chatbot không chỉ giúp giảm tải cho đội ngũ tư vấn của nhà trường mà còn có thể phục vụ hiệu quả cho một số lượng lớn sinh viên, từ đó nâng cao mức độ hài lòng và tối ưu hóa trải nghiệm cá nhân hóa [1, 2].

Mặc dù chatbot truyền thống mang lại nhiều lợi ích, chúng vẫn còn tồn tại một số hạn chế như khả năng tổ chức tri thức còn hạn chế, khó khăn trong việc xác định đúng bối cảnh và kết nối các nội dung liên quan, sự cứng nhắc và thiếu khả năng tạo ra cuộc hội thoại tự nhiên, và khả năng cập nhật thông tin tức thời [1, 3]. Mô hình ngôn ngữ lớn (LLM), một thành tựu vượt bậc của AI, cũng gặp phải những hạn chế như khó khăn trong việc cập nhật thông tin thời gian thực và hiện tượng "ảo giác" (hallucination) tạo ra nội dung không chính xác [4]. Để giải quyết những vấn đề còn tồn tại của LLM và cải thiện khả năng của chatbot, mô hình Tạo sinh và Truy xuất tăng cường (RAG) được xem là một giải pháp hiệu quả [5]. RAG kết hợp khả năng tìm kiếm thông tin từ cơ sở dữ liệu với mô hình sinh văn bản, giúp cải thiện khả năng tổ chức tri thức, duy trì ngữ cảnh và tạo ra các phản hồi tự nhiên, linh hoạt hơn.

Thông qua bài báo này, nhóm nghiên cứu trình bày quá trình thiết kế, triển khai và đánh giá hệ thống trợ lý ảo UITWiki tại Trường Đại học Công nghệ Thông tin – Đại học Quốc gia TP HCM, một mô hình chatbot được xây dựng nhằm hỗ trợ sinh viên giải quyết các vấn đề thường gặp trong môi trường đại học, từ việc tra cứu thông tin về quy chế, học phí đến giải đáp các câu hỏi phổ biến. Nghiên cứu tập trung vào việc nâng cao chất

https://doi.org/10.55401/dwecxz69

lượng mô hình thông qua cải tiến RAG cơ bản bằng các kỹ thuật tiên tiến, đồng thời tiến hành kiểm chứng và so sánh hiệu quả giữa hai phiên bản: Naive RAG và Advanced RAG. Đối tượng nghiên cứu trong bài báo là sinh viên tại UIT, với phạm vi triển khai tập trung vào việc xây dựng một hệ thống chatbot ứng dụng trong lĩnh vực giáo dục nhằm cung cấp thông tin một cách nhanh chóng, chính xác và đáp ứng nhu cầu thực tiễn của người học.

2. Phương pháp nghiên cứu

2.1. Xử lí và chuẩn hóa dữ liệu đầu vào

Trong giai đoạn đầu, nhóm nghiên cứu tập trung vào việc xây dựng tập dữ liệu nền tảng cho hệ thống. Dữ liệu được thu thập từ các nguồn chính thức của UIT, bao gồm các văn bản từ Phòng Đào tạo, Phòng Công tác Sinh viên, Ban Tuyển sinh và chương trình đào tạo ngành Thương mại điện tử qua các năm.

Sau khi thu thập, dữ liệu được xử lý theo hai bước: (1) trích xuất nội dung văn bản từ các tệp PDF bằng công cụ Llama Parse, và (2) phân đoạn nội dung phục vụ cho mô hình RAG. Ban đầu, nhóm sử dụng thuật toán tách văn bản theo cơ chế đệ quy (Recursive Character Text Splitter) của Langchain, tuy nhiên nhận thấy thuật toán có sự phụ thuộc cao đối với các tham số chunk_size và chunk_overlap, nhóm chuyển sang phân đoạn theo cấu trúc Markdown nhằm bảo toàn mạch văn bản và ngữ cảnh logic. Việc xử lý và chuẩn hóa dữ liệu ở bước này đóng vai trò nền tảng, quyết định chất lượng của toàn bộ quá trình truy xuất và sinh phản hồi.

2.2. Truy xuất và biểu diễn vector ngữ cảnh

Bước tiếp theo trong quy trình nghiên cứu là truy xuất ngữ cảnh phù hợp với truy vấn đầu vào. Nhóm triển khai và đánh giá các phương pháp mã hóa văn bản sử dụng kỹ thuật truy xuất văn bản dày đặc (Dense Passage Retrieval) [6]. Các đoạn văn bản sau khi được phân đoạn sẽ được mã hóa thành vector và lưu trữ trong cơ sở dữ liệu vector. Đối với mỗi truy vấn, hệ thống tạo vector tương ứng và thực hiện phép đo độ tương đồng ngữ nghĩa để truy xuất các đoạn văn phù hợp nhất.

Trong quá trình thực nghiệm, nhóm tiến hành so sánh hiệu năng giữa ba mô hình nhúng: embed-multilingual-v3.0 (Cohere), text-embedding-004 (Google) và vietnamese-bi-encoder – một mô hình huấn luyện dành riêng cho tiếng Việt [7]. Việc lựa chọn mô hình nhúng phù hợp là bước then chốt nhằm tối ưu hóa độ chính xác của truy xuất ngữ cảnh.

