Kính thưa quý thầy trong hội đồng bảo vệ luận văn

Hôm nay, nhóm chúng em là Nguyễn Việt Pha và Cao Thành Nam sẽ báo cáo luận văn tốt nghiệp với đề tài: “Xây dựng chatbot tư vấn học vụ Nông Lâm kết hợp giữa hệ thống GRAG (Graph Retrieval-Augmented Generation) và RAG (Retrieval-Augmented Generation)”. Bao gồm các nội dung chính: giới thiệu đề tài, cơ sở lý thuyết, phương pháp luận, triển khai thực hiện, kết quả đạt được, kết luận và hướng phát triển, cùng với phần demo. Chúng em hy vọng sẽ mang đến một cái nhìn rõ ràng và toàn diện về công trình nghiên cứu này.

**1. Giới thiệu đề tài**

Trước tiên, em xin phép trình bày về mục tiêu và phạm vi của đề tài. Trong bối cảnh nhu cầu tư vấn học vụ tại các trường đại học ngày càng tăng, việc xây dựng một chatbot thông minh, có khả năng trả lời chính xác các câu hỏi dựa trên tài liệu học vụ là một hướng đi đầy tiềm năng. Đề tài của chúng em đặt mục tiêu:

Sử dụng LLM Gemini-1.5-flash làm nền tảng để xây dựng hệ thống hỏi đáp.

Tìm hiểu và kết hợp hai tiên tiến: RAG và Graph-RAG cùng với một Critic mô-đun để nâng cao độ chính xác của câu trả lời.

So sánh hiệu quả của kiến trúc kết hợp này với RAG và GRAG truyền thống.

Về phạm vi, nghiên cứu tập trung vào việc xử lý tài liệu PDF, cụ thể là sổ tay sinh viên của Trường Đại học Nông Lâm năm 2024. Ứng dụng được phát triển sẽ nhận đầu vào là một tệp PDF và một câu hỏi, sau đó trả lời dựa trên nội dung tài liệu.

**2. Cơ sở lý thuyết**

Tiếp theo, em xin trình bày cơ sở lý thuyết của RAG và GRAG

**2.1 RAG (Retrieval-Augmented Generation)**

RAG là một phương pháp kết hợp giữa truy xuất thông tin và sinh văn bản, nhằm khắc phục hạn chế về dữ liệu của các mô hình ngôn ngữ lớn. RAG hoạt động bằng cách sử dụng ngữ cảnh bên ngoài, để bổ sung kiến thức cho LLM.

Question: Câu hỏi người dùng đặt ra.

Context: Cơ sở dữ liệu vector lưu trữ thông tin.

Retriever: Thành phần truy xuất ngữ cảnh phù hợp từ cơ sở dữ liệu.

LLM: Mô hình ngôn ngữ lớn sử dụng ngữ cảnh để sinh câu trả lời.

Response: Kết quả trả lời của hệ thống.

Để tối ưu hóa RAG, chúng em áp dụng các kỹ thuật như sử dụng vector index tối ưu hóa cho tìm kiếm dense vector. Chiến lược tìm kiếm nhiều giai đoạn làm tăng tính chính xác và hiệu suất. Chia đoạn văn bản bằng LLM.

Hạn chế lớn nhất của RAG là bỏ qua mối quan hệ cấu trúc, hạn chế về độ sâu ngữ cảnh, và khó xử lý các truy vấn đòi hỏi suy luận nhiều bước.

**2.2 GRAG (Graph Retrieval-Augmented Generation)**

GraphRAG ra đời nhằm tận dụng cấu trúc đồ thị để biểu diễn mối quan hệ giữa các thực thể trong tài liệu. GRAG có ưu điểm trong việc xử lý các truy vấn đa bước. Tuy nhiên, GRAG cũng đối mặt với thách thức như chi phí xây dựng đồ thị cao, khó mở rộng, và phụ thuộc vào chất lượng đồ thị.

**3. Phương pháp luận**

Để giải quyết bài toán, em xây dựng một kiến trúc hoàn chỉnh gồm các thành phần:

Agent: Ra quyết định và chọn nguồn tài liệu phù hợp cho câu hỏi.

Retriever Bank: Gồm 2 môn-đun. mô-đun truy xuất văn bản và mô-đun truy xuất đồ thị.

