Lưu Thị Ngọc Quỳnh, Nguyễn Hữu Xuân Trường (2025). Xây dựng trợ lý ảo thông minh cho sinh viên Học viện Chính sách và Phát triển. *Tạp chí nghiên cứu Chính sách và Phát triển,* 04(2025),164-172

DOI: https://doi.org/10.63640/3030-4091/jpd.apd.120

Tạp chí Nghiên cứu Chính sách và Phát triển

Bài báo khoa học

Xây dựng trợ lý ảo thông minh cho sinh viên Học viện Chính sách và Phát triển

Lưu Thị Ngọc Quỳnh
Nguyễn Hữu Xuân Trường (TS)
Học viện Chính sách và Phát triển
Email: luuthingocquynh04@gmail.com

Ngày nhận bài: 05/03/2025
Bản sửa lại lần 1: 25/04/2025
Ngày duyệt bài: 10/05/2025

Mã số: TC130425

Tóm tắt:

Sinh viên Học viện Chính sách và Phát triển luôn cần giải đáp những thắc mắc liên quan đến các quy định của Học viện hay cần được tư vấn về các vấn đề liên quan trong quá trình học tập và rèn luyện tại Học viện. Xuất phát từ nhu cầu này, nhóm đã nghiên cứu xây dựng trợ lý ảo thông minh dưới dạng một chatbot thân thiện. Dựa trên sự kết hợp giữa mô hình ngôn ngữ lớn (LLM) và kỹ thuật tạo sinh tăng cường (RAG), chatbot được thiết kế thành một bộ giao diện lập trình ứng dụng (API) với WebSocket, giúp mô hình có thể tương tác liên tục với người dùng. Nhóm đã triển khai thiết kế với Gemini 1.5 Flash và dựng API bằng FastAPI, kết hợp với xây dựng giao diện đơn giản bằng React. Chatbot hoạt động ổn định với độ chính xác 85%. Hệ thống bước đầu thể hiện tiềm năng phát triển với sự linh hoạt từ cách triển khai đến khả năng tích hợp các chức năng mới trong tương lai. Từ kết quả thu được, nhóm tiếp tục cải tiến để đem đến trải nghiệm tốt hơn cho người dùng.

Từ khóa: *Trợ lý ảo, chatbot, mô hình ngôn ngữ lớn, RAG, Gemini*

Abstract:

Students at the Academy of Policy and Development frequently require support in understanding institutional regulations and seeking advice related to their academic and personal development. In response to this need, our team developed an intelligent virtual assistant in the form of a user-friendly chatbot. Leveraging the capabilities of Large Language Models (LLM) combined with Retrieval-Augmented Generation (RAG) techniques, the chatbot was architected as an Application

Programming Interface (API)-based system with WebSocket integration to ensure real-time, continuous interaction. The system was implemented using Gemini 1.5 Flash for the core model, FastAPI for the backend, and a minimal React-based frontend. The chatbot has demonstrated stable performance, achieving an accuracy rate of approximately 85%. These initial results highlight the system's potential for future development, particularly in terms of deployment flexibility and scalability to integrate additional functionalities. Building on this foundation, we are continuing to optimize the system to enhance user experience and expand its application scope.

Keywords: *Virtual assistant, chatbot, LLM, RAG, Gemini*

1. Giới thiệu

Hiện nay, việc ứng dụng trí tuệ nhân tạo (AI) nói chung và mô hình xử lý ngôn ngữ tự nhiên (NLP) nói riêng hỗ trợ các hoạt động giảng dạy, học tập và rèn luyện tại trường học đang được quan tâm nghiên cứu và phát triển. Với trí tuệ nhân tạo tạo sinh (Genertive AI), cụ thể là mô hình ngôn ngữ lớn (LLM), học viên có thể sử dụng để tra cứu thông tin, giải đáp thắc mắc về nhà trường, về các thông tin quy định, chính sách của trường, tư vấn học tập, hỗ trợ nghiên cứu hoặc đổi mới sáng tạo. Có thể nói, sự phổ biến của các mô hình ngôn ngữ lớn đã hoàn toàn thay đổi cách thức một trợ lý ảo hỗ trợ người dùng nói chung và học viên nói riêng. Thay vì chỉ là một trợ lý ảo với các truy vấn dữ liệu đơn giản và tạo câu trả lời hạn chế, giờ đây trợ lý ảo có khả năng trả lời các thắc mắc của người dùng một cách tự nhiên, thân thiện và dễ hiểu (Kooli, 2023).

Nền móng cho sự phát triển của các mô hình ngôn ngữ lớn đã có từ những năm 90 của thế kỷ 20 với sự xuất hiện của mô hình IBM alignment. Tuy nhiên sự phát triển và bùng nổ của các mô hình ngôn ngữ lớn được đánh dấu với sự ra mắt của ChatGPT vào năm 2022. Dù đã phát triển GPT-1 từ 2018 và GPT-2 năm 2019 nhưng vì chưa kiểm soát được khả năng lạm dụng cho các hành vi độc hại nên chưa thể công bố phiên bản chính thức. Chỉ đến khi phiên bản GPT-3.5 được phát triển và triển khai, OpenAI mới công bố chính thức và trở thành cơn sốt trên thế giới. Đến thời điển hiện tại, dù mới chỉ là cơn sốt trong khoảng 3 năm trở lại song các công trình ứng dụng các mô hình ngôn ngữ lớn trong mọi hoạt động của đời sống xã hội đã được nghiên cứu và triển khai, một số trở thành công cụ đắc lực cho các hoạt động chuyên biệt.

Bên cạnh đó, việc cần hỗ trợ trong các hoạt động học tập và rèn luyện của học viên luôn là nhu cầu cấp thiết. Học viên khi cần được trợ giúp có thể tra cứu thông tin thông qua các công cụ tìm kiếm, tìm trên trang web của trường hay các đặt câu hỏi tại các nhóm hỗ trợ hay tại các trang thông tin của nhà trường. (Phan và cộng sự, 2024). Nếu cần, học viên có thể lên phòng Công tác sinh viên hoặc phòng Đào tạo để được giải đáp thắc mắc. Có thể thấy, việc có một giải pháp tất cả trong một trong việc giải đáp các thắc mắc của học viên trong quá trình học tập và rèn luyện sẽ tiết kiệm được rất nhiều thời gian. Thay vì phải tra cứu thủ công hay phải đến tận các phòng ban để hỏi thì học viên có thể hỏi trợ lý ảo, mọi công việc tra cứu sẽ được thực hiện tự động, trong trường hợp không tra cứu được có thể đưa ra lời khuyên và hướng giải quyết linh hoạt cho học viên. Chính vì vậy, việc nghiên cứu ứng dụng LLM xây dựng trợ lý ảo thông minh được nghiên cứu và phát triển.

Nghiên cứu được cấu trúc thành 5 phần, gồm: giới thiệu về bối cảnh nghiên cứu; cơ sở lý thuyết cho nghiên cứu; phương pháp nghiên cứu và hướng triển khai và thu thập đánh giá; kết quả thu được và đánh giá sơ bộ; kết luận và hướng phát triển của trợ lý ảo thông minh.

2. Cơ sở lý thuyết

Trong quá trình xây dựng trợ lý ảo, nhóm nghiên cứu đã tham khảo và tích hợp nhiều lý thuyết nền tảng từ lĩnh vực trí tuệ nhân tạo, đặc biệt là các mô hình ngôn ngữ lớn (LLM), kỹ thuật tạo sinh tăng cường (RAG), cùng với phương pháp triển khai hệ thống thời gian thực qua WebSocket. Phần này trình bày các khái niệm chính, các nghiên cứu liên quan và cách chúng được ứng dụng trong mô hình đề xuất.

2.1. Mô hình ngôn ngữ lớn (LLM)

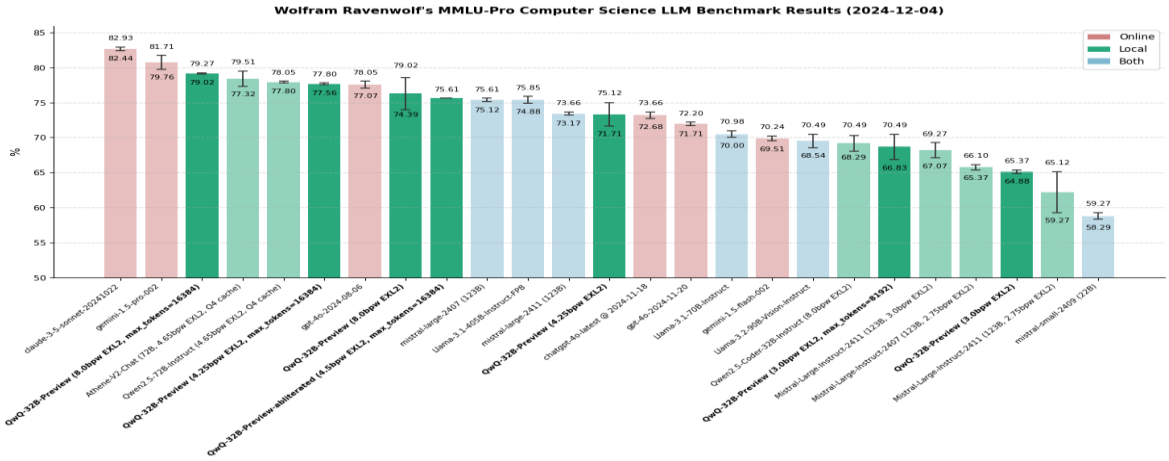
Mô hình ngôn ngữ lớn là một loại trí tuệ nhân tạo tiên tiến có khả năng xử lý và tạo sinh ngôn ngữ tự nhiên. Mô hình ngôn ngữ lớn là cốt lõi của các hệ thống chatbot thông minh hiện tại khi vừa có thể truy xuất thông tin từ dữ liệu đã được huấn luyện, vừa có thể tạo ra những câu trả lời gần với tự nhiên, giúp người sử dụng có thể hiểu các thông tin cần thiết một cách dễ dàng (Trần và cộng sự, 2024). Các mô hình ngôn ngữ lớn phổ biến hiện nay như ChatGPT, ClaudeAI, Gemini, Deepseek,… không chỉ đưa ra câu trả lời thông thường từ câu hỏi của người dùng mà còn có thể thực hiện được nhiều tác vụ khác nhau như suy luận logic, tạo ảnh và video, đọc hiểu văn bản, lập trình, tra cứu thông tin mạng, … Ngoài ra, các mô hình này còn cung cấp giải pháp cho các nhà phát triển ứng dụng thông qua API, mở ra khả năng sáng tạo vô tận cho các nhà phát triển.

Tuy vậy, mô hình ngôn ngữ lớn vẫn có những hạn chế. Một trong số những hạn chế ảnh hưởng trực tiếp đến việc xây dựng và triển khai chatbot là khả năng tiếp cận với dữ liệu nội bộ và các dữ liệu mới, tùy chỉnh. Mô hình ngôn ngữ lớn hiện hành vốn được xây dựng để giải quyết các vấn đề chung nên với những bài toán và đối tượng cụ thể, mô hình ngôn ngữ có thể đưa ra những câu trả lời với độ chính xác không cao.

Để giải quyết vấn đề này, các phương pháp đã được đưa ra và sử dụng. Đầu tiên, chất lượng câu trả lời phụ thuộc vào cách thức mô hình hiểu được câu hỏi của người dùng. Do đó việc đưa ra một câu hỏi dễ hiểu với mô hình sẽ cải thiện đáng kể khả năng trả lời câu hỏi đúng trọng tâm. Đây là cơ sở của kỹ thuật tạo câu lệnh, hay prompt engineering. Ở đó, các câu hỏi sẽ được đưa vào một khuôn lệnh, từ đó mới đưa vào mô hình để tạo câu trả lời. Kỹ thuật này giúp giới hạn chủ đề trả lời, đưa ra cách trả lời sát với chủ đề câu hỏi và có thể chỉnh sửa phong cách trả lời của mô hình. Một khuôn lệnh đủ tốt sẽ làm mô hình trả lời một cách tự nhiên mà vẫn đảm bảo độ chính xác. Hiện nay, ngoài việc chèn trực tiếp khuôn lệnh vào câu hỏi, các mô hình còn cho phép tạo sẵn các khuôn lệnh và chèn vào mô hình dưới dạng chỉ thị hệ thống (system instruction). Với chỉ thị hệ thống, ta có thể yêu câu mô hình trả lời theo cách thức mà ta mong muốn mà vẫn tách biệt với câu hỏi của người dùng, đảm bảo tối ưu cho mô hình.

Một phương pháp khác cũng được sử dụng đó là fine-tuning. Với fine-tuning, ta có thể huấn luyện một mô hình có sẵn trả lời chính xác các câu hỏi trong phạm vi kiến thức hẹp hơn thông qua việc nạp dữ liệu riêng vào

mô hình và từ đó mong muốn cách trả lời với dữ liệu riêng đó. Do đó, mô hình mới sẽ có đủ kiến thức và phong cách mong muốn. Tuy nhiên điểm yếu của fine-tuning là tính linh hoạt sẽ giảm vì dữ liệu nạp vào là dữ liệu tĩnh và có thể dẫn tới tình trạng quá khớp, khiến câu trả lời cho các lĩnh vực khác không còn chính xác. Để giải quyết vấn đề này, đảm bảo cung cấp tri thức đầy đủ cho mô hình, cách kỹ thuật nạp dữ liệu động đã được nghiên cứu và phát triển. Một trong số đó là Kỹ thuật Tạo sinh Tăng cường (Retrieval Augmented Generation - RAG).



Hình 1: Kết quả thông số MMLU-Pro Computer Science LLM của Wolfram Ravenwolf

Nguồn: LLM Comparison/Test: 25 SOTA LLMs (including QwQ) through 59 MMLU-Pro CS benchmark runs. (2024).

2.2. Tạo sinh tăng cường truy xuất (RAG)

Retrieval-Augmented Generation (RAG) hay còn gọi là Tạo sinh tăng cường truy xuất, là một kỹ thuật trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) giúp cải thiện chất lượng và độ chính xác của các mô hình ngôn ngữ lớn (LLM). Thay vì chỉ dựa vào kiến thức có sẵn trong quá trình huấn luyện, RAG cho phép mô hình truy xuất thông tin từ các nguồn dữ liệu bên ngoài (như cơ sở dữ liệu, tài liệu, web) và sử dụng thông tin này để tạo ra phản hồi. RAG cung cấp khả năng truy vấn dữ liệu lớn, đảm bảo tính chính xác về ngữ nghĩa và từ khóa cho các dữ liệu sẽ được thêm vào để kèm với câu hỏi tạo ra một câu lệnh (prompt) cho mô hình ngôn ngữ lớn hiểu và sinh ra câu trả lời chính xác.

Hiện tại, có hai phương pháp triển khai RAG cơ bản là VectorRAG và GraphRAG. Với VectorRAG, các khối dữ liệu sẽ được lưu trữ và được đánh dấu bởi một véc-tơ trong không gian với số chiều xác định, đã được tính toán dựa trên ngữ nghĩa của từng từ trong khối dữ liệu. Số chiều này sẽ quyết định độ chính xác và mức độ phân loại của các véc-tơ trong không gian lưu trữ. Số chiều càng lớn, các véc-tơ càng có sự phân loại tốt, song lưu trữ sẽ càng nặng và có thể trở nên thừa thãi nếu số lượng véc-tơ không quá nhiều. Khi một câu hỏi được đưa ra, câu hỏi đó sẽ được tính toán thành một véc-tơ và véc-tơ đó sẽ được dùng để truy xuất các khối dữ liệu có nội dung tương tự mà câu hỏi đang đề cập, thường sẽ gần với véc-tơ câu hỏi. VectorRAG thích hợp cho việc truy xuất dữ liệu yêu cầu ngữ nghĩa có liên quan đến câu hỏi mà không cần độ chính xác tuyệt đối. Với GraphRAG, dữ liệu sẽ được lưu dưới dạng các nút nối tiếp nhau trên mạng lưới, theo một quy luật liên kết

logic nào đó và thực hiện truy xuất giống với cơ sở dữ liệu thông thường. GraphRAG thích hợp với những kiểu dữ liệu có ràng buộc ngữ nghĩa và yêu cầu độ chính xác cao. Cả hai phương pháp này đều có điểm mạnh và điểm yếu riêng nên ta có thể thực hiện kết hợp các giải pháp khác nhau để tạo thành một RAG hoàn chỉnh thông qua việc mô-đun hóa RAG, chia tỉ lệ phụ thuộc ngữ nghĩa, từ khóa hay tạo ra thang chấm điểm kết quả phù hợp.

2.3. FastAPI và WebSocket

FastAPI là một framework Python hiện đại, hiệu suất cao, được thiết kế để xây dựng API nhanh chóng và dễ bảo trì. Với sự hỗ trợ mạnh mẽ cho kiểu dữ liệu, FastAPI giúp phát triển các ứng dụng web và RESTful API dễ dàng hơn bằng cách tận dụng khả năng kiểm tra kiểu dữ liệu (type hints) của Python. Một trong những tính năng quan trọng của FastAPI là hỗ trợ WebSocket, giúp xây dựng các ứng dụng realtime như chat, streaming dữ liệu, và dashboard cập nhật liên tục.

Bằng cách kết hợp FastAPI với WebSocket, các nhà phát triển có thể tận dụng hiệu suất cao của ASGI (Asynchronous Server Gateway Interface) để xử lý hàng nghìn kết nối đồng thời mà không làm giảm tốc độ phản hồi. Điều này giúp xây dựng các ứng dụng có khả năng mở rộng tốt, đồng thời vẫn giữ được mã nguồn gọn gàng, dễ bảo trì.

3. Phương pháp nghiên cứu

Với lý thuyết đã thu được, nhóm thực hiện triển khai nhằm đưa ra đánh giá và tiếp tục cải thiện.

3.1. Lựa chọn mô hình ngôn ngữ lớn

Hiện nay trên thị trường có rất nhiều bên cung cấp mô hình ngôn ngữ lớn mạnh mẽ và dễ dàng triển khai, song việc lựa chọn mô hình còn phụ thuộc vào yêu cầu bài toán. Hình 1 cho thấy điểm của model Claude-3.5-sonnet là cao nhất, song để tiết kiệm chi phí phát triển mà vẫn đảm bảo được độ chính xác thì mô hình Gemini Flash đã được lựa chọn. Đây là mô hình ngôn ngữ tuy nhỏ hơn bản Pro nhưng đưa ra câu trả lời với chất lượng ổn mà vẫn đảm bảo chi phí. Chưa kể mô hình Pro của Gemini chỉ có thông số đánh giá thấp hơn Claude một chút. Đến thời điểm hiện tại, Gemini Flash đã có bản 2.5 ổn định, dù vậy trong khuôn khổ nghiên cứu, mô hình được lựa chọn là gemini-2.0-flash.

3.2. Thiết kế khuôn dữ liệu và triển khai RAG

Để đảm bảo khả năng phản hồi chính xác và dựa trên cơ sở dữ liệu có thật, nghiên cứu sử dụng kỹ thuật Tạo sinh Tăng cường (Retrieval-Augmented Generation – RAG). Trước hết, các tài liệu nội bộ của Học viện Chính sách và Phát triển như quy chế đào tạo, chính sách học bổng, hướng dẫn học vụ, văn bản hành chính,... được thu thập, xử lý và chuyển đổi sang dạng văn bản thuần túy (text). Sau đó, nội dung được chia thành các đoạn ngữ nghĩa nhỏ (chunk) với độ dài phù hợp để mô hình dễ dàng tiếp nhận và xử lý.

Các đoạn văn bản này được véc-tơ hóa sử dụng mô hình nhúng văn bản (embedding) tương thích với mô hình Gemini. Sau đó, dữ liệu được lập chỉ mục và lưu trữ bằng công cụ FAISS – một thư viện tối ưu hóa tìm kiếm gần đúng hiệu quả cao. Khi người dùng gửi câu hỏi, hệ thống tiến hành truy vấn ngữ nghĩa từ kho dữ liệu đã lập chỉ mục, sau đó truyền kết quả vào mô hình ngôn ngữ để sinh phản hồi dựa trên thông tin truy xuất được. Điều này giúp hạn chế hiện tượng mô hình "tự tưởng tượng" thông tin (hallucination), đồng thời đảm bảo câu trả lời có tính xác thực và cập nhật.

3.3. Thiết kế API và giao diện đơn giản

Hệ thống được triển khai theo mô hình máy khách - máy chủ, trong đó lớp API trung gian đóng vai trò kết nối và xử lý yêu cầu giữa người dùng và mô hình trí tuệ nhân tạo. FastAPI được lựa chọn để xây dựng hệ thống API nhờ khả năng xử lý bất đồng bộ, hiệu suất cao và hỗ trợ WebSocket một cách tối ưu. Giao thức WebSocket giúp duy trì kết nối liên tục giữa máy khách và máy chủ, đảm bảo trải nghiệm hội thoại theo thời gian thực và giảm độ trễ trong quá trình phản hồi.

Giao diện người dùng được thiết kế bằng React với tiêu chí đơn giản, dễ sử dụng và thân thiện với sinh viên. Các chức năng chính bao gồm khung trò chuyện, lịch sử trao đổi, gợi ý câu hỏi phổ biến và nút đánh giá chất lượng phản hồi. Giao diện cũng được tối ưu hóa để tương thích với thiết bị di động, giúp người dùng có thể truy cập mọi lúc, mọi nơi.

3.4. Tiêu chí đánh giá

Để kiểm chứng hiệu quả của hệ thống trợ lý ảo, nhóm nghiên cứu đề xuất các tiêu chí đánh giá như sau:

− Độ chính xác nội dung phản hồi: So sánh câu trả lời từ chatbot với đáp án đúng được trích xuất từ tài liệu chính thức. Tỷ lệ chính xác được tính dựa trên số câu trả lời hoàn toàn đúng và có căn cứ.

− Thời gian phản hồi: Đo độ trễ trung bình từ thời điểm người dùng gửi câu hỏi đến khi hệ thống phản hồi, nhằm đảm bảo trải nghiệm mượt mà và tức thì.

− Khả năng xử lý truy vấn đồng thời: Kiểm thử hệ thống với nhiều truy vấn diễn ra cùng lúc nhằm đánh giá tính ổn định và khả năng mở rộng.

− Mức độ hài lòng của người dùng: Thu thập phản hồi từ người sử dụng thử thông qua bảng khảo sát để đánh giá giao diện, độ chính xác, độ nhanh và tính tiện ích chung.

4. Kết quả nghiên cứu và thảo luận

4.1. Kết quả nghiên cứu

Với cơ sở lý thuyết và thiết kế hệ thống kể trên, nhóm nghiên cứu đã xây dựng một nguyên mẫu chatbot hỗ trợ sinh viên dưới dạng một trang web đơn giản. Cốt lõi của chatbot đến từ phía máy chủ với các công nghệ AI được áp dụng, triển khai dưới dạng một bộ API đơn giản. Điều này phù hợp với mục đích thực hiện đề tài của nhóm nghiên cứu, xây dựng được một mô-đun có khả năng tích hợp vào hệ thống của nhà trường.Dữ liệu được sử dụng để tra cứu RAG là các quy định, chính sách của Học viện được viết dưới dạng văn bản, đã được phân nhỏ và tái cấu trúc phù hợp và được lưu trữ nội bộ tại máy chủ triển khai. Các lời gọi API được thiết kế bao gồm một API POST phục vụ việc nạp dữ liệu vào máy chủ thông qua tệp dữ liệu ngăn cách bởi dấu phẩy (.csv) và một API WebSocket là triển khai chatbot thực hiện trả lời câu hỏi từ người dùng.

Sau khi hoàn thiện hệ thống trợ lý ảo thông minh, nhóm nghiên cứu đã tiến hành triển khai thử nghiệm với một tập hợp các câu hỏi thực tế từ sinh viên Học viện Chính sách và Phát triển. Các kết quả kiểm thử được trình bày như sau:

Bảng 1. Đánh giá độ chính xác phản hồi của chatbot theo từng nhóm nội dung

Nhóm câu hỏi	Số câu hỏi	Số câu trả lời đúng	Tỷ lệ chính xác (%)
Quy chế đào tạo	20	17	85.0%
Học phí – học bổng	20	18	90.0%
Thủ tục hành chính	20	16	80.0%
Đăng ký tín chỉ - lịch học	20	17	85.0%
Câu hỏi khác (cơ sở vật chất,…)	20	16	80.0%
Tổng cộng	100	84	84.0%

Nguồn: Kết quả đo lường nội bộ của nhóm nghiên cứu.

Ngoài ra, các tiêu chí khác cũng được ghi nhận như sau:Thời gian phản hồi trung bình: Khoảng 1,89 giây/truy vấn.

− Khả năng xử lý đồng thời: Hệ thống hoạt động ổn định với 100 kết nối đồng thời mà không xảy ra lỗi hoặc trễ bất thường.

− Khảo sát trải nghiệm người dùng: Trong số 30 sinh viên tham gia thử nghiệm, 86% đánh giá hệ thống "hữu ích" hoặc "rất hữu ích", đặc biệt trong các trường hợp cần tra cứu nhanh và ngoài giờ hành chính.

4.2. Thảo luận

Kết quả thử nghiệm cho thấy hệ thống trợ lý ảo đã đạt được mức độ chính xác đáng tin cậy, đặc biệt đối với các nhóm câu hỏi phổ biến như quy chế đào tạo, học phí – học bổng, hay lịch học. Việc tích hợp kỹ thuật RAG đã giúp cải thiện đáng kể chất lượng phản hồi so với các hệ thống chatbot chỉ dựa trên rule-based hoặc retrieval thuần túy. Độ trễ thấp và khả năng duy trì kết nối thời gian thực nhờ WebSocket cũng góp phần nâng cao trải nghiệm người dùng, khiến chatbot trở nên hữu ích trong môi trường học đường.So với các giải pháp hiện có (ví dụ như cổng hỏi – đáp FAQ tĩnh hoặc chatbot nền tảng mạng xã hội), hệ thống đề xuất trong nghiên cứu này có lợi thế rõ rệt về độ linh hoạt, khả năng mở rộng, và mức độ tùy biến theo dữ liệu nội bộ của đơn vị giáo dục. Tuy nhiên, vẫn còn một số hạn chế như: hệ thống hiện mới dừng lại ở mức xử lý câu hỏi đơn giản theo từng lượt riêng biệt, chưa tối ưu cho hội thoại phức tạp nhiều lượt; việc cập nhật dữ liệu mới cần thao tác thủ công; và độ chính xác có thể bị ảnh hưởng nếu truy vấn thuộc về các chủ đề chưa được khai báo đầy đủ trong kho dữ liệu.

Những hạn chế trên là cơ sở cho các hướng phát triển tiếp theo như: tích hợp chức năng quản lý hội thoại theo ngữ cảnh (contextual memory), bổ sung cơ chế tự động cập nhật dữ liệu văn bản mới, hoặc mở rộng khả năng đa ngôn ngữ để hỗ trợ sinh viên quốc tế.

5. Kết luận & Gợi ý

5.1. Kết luận

Nghiên cứu đã trình bày quá trình xây dựng và triển khai một hệ thống trợ lý ảo thông minh ứng dụng mô hình ngôn ngữ lớn kết hợp kỹ thuật RAG nhằm hỗ trợ sinh viên Học viện Chính sách và Phát triển trong việc tra cứu thông tin học vụ. Hệ thống được thiết kế theo hướng tối ưu chi phí, đảm bảo hiệu suất với mô hình Gemini Flash 2.0 và cơ chế

truy xuất ngữ nghĩa từ dữ liệu chính thức.

Kết quả thử nghiệm cho thấy hệ thống đạt độ chính xác phản hồi khoảng 85%, thời gian phản hồi nhanh, khả năng phục vụ truy vấn đồng thời ổn định và được đánh giá tích cực từ người dùng thử. Những kết quả này cho thấy tính khả thi và hiệu quả ban đầu của giải pháp, đặc biệt trong bối cảnh chuyển đổi số đang được đẩy mạnh tại các cơ sở giáo dục đại học ở Việt Nam.