2.3. Tạo sinh và điều chỉnh phản hồi từ mô hình ngôn ngữ lớn

Sau khi truy xuất được các đoạn văn liên quan, bước tiếp theo trong quy trình nghiên cứu là sinh phản hồi sử dụng mô hình ngôn ngữ lớn. Trong nghiên cứu này, nhóm lựa chọn Gemini 1.5 Flash của Google để thực hiện sinh văn bản dựa trên ngữ cảnh đầu vào. Mô hình này có khả năng xử lý văn bản dài với độ nhớ ngữ cảnh gần như hoàn hảo (99,7 %) với văn bản 1 triệu tokens và 99, 2 % độ chính xác với văn bản 10 triệu tokens [8], giúp tạo câu trả lời chính xác từ tài liệu dài và phức tạp, phù hợp với yêu cầu của bài toán trong môi trường học thuật.

Để tăng tính hiệu quả và kiểm soát chất lượng phản hồi, nhóm áp dụng kỹ thuật thiết kế prompt có cấu trúc, cụ thể là Chain-of-Note (CON) [9]. Cấu trúc này gồm ba thành phần: truy vấn người dùng, nội dung đã truy xuất và các ghi chú hỗ trợ. Cách tổ chức thông tin này giúp mô hình ngôn ngữ hiểu ngữ cảnh tốt hơn, từ đó tạo phản hồi chính xác, ngắn gọn và sát với nhu cầu thực tiễn.

2.4. Tối ưu hóa quy trình truy xuất và chất lượng phản hồi

Giai đoạn cuối cùng của quy trình nghiên cứu tập trung vào việc cải tiến hiệu suất hệ thống nhằm nâng cao độ chính xác và trải nghiệm người dùng. Các kỹ thuật được áp dụng bao gồm:

- Tối ưu hóa xử lý PDF: PDF chứa hình ảnh hoặc cấu trúc phức tạp gây khó khăn cho RAG. Nhóm sử dụng công nghệ OCR để trích xuất văn bản, sau đó tiến hành tóm tắt nội dung bằng LLM trước khi đưa vào pipeline truy xuất.
- Gán metadata cho tài liệu: nhằm cải thiện khả năng hiểu ngữ cảnh và tăng hiệu quả truy xuất, nhóm thực hiện gán nhãn metadata cho từng đoạn dữ liệu [10].
- Cập nhật kho tri thức: nhóm triển khai cơ chế upsert/delete để đảm bảo kho dữ liệu luôn phản ánh thông tin mới nhất từ nhà trường, đáp ứng yêu cầu cập nhật theo thời gian thực.
- Truy xuất kết hợp: theo nghiên cứu [11], kỹ thuật này nâng cao độ chính xác. Nhóm ứng dụng BM25, một phương pháp truy xuất thưa, đánh giá mức độ liên quan dựa trên tần suất từ khóa và độ dài tài liệu. Rerankers sắp xếp lại tài liệu theo mức độ liên quan chính xác hơn. Nghiên cứu cho thấy chúng cải thiện đáng kể độ chính xác [12]. Nhóm sử dụng rerank-multilingual-v3.0 từ Cohere đã được sử dụng trong nghiên cứu [13] để nâng cao chất lượng truy xuất.
- Triển khi bộ nhớ ngữ nghĩa: để xử lý các truy vấn tương đồng, nhóm triển khai Semantic Cache giúp giảm

thời gian phản hồi, tối ưu chi phí và tránh việc tính toán lại.

- Lịch sử hội thoại: duy trì ngữ cảnh hội thoại là thách thức lớn. Không có bộ nhớ, chatbot khó hiểu câu hỏi liên quan trước đó. Nhóm triển khai bộ nhớ hội thoại giúp chatbot ghi nhớ, phân tích tốt hơn, cá nhân hóa trải nghiệm và nâng cao chất lượng đối thoại.

3. Kết quả và thảo luận

Theo kết quả nghiên cứu của Barnett và các cộng sự [14], việc kiểm thử một mô hình RAG là khó khăn khi mà việc kiểm tra hiệu suất chỉ có thể thực hiện khi mô hình được vận hành. Hệ thống RAG phụ thuộc vào dữ liệu động trong cơ sở tri thức cho nên các câu hỏi và thông tin đầu vào có thể không được biết trước. Nhóm đã ứng dụng RAGAS – một framework phổ biến và được đánh giá là một trong những công cụ tự động hàng đầu trong việc đánh giá hiệu suất của các mô hình RAG bằng LLM [15]. Nhóm đã tiếp cận được phương pháp sử dụng LLM để tạo bộ kiểm thử gồm 100 cặp câu hỏi-câu trả lời tham chiếu tương ứng. Trước tiên, nhóm đính kèm tài liệu cần phân tích vào hệ thống LLM, chẳng hạn như ChatGPT. Sau đó, nhóm nhập một lệnh (prompt) chi tiết để yêu cầu LLM tạo ra số lượng câu hỏi kèm theo câu trả lời tương ứng. Các câu hỏi này phải tuân theo các quy tắc của nhóm, chẳng hạn như tập trung vào các nội dung quan trọng, phù hợp với phạm vi tài liệu, và được viết ngắn gọn, súc tích.

3.1. Các độ đo

Để có thể đánh giá toàn vẹn mô hình RAG, nhóm thực hiện đánh giá trên từng thành phần: truy xuất ngữ cảnh, sinh câu trả lời và hiệu suất tổng thể. Các chỉ số đánh giá được cung cấp bởi RAGAS bao gồm: Context Precision (tỷ lệ đoạn ngữ cảnh liên quan được truy xuất), Context Recall (khả năng thu thập bao quát thông tin quan trọng), và F1-score (trung bình điều hòa giữa Precision và Recall). Ngoài ra, Faithfulness đánh giá độ chính xác nội dung của mô hình ngôn ngữ lớn, giúp phát hiện thông tin sai lệch; Response Relevance đo mức độ phù hợp của câu trả lời với câu hỏi thông qua tính tương đồng ngữ nghĩa; Answer Correctness so sánh mức độ khớp giữa câu trả lời được tạo và câu trả lời tham chiếu.

3.1.1. Truy xuất ngữ cảnh

Độ chính xác ngữ cảnh (Context Precision) là một thước đo đánh giá tỷ lệ các đoạn ngữ cảnh liên quan câu hỏi nằm trong tổng số các ngữ cảnh được truy xuất Chỉ số này được tính bằng giá trị trung bình của precision@k ứng với từng đoạn trong ngữ cảnh. Precision@k phản ánh tỷ lệ số đoạn ngữ cảnh liên quan ở vị trí xếp hạng k so với tổng số đoạn ngữ cảnh tại vị trí đó.

(1)

$$\text{Precision@k} = \frac{\text{true positive@k}}{(\text{true positive@k} + \text{false positive@k})}$$

- Precision@k: Độ chính xác tại vị trí k
- true positive@k: Số lượng kết quả đúng tại vị trí k
- false positive@k: Số lượng kết quả sai tại vị trí k

(2)

$$\text{Context Precision@K} = \frac{\sum_{k=1}^{K}(\text{Precision@k} \times v_k)}{\text{Total number of relevant contexts in top K results}}$$

- v_k: Chỉ số ở dạng nhị phân (0,1) nhằm cho biết tài liệu có liên quan không. 1 nếu ngữ cảnh tại vị trí k liên quan và 0 nếu ngữ cảnh tại vị trí k không liên quan
- Total number of relevant contexts in top K results: Là tổng số lần các ngữ cảnh liên quan xuất hiện trong top K kết quả truy hồi

Độ phủ của ngữ cảnh (Context Recall) đo lường khả năng truy xuất được bao nhiêu tài liệu (hoặc thông tin) quan trọng. Chỉ số này tập trung vào việc tránh bỏ sót các kết quả quan trọng. Độ phủ cao đồng nghĩa với việc ít tài liệu liên quan bị bỏ qua. Chỉ số được tính bằng tỷ lệ giữa số lượng các ngữ cảnh được truy xuất đúng và liên quan đến câu hỏi so với tổng số ngữ cảnh tham chiếu (ngữ cảnh mẫu) trong câu trả lời chuẩn.

(3)

$$\text{Context Recall} = \frac{\text{Number of relevant contexts in top k results}}{\text{All contexts that contain ground truth}}$$

- Number of relevant contexts in top k results: Là số lượng kết quả trong top k truy hồi mà chứa ngữ cảnh có liên quan đến đáp án thực tế (ground truth)
- All contexts that contain ground truth: Là tổng số ngữ cảnh trong có chứa thông tin chính xác cần tìm

Nhằm cân bằng giữa 2 chỉ số trên, nhóm sử dụng độ đo F1-score là trung bình điều hòa giữa Precision và Recall

(4)

$$\text{F1 score} = 2 \times \frac{\text{Context Precision} \times \text{Context Recall}}{\text{Context Precision} + \text{Context Recall}}$$

3.1.2. Sinh câu trả lời

Mức độ trung thực (Faithfulness) là một chỉ số đo lường mức độ chính xác về mặt nội dung của câu trả lời được tạo ra so với ngữ cảnh cung cấp. Nó đánh giá liệu các thông tin được trình bày trong câu trả lời có được suy ra hợp lý từ ngữ cảnh đã truy xuất hay không.Chỉ số này có thể giúp phát hiện hiện tượng hallucination của mô hình LLM, kiểm tra xem mô hình LLM có tự ý đưa những thông tin khác vào trong câu trả lời không.

(5)

$$\text{Faithfulness score} = \frac{\text{Number of claims in the generated answer can be inferred from given contexts}}{\text{Total number of claims in the generated answer}}$$

- Number of claims in the generated answer can be inferred from given contexts: Số lượng khẳng định trong câu trả lời sinh ra có thể suy ra từ các ngữ cảnh đã được cung cấp trước đó
- Total number of claims in the generated answer: Toàn bộ các khẳng định trong câu trả lời sinh ra

Mức độ tương đồng trong câu trả lời (Response Relevance) là một chỉ số dùng để đánh giá mức độ phù hợp,liên quan của câu trả lời được tạo ra so với câu hỏi đầu vào. Mục tiêu của chỉ số này là xác định xem câu trả lời có liên quan và chính xác với yêu cầu hay không. Công thức này được tính bằng cách sử dụng câu trả lời đầu vào để sinh ra một tập hợp các câu hỏi giả lập (3 câu) bằng mô hình ngôn ngữ lớn. Sau đó, độ tương đồng ngữ nghĩa giữa từng câu hỏi giả lập và câu hỏi gốc được đo lường thông qua chỉ số cosine similarity. Cuối cùng, giá trị trung bình của các độ tương đồng cosine được tính để xác định mức độ liên quan của câu trả lời với câu hỏi ban đầu.

(6)

$$\text{Response relevancy} = \frac{1}{3}\sum_{i=1}^{3} \text{cosinesimilarity}\left(E_{g_i}\ ,\ E_0\right) = \frac{1}{3}\sum_{i=1}^{3} \frac{gE_{g_i} \times E_0}{\|E_{g_i}\|\ \|E_0\|}$$

- E_g: Là vector đặc trưng embedding của câu hỏi giả lập được sinh ra từ câu trả lời của mô hình
- E_o: Là vector đặc trưng embedding của câu hỏi gốc trong tập dữ liệu kiểm thử

3.1.3. Hiệu suất tổng thể

Độ chính xác về mặt thực tế (Answer Correctness) là một chỉ số được sử dụng để so sánh và đánh giá mức độ chính xác của câu trả lời được tạo ra so với câu trả lời tham chiếu. Chỉ số này xác định mức độ phù hợp giữa câu trả lời được tạo ra và câu trả lời tham chiếu.

(7)

$$\text{Answer Correctness (Recall Mode)} = \frac{\text{Number of relevant claims in generated answer}}{\text{Number of claims in referenced answer}}$$

- Number of relevant claims in generated answer: Số lượng khẳng định có liên quan trong câu trả lời sinh ra
- Number of claims in referenced answer: số lượng khẳng định có trong câu trả lời tham chiếu

3.2. Kết quả đạt được

Trong phần này, nhóm nghiên cứu tiến hành đánh giá định lượng hiệu quả hoạt động của hệ thống theo hai kiến trúc: Naive RAG và Advanced RAG; khi kết hợp với ba loại mô hình embedding khác của Cohere, Google và BKAI. Thí nghiệm được thực hiện bằng cách thay đổi số lượng đoạn ngữ cảnh được truy xuất với các giá trị K = 2, 3, 5, 7 và 10, từ đó quan sát sự thay đổi về hiệu suất ở từng khâu của hệ thống.

Kết quả trong Bảng 1 cho thấy mô hình embedding của Cohere hoạt động đặc biệt hiệu quả ngay cả với kiến trúc Naive RAG. Chất lượng truy xuất ngữ cảnh (F1-score) đạt đỉnh ở mức 0,86, đồng thời Faithfulness cũng đạt đến 0,90 tại K = 5. Độ chính xác của câu trả lời dao động ổn định quanh mức từ 0,77 đến 0,80.

Bảng 1. Kết quả đánh giá mô hình Cohere trong kiến trúc Naive RAG với các mức K khác nhau

	Truy xuất ngữ cảnh	Tạo câu trả lời bằng LLM	Hiệu suất tổng thể

Tham số K (số lượng tài liệu truy xuất)	Precision	Recall	F1-score	Faith.	Relev.	Correctn.
2	0,88	0,76	0,82	0,84	0,63	0,77
3	0,88	0,80	0,84	0,86	0,67	0,79
5	0,85	0,85	0,85	0,90	0,65	0,79
7	0,83	0,90	0,86	0,88	0,67	0,77
10	0,79	0,92	0,85	0,88	0,69	0,80

Trái ngược với kết quả trong Bảng 1,mô hình embedding của Google trong Bảng 2 lại cho hiệu suất rất thấp trong kiến trúc Naive RAG. Precision và Recall chỉ duy trì quanh mức 0,2 đến 0,4, kéo theo F1 thấp và Relevance cực kỳ yếu (luôn dưới 0,20). Mặc dù chỉ số Faithfulness có tăng dần theo K, nhưng độ chính xác câu trả lời vẫn chỉ đạt tối đa 0,43. Điều này phản ánh rõ sự không tương thích giữa Google embedding và kiến trúc Naive RAG.

Bảng 2. Kết quả đánh giá mô hình Google trong kiến trúc Naive RAG với các mức K khác nhau

Tham số K (số lượng tài liệu truy xuất)	Truy xuất ngữ cảnh			Tạo câu trả lời bằng LLM		Hiệu suất tổng thể
	Precision	Recall	F1-score	Faith.	Relev.	Correctn.
2	0,20	0,10	0,13	0,64	0,05	0,31
3	0,20	0,15	0,17	0,67	0,11	0,35
5	0,22	0,23	0,23	0,72	0,13	0,36
7	0,24	0,32	0,27	0,72	0,16	0,41
10	0,23	0,40	0,29	0,81	0,19	0,43

Trong Bảng 3, BKAI embedding trong kiến trúc Naive RAG đạt kết quả ở mức trung bình. Các chỉ số tăng dần theo K, đặc biệt là chỉ số Recall và chỉ số Faithfulness, tuy nhiên độ liên quan và độ chính xác câu trả lời vẫn còn hạn chế so với mô hình của Cohere trong Bảng 1. Đây là một cấu hình tương đối ổn định nhưng không nổi bật.

Bảng 3. Kết quả đánh giá mô hình BKAI trong kiến trúc Naive RAG với các mức K khác nhau

Tham số K (số lượng tài liệu truy xuất)	Truy xuất ngữ cảnh			Tạo câu trả lời bằng LLM		Hiệu suất tổng thể
	Precision	Recall	F1-score	Faith.	Relev.	Correctn0.
2	0,72	0,57	0,64	0,78	0,47	0,68
3	0,73	0,64	0,68	0,83	0,54	0,74
5	0,74	0,74	0,74	0,85	0,56	0,72
7	0,72	0,75	0,73	0,86	0,57	0,74

10	0,69	0,82	0,75	0,88	0,62	0,71

Mô hình embedding của Cohere tiếp tục khẳng định sự vượt trội trong kiến trúc Advanced RAG dựa theo kết quả của Bảng 4. Ở K = 7, hệ thống đạt đồng thời độ trung thực cao nhất đạt đến 0,91 và độ chính xác câu trả lời lớn nhất là 0,83 trong toàn bộ thí nghiệm. Điều này khẳng định tính phù hợp mạnh mẽ giữa kiến trúc Advanced và embedding của Cohere.

Bảng 4. Kết quả đánh giá mô hình Cohere trong kiến trúc Advanced RAG với các mức K khác nhau

Tham số K (số lượng tài liệu truy xuất)	Truy xuất ngữ cảnh			Tạo câu trả lời bằng LLM		Hiệu suất tổng thể
	Precison	Recall	F1-score	Faith.	Relev.	Correctn.
2	0,91	0,77	0,83	0,83	0,66	0,81
3	0,90	0,80	0,85	0,87	0,69	0,79
5	0,87	0,86	0,87	0,88	0,65	0,79
7	0,84	0,89	0,86	0,91	0,72	0,83
10	0,80	0,89	0,84	0,86	0,70	0,79

Mặc dù embedding của Google cho kết quả thấp trong Bảng 2, nhưng trong cấu hình của kiến trúc Advanced RAG, mô hình này đạt hiệu suất tương đối ổn định. Các chỉ số đánh giá trong Bảng 5 đều được cải thiện và vượt trội hơn so với Bảng 2, cho thấy kiến trúc Advanced RAG có khả năng nâng cao hiệu quả hơn với các mô hình embedding kém trong kiến trúc Naive RAG.

Bảng 5. Kết quả đánh giá mô hình Google trong kiến trúc Advanced RAG với các mức K khác nhau

Tham số K (số lượng tài liệu truy xuất)	Truy xuất ngữ cảnh			Tạo câu trả lời bằng LLM		Hiệu suất tổng thể
	Precison	Recall	F1-score	Faith.	Relev.	Correctn.
2	0,88	0,73	0,80	0,84	0,60	0,74
3	0,87	0,76	0,81	0,85	0,63	0,75
5	0,82	0,81	0,81	0,86	0,64	0,74
7	0,80	0,84	0,82	0,89	0,65	0,65
10	0,77	0,88	0,82	0,88	0,68	0,79

Advanced RAG khi kết hợp với BKAI embedding mang lại hiệu suất ổn định trên các chỉ số đánh giá trong Bảng 6, đặc biệt là trong truy xuất ngữ cảnh và mức độ trung thực của câu trả lời. Cấu hình này vẫn cho thấy khả năng tổng thể đáng tin cậy, phù hợp với các hệ thống yêu cầu tính cân bằng giữa chi phí và hiệu quả

Bảng 6. Kết quả đánh giá mô hình BKAI trong kiến trúc Advanced RAG với các mức K khác nhau

Tham số K (số lượng tài liệu truy xuất)	Truy xuất ngữ cảnh			Tạo câu trả lời bằng LLM		Hiệu suất tổng thể
	Precison	Recall	F1-score	Faith.	Relev.	Correctn.
2	0,90	0,73	0,81	0,83	0,62	0,75
3	0,90	0,79	0,84	0,84	0,69	0,76
5	0,87	0,84	0,85	0,87	0,69	0,75
7	0,85	0,85	0,85	0,88	0,69	0,77
10	0,81	0,89	0,85	0,90	0,68	0,77

3.3. Thảo luận

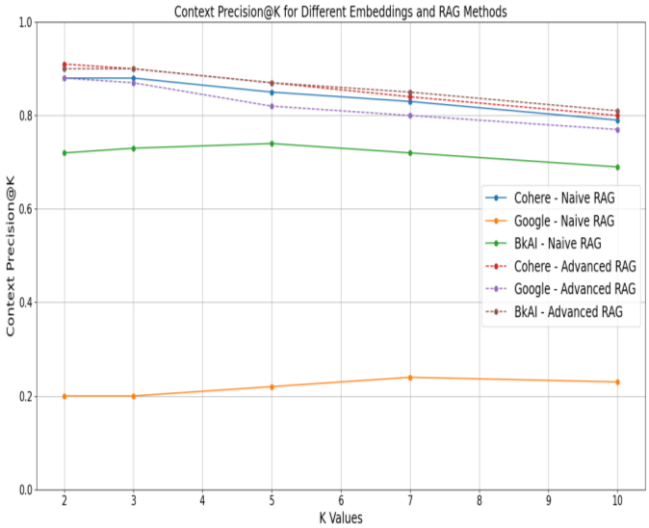
Dựa trên các bảng kết quả thực nghiệm (Bảng 1–6), mô hình Advanced RAG do nhóm cải tiến cho thấy hiệu năng vượt trội so với mô hình Naive RAG trong cả truy xuất và xử lý thông tin. Cụ thể, các cấu hình Advanced RAG (Bảng 4 đến Bảng 6) đều đạt điểm cao hơn các cấu hình Naive RAG tương ứng (Bảng 1 đến Bảng 3) trên hầu hết các chỉ số đánh giá, cho thấy vai trò thiết yếu của các thành phần mở rộng trong việc nâng cao chất lượng hệ thống.