Generator: LLM tổng hợp câu trả lời dựa trên ngữ cảnh truy xuất.

Critic Module: Đánh giá và cung cấp phản hồi để cải thiện câu trả lời.

Quy trình thực hiện

Thứ nhất là chuẩn bị mô hình: Sử dụng Gemini-1.5-flash cho tất cả các thành phần: Agent, Validator, Commentor, và Generator.

Thứ hai là chuẩn bị dữ liệu: Sử dụng sổ tay sinh viên Nông Lâm 2024 làm tài liệu chính lưu vào qdrant và neo4j. thứ ba là tiền xử lý dữ liệu

Cuối cùng là so sánh hiệu quả của kiến trúc kết hợp (RAG+GRAG+Agent) với RAG và GRAG truyền thống dựa trên các chỉ số: độ tương đồng, độ chính xác (accuracy), hiện tượng ảo giác (hallucination), và thông tin bị thiếu (missing).

**4. Triển khai thực hiện**

**4.1 Xây dựng nguồn kiến thức cho RAG**

Em sử dụng thư viện **pdfplumber** để trích xuất nội dung từ PDF và Gemini-2.5-Pro-Preview-05-06 để chunking và tạo tóm tắt.

Đây là các mô hình nhúng được em chọn để nhúng

Để chọn các mô hình nhúng em tiến hành đánh giá các mô hình trên mteb ranking và đề xuất bởi AI. Các mô hình này được đánh giá trên tập dữ liệu 700 câu hỏi thuộc 7 lĩnh vực để chọn ra mô hình có độ tương đồng cao nhất.

**4.2 Xây dựng mô-đun tìm kiếm cho RAG**

Mô-đun tìm kiếm sử dụng kỹ thuật truy vấn nhiều bước do Qdrant hỗ trợ:

Bước 1: Truy vấn với vector 512 chiều, lấy ra 200 vector

Bước 2: Truy vấn với vector 768 chiều lấy ra 100 vectorr.

Bước 3: Truy vấn với vector 1024 chiều lấy ra 50 vector.

Bước 4: Sử dụng mô hình rerank llama để xếp hạng lại và chọn các chunk có relevence socre > 0.

**4.3 Xây dựng nguồn kiến thức cho GRAG**

Chúng em sử dụng LLM Gemini-2.5-Pro-Preview-05-06 để trích xuất các node và relationship, đồng thời tạo tiêu đề cho từng đoạn văn bản. Đồ thị được thiết kế phân tầng dựa trên mục lục của sổ tay sinh viên 2024.

**4.4 Xây dựng mô-đun tìm kiếm cho GRAG**

Sử dụng LLM để dự đoán câu hỏi thuộc phần nào trong mục lục, sau đó áp dụng truy vấn Cypher để truy xuất thông tin. Tuy nhanh và hiệu quả trong việc thu hẹp phạm vi tìm kiếm, phương pháp này có thể gặp lỗi nếu LLM tạo sai cú pháp hoặc dự đoán sai mục lục.

**4.5 Xây dựng Critic Module**

Đây là mô đun quan trọng nhất trong hệ thống, giúp Agent tự phản ánh và điều chỉnh hành động của mình một cách hợp lý

Agent: dự đoán câu hỏi truy xuất nguồn tài liệu nào

Generator: tạo ra câu trả lời dựa vào tài liệu được truy xuất

Valid: quyết định câu trả lời có đúng hay không. Nhiệm vụ nhị phân

Commentor: Nếu câu hỏi chưa đúng thì bình luận phản hồi cho Agent

Quá trình này lặp lại cho đến khi câu trả lời được xác nhận là đúng hoặc đạt giới hạn vòng lặp.

**4.6 Trả lời câu hỏi**

Đây là diagram mô phỏng rõ hơn cách hệ thống trả lời một câu hỏi

**4.7 So sánh và đánh giá**

Chúng em sử dụng tập dữ liệu 543 cặp câu hỏi-câu trả lời, được tạo từ sổ tay sinh viên 2024. Kết quả đánh giá trên 3 kiến trúc (RAG+GRAG+Agent, RAG, GRAG) với 3 mô hình nhúng cho thấy: