KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

ĐỀ CƯƠNG CHI TIẾT
KHÓA LUẬN TỐT NGHIỆP NGÀNH CÔNG NGHỆ THÔNG TIN

Họ tên sinh viên: **Nguyễn Đại Hoàng Phúc** MSSV: 110121087

Lớp: Công nghệ Thông tin B Khóa: 2021-2025

Tên đề tài: Nghiên cứu RAG và xây dựng chatbot hỗ trợ môn Cơ sở dữ liệu.

1. Mục tiêu của đề tài:

- Nghiên cứu sâu về lý thuyết và kiến trúc của Retrieval-Augmented Generation (RAG), các thành phần cốt lõi (embedding, vector database, large language models - LLMs).
- Tìm hiểu và áp dụng các kỹ thuật Xử lý Ngôn ngữ Tự nhiên (NLP) cần thiết cho việc xây dựng chatbot dựa trên RAG (tập trung vào tiền xử lý văn bản, embedding).
- Thu thập, tiền xử lý và tổ chức tài liệu học tập môn Cơ sở dữ liệu (giáo trình, bài giảng, bài tập,...) để xây dựng cơ sở tri thức dưới dạng vector embeddings cho chatbot.
- Phân tích, thiết kế và xây dựng một hệ thống chatbot hoàn chỉnh sử dụng kỹ thuật RAG với cơ chế truy xuất dựa trên vector (vector retrieval), có khả năng truy xuất thông tin từ cơ sở tri thức và sinh câu trả lời phù hợp, chính xác cho các câu hỏi liên quan đến môn Cơ sở dữ liệu.
- Xây dựng giao diện người dùng thân thiện (web-based) để sinh viên có thể tương tác dễ dàng với chatbot.

2. Nội dung thực hiện:

Chương 1: Tổng quan:

- Giới thiệu về lĩnh vực chatbot, trí tuệ nhân tạo và xử lý ngôn ngữ tự nhiên.
- Giới thiệu về kỹ thuật Retrieval-Augmented Generation (RAG) và tiềm năng ứng dụng.
- Đặt vấn đề: Nhu cầu hỗ trợ học tập môn Cơ sở dữ liệu và vai trò của chatbot.
- Xác định mục tiêu, đối tượng và phạm vi nghiên cứu của đề tài