Các Hình 1 đến Hình 3 trình bày ba chỉ số chính trong giai đoạn truy xuất ngữ cảnh. Khi giá trị K tăng, số lượng ngữ cảnh được truy xuất cũng tăng theo, dẫn đến Context Precision@K giảm (Hình 1) do xuất hiện ngữ cảnh nhiễu, trong khi Context Recall tăng (Hình 2) nhờ bao phủ tốt hơn các ngữ cảnh liên quan. Hình 3 cho thấy mô hình Advanced RAG duy trì chỉ số F1-score ổn định ở mức trên 0,80, phản ánh sự cân bằng giữa độ chính xác và khả năng bao phủ trong bước truy xuất.

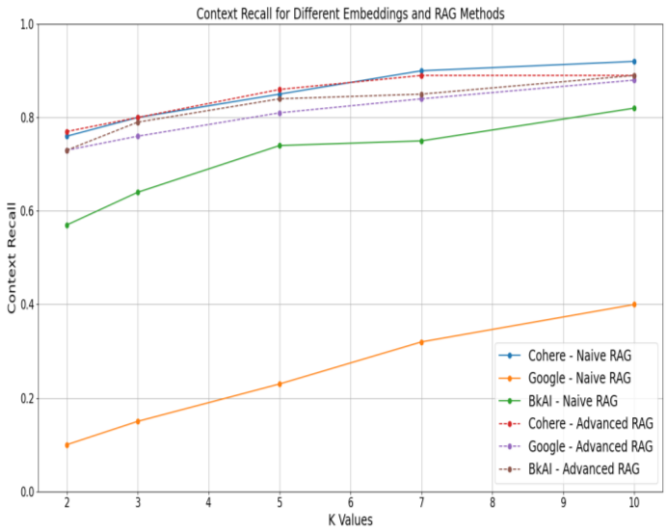
Trong khâu sinh câu trả lời, hai chỉ số Faithfulness và Answer Relevance (Hình 4 và Hình 5) đều được cải thiện rõ rệt khi sử dụng Advanced RAG. Hiệu quả tăng rõ nhất được ghi nhận với embedding của Google, vốn có hiệu năng thấp trong kiến trúc Naive RAG. Điều này cho thấy các thành phần cải tiến trong Advanced RAG đã góp phần nâng cao chất lượng ngữ cảnh đầu vào, giúp LLM tạo ra câu trả lời chính xác và phù hợp hơn. Với Cohere và BKAI, mô hình cũng đạt được hiệu suất ổn định và cao.

Xét về chất lượng embedding, Cohere thể hiện ưu thế nổi bật nhờ khả năng biểu diễn ngữ nghĩa đa ngôn ngữ, giúp duy trì độ chính xác cao và ổn định trong cả hai kiến trúc RAG, đặc biệt khi K tăng. Trong khi đó, Google embedding gặp nhiều hạn chế trong Naive RAG do chưa phù hợp với tiếng Việt, nhưng lại được cải thiện đáng kể trong Advanced RAG nhờ áp dụng cơ chế reranking. BKAI embedding duy trì kết quả ở mức trung bình giữa hai mô hình trên, nhưng vẫn thấp hơn Cohere ở hầu hết chỉ số.

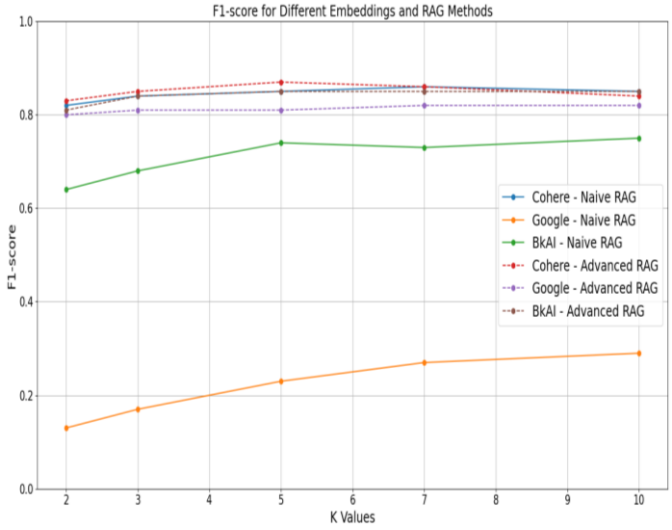
Từ các phân tích trên, nhóm quyết định lựa chọn cấu hình Advanced RAG kết hợp với embedding Cohere và tham số truy xuất K = 7 để triển khai hệ thống trong thực tế. Cấu hình này cho thấy sự cân bằng tốt giữa độ chính xác, khả năng bao phủ ngữ cảnh và chất lượng sinh câu trả lời, đồng thời đảm bảo hiệu suất ổn định trên cả ba giai đoạn: truy xuất ngữ cảnh, tạo câu trả lời và đánh giá tổng thể.



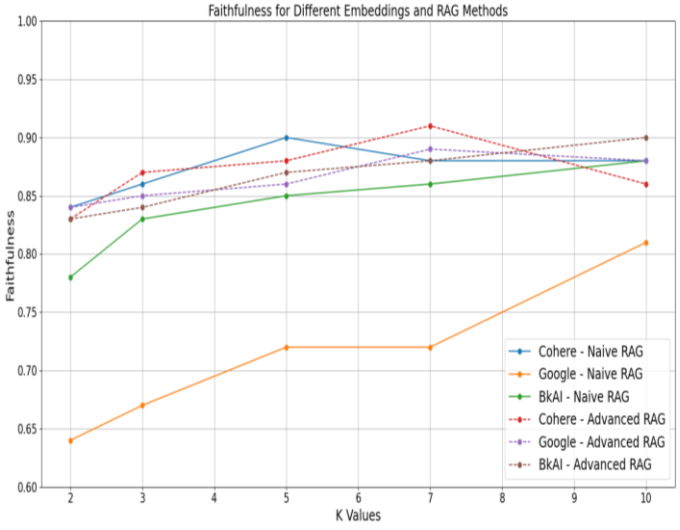
Hình 1. Biểu đồ đánh giá Context Precision@k