5.2. Gợi ý

Từ kết quả nghiên cứu, nhóm đề xuất một số gợi ý nhằm tiếp tục hoàn thiện và ứng dụng rộng rãi hệ thống:

− Đối với nhà trường: Nên xem xét áp dụng thí điểm trợ lý ảo vào các bộ phận thường xuyên tiếp nhận câu hỏi từ sinh viên như Phòng Quản lý Đào tạo, Phòng Chính trị và Công tác sinh viên, Khoa chuyên môn... để giảm tải khối lượng công việc và nâng cao chất lượng phục vụ.

− Về mặt kỹ thuật: Cần phát triển thêm khả năng ghi nhớ ngữ cảnh hội thoại, mở rộng dữ liệu truy vấn theo thời gian thực và tự động hóa quy trình cập nhật nội dung văn bản mới.

− Đối với nghiên cứu tương lai: Hướng tới xây dựng hệ thống hỗ trợ đa ngôn ngữ, tích hợp giọng nói và kết nối với các nền tảng học tập trực tuyến để tạo ra một hệ sinh thái hỗ trợ học tập toàn diện.

Việc phát triển các giải pháp công nghệ thân thiện, hiệu quả và chi phí thấp như trợ lý ảo trong môi trường giáo dục không chỉ giúp cải thiện trải nghiệm của sinh viên mà còn góp phần hiện đại hóa hoạt động quản trị tại các cơ sở giáo dục đại học trong kỷ nguyên số.

TÀI LIỆU THAM KHẢO

Facebook AI Research. (2023). FAISS - A library for efficient similarity search. Retrieved from https://github.com/facebookresearch/faiss

Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). Sebastopol, CA: O'Reilly Media.

Google Cloud. (2023). Cloud Vision API documentation. Retrieved from https://cloud.google.com/vision/docs

Google DeepMind. (2024). Gemini 1.5 technical report & Google AI Studio. Retrieved from https://deepmind.google/technologies/gemini

Học viện Chính sách và Phát triển. (2023). Quy chế đào tạo và văn bản học vụ. Retrieved from https://apd.edu.vn

Hugging, F. (2023a). *Retrieval-augmented generation (RAG)*. Retrieved from https://huggingface.co/docs/transformers/model_doc/rag

Hugging, F. (2023b). *Transformers documentation - Large language models*. Retrieved from https://huggingface.co/docs/transformers/index

Jurafsky, D., & Martin, J. H. (2019). Speech and language processing (3rd ed.). Stanford, CA: Stanford University.

Kooli, C. (2023). Chatbots in Education and Research: A Critical Examination of Ethical Implications and Solutions. *Sustainability*, 15(7), 5614. https://doi.org/10.3390/su15075614

Phan, T. K., Nguyễn Đ. C., & Đinh, T. D. (2024). Ứng dụng trí tuệ nhân tạo trong dạy học và nghiên cứu khoa học tại các trường đại học. *Tạp chí Giáo dục*, 24(24), 14–19. Truy vấn

từ https://tcgd.tapchigiaoduc.edu.vn/index.php/tapchi/article/view/2677

Python WebSockets Library. (2023). WebSocket for Python. Retrieved from https://websockets.readthedocs.io

Raaijmakers, S. (2022). Deep learning for natural language processing. Shelter Island, NY: Manning Publications.

Russell, S., & Norvig, P. (2022). Artificial intelligence: A modern approach (4th ed.). Upper Saddle River, NJ: Pearson Education.

SentenceTransformers. (2023). All-MiniLM-L6-v2: Pretrained models. Retrieved from https://www.sbert.net/docs/pretrained_models.html

Tesseract OCR. (2023). Tesseract OCR engine. Retrieved from https://github.com/tesseract-ocr/tesseract

Trần, K. T., Đinh, M. H., Phạm, N. B., Huỳnh, V. L., Lê, H. N., & Nguyễn, T. T. A. (2024). Mô hình ngôn ngữ lớn và ứng dụng trong giáo dục đại học. Tạp chí Khoa học HUFLIT, 14(1), 55–65.

VinAI Research. (2021a). *Vietnamese-BERT*. Retrieved from https://github.com/VinAIResearch/BERTweet

VinAI Research. (2021b). PhoBERT và ViT5 - Mô hình ngôn ngữ tiếng Việt. Retrieved from https://github.com/VinAIResearch/PhoWhisper