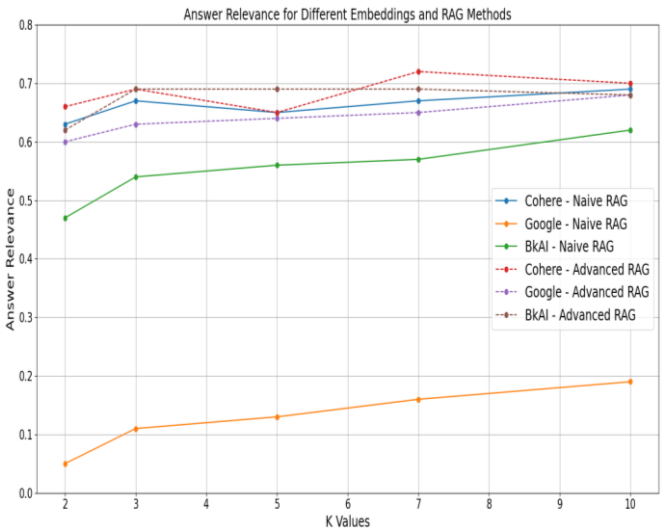
Hình 2. Biểu đồ đánh giá Context Recall



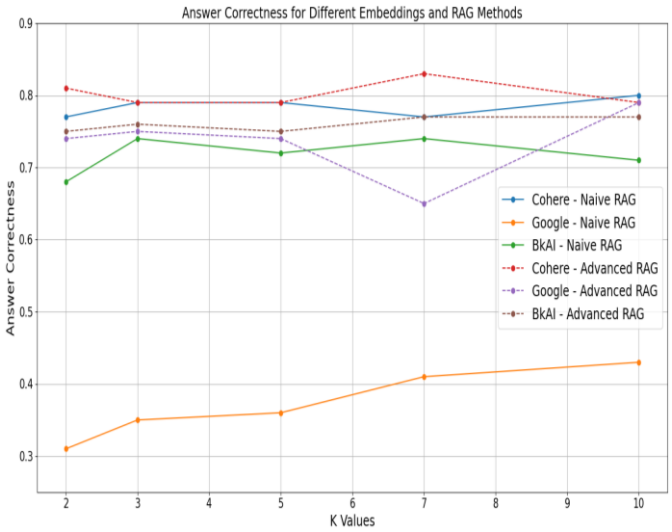
Hình 3. Biểu đồ đánh giá F1-score



Hình 4. Biểu đồ đánh giá Faithfullnes



Hình 5. Biểu đồ đánh giá Answer Relevance



Hình 6. Biểu đồ đánh giá Answer Correctness

4. Kết luận và đề xuất

Trong nghiên cứu này, nhóm đã xây dựng thành công hệ thống trợ lý ảo UITWiki phục vụ sinh viên tại Trường Đại học Công nghệ Thông tin – ĐHQG TP HCM, với khả năng hỗ trợ tự động trả lời các câu hỏi liên quan đến quy chế học vụ, học phí, tuyển sinh và các thông tin thường gặp trong môi trường đại học. Thông qua việc thiết kế và triển khai mô hình dựa trên kiến trúc RAG, nghiên cứu đã đạt được ba mục tiêu chính: (i) xây dựng hệ thống chatbot ứng dụng cho bài toán tư vấn sinh viên trong giáo dục đại học, (ii) cải tiến mô hình Naive RAG cơ bản với các thành phần mở rộng nhằm khắc phục các điểm hạn chế, (iii) kiểm chứng và so sánh hiệu quả định lượng giữa kiến trúc Naive RAG và Advanced RAG với các loại embedding khác nhau.

Các kết quả thực nghiệm cho thấy cấu hình Advanced RAG kết hợp với Cohere embedding và tham số K = 7 đạt hiệu suất cao nhất trong các cấu hình thử nghiệm, đảm bảo sự cân bằng giữa độ chính xác truy xuất, chất lượng sinh câu trả lời và khả năng phản hồi chính xác. Tuy nhiên, hệ thống vẫn còn một số hạn chế như khả năng mở rộng khi truy cập đồng thời lớn, và độ chính xác cần nâng cao ở các câu hỏi phức tạp. Do đó, trong các hướng phát triển tiếp theo, nhóm sẽ tập trung vào việc fine-tune mô hình LLM và embedding, cũng như tích hợp sâu với hệ thống dữ liệu nội bộ của trường, nhằm nâng cao độ chính xác và trải nghiệm người dùng trong các tình huống thực tế.

Lời cảm ơn

Nhóm tác giả xin trân trọng cảm ơn các thầy cô và các đơn vị chuyên môn tại Trường Đại học Công nghệ Thông tin – ĐHQG TPHCM đã tạo điều kiện thuận lợi, cung cấp môi trường nghiên cứu và những hỗ trợ cần thiết để triển khai đề tài liên quan đến phát triển trợ lý ảo phục vụ sinh viên.

Tài liệu tham khảo

1. Dibitonto, M., Leszczynska, K., Tazzi, F., & Medaglia, C. M. (2018). Chatbot in a Campus Environment: Design of LiSA, a Virtual Assistant to Help Students in Their University Life. Trong M. Kurosu (B.t.v), *Human-Computer Interaction. Interaction Technologies* (Vol 10903, tr 103–116). Springer International Publishing. https://doi.org/10.1007/978-3-319-91250-9_9

2. Hien, H. T., Cuong, P.-N., Nam, L. N. H., Nhung, H. L. T. K., & Thang, L. D. (2018). Intelligent Assistants in Higher-Education Environments: The FIT-EBot, a Chatbot for Administrative and Learning Support. *Proceedings of the Ninth International Symposium on Information and Communication Technology - SoICT 2018*, 69-76. https://doi.org/10.1145/3287921.3287937

3. Dinh, H., & Tran, T. K. (2023). EduChat: An AI-Based Chatbot for University-Related Information Using a Hybrid Approach. *Applied Sciences*, *13*(22), 12446. https://doi.org/10.3390/app132212446

4. Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., & Xie, X. (2024). A Survey on Evaluation of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, *15*(3), 1-45. https://doi.org/10.1145/3641289

5. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS 2020)*, 9459-9474. https://dl.acm.org/doi/pdf/10.5555/3495724.3496517

6. Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. (2020). Dense Passage Retrieval for Open-Domain Question Answering. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6769-6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

7. Nguyen, P.-V., Tran, M.-N., Nguyen, L., & Dinh, D. (2024). Advancing Vietnamese Information Retrieval with Learning Objective and Benchmark. *Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation*, 46-56. https://aclanthology.org/2024.paclic-1.5/

8. Team Google, Georgiev, P., Lei, V. I., Burnell, R., Bai, L., Gulati, A., Tanzer, G., Vincent, D., Pan, Z., Wang, S., Mariooryad, S., Ding, Y., Geng, X., Alcober, F., Frostig, R., Omernick, M., Walker, L., Paduraru, C., Sorokin, C., … Vinyals, O. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context (arXiv:2403.05530). arXiv. https://doi.org/10.48550/arXiv.2403.05530

9. Yu, W., Zhang, H., Pan, X., Cao, P., Ma, K., Li, J., Wang, H., & Yu, D. (2024). Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 14672-14685. https://doi.org/10.18653/v1/2024.emnlp-main.813

10. Khan, A. A., Hasan, M. T., Kemell, K. K., Rasku, J., & Abrahamsson, P. (2024). Developing Retrieval Augmented Generation (RAG) based LLM Systems from PDFs: An Experience Report (arXiv:2410.15944). arXiv. https://doi.org/10.48550/arXiv.2410.15944

11. Finardi, P., Avila, L., Castaldoni, R., Gengo, P., Larcher, C., Piau, M., Costa, P., & Caridá, V. (2024). The Chronicles of RAG: The Retriever, the Chunk and the Generator (arXiv:2401.07883). arXiv. http://arxiv.org/abs/2401.07883

12. Moreira, G. de S. P., Ak, R., Schifferer, B., Xu, M., Osmulski, R., & Oldridge, E. (2024). Enhancing Q&A Text Retrieval with Ranking Models: Benchmarking, fine-tuning and deploying Rerankers for RAG (arXiv:2409.07691). arXiv. https://doi.org/10.48550/arXiv.2409.07691
13. Krayko, N., Sidorov, I., Laputin, F., Galimzianova, D., & Konovalov, V. (2024). Efficient Answer Retrieval System (EARS): Combining Local DB Search and Web Search for Generative QA. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 1584-1594. https://doi.org/10.18653/v1/2024.emnlp-industry.116
14. Barnett, S., Kurniawan, S., Thudumu, S., Brannelly, Z., & Abdelrazek, M. (2024). Seven Failure Points When Engineering a Retrieval Augmented Generation System. *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI*, 194-199. https://doi.org/10.1145/3644815.3644945
15. Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2024). RAGAS: Automated Evaluation of Retrieval Augmented Generation. *Association for Computational Linguistics*, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 150-158.

Developing a virtual assistant to enhance student support at the University of Information Technology, VNU HCM

Trinh Trong Tin[1*], Dao Gia Hai[2], Doan Van Anh Hien[3]
University of Information Technology, VNU – HCM
tintt@uit.edu.vn

Abstract In the context of modern higher education, students increasingly demand immediate, accurate, and comprehensive support across various domains such as academic regulations, administrative procedures, and access to institutional information. Addressing these needs requires innovative solutions that can scale efficiently. This study introduces UITWiki, a virtual assistant system designed to automatically answer student queries related to the rules, policies, and services of departments at the University of Information Technology (UIT). The system is built upon the Retrieval-Augmented Generation (RAG) framework, where data from official departmental documents are preprocessed and indexed into a structured knowledge base to support semantic retrieval and answer generation. Recognizing the limitations of a Naive RAG approach, we propose a series of improvements across three key phases: data preprocessing embedding-based document retrieval, and context-aware answer generation. To evaluate the effectiveness of the proposed system, we conduct experiments using several semantic embedding models, including those developed by Google, Cohere, an open-source model fine-tuned for the Vietnamese language. Our results demonstrate that the enhanced Advanced RAG model significantly outperforms the baseline in terms of answer accuracy, response fluency, and user satisfaction. These findings highlight the system's potential as a scalable AI-powered support solution for students in Vietnamese universities and contribute to the broader application of natural language processing in education.

Keywords Virtual assistant; Consulting chatbot; RAG; LLM; Student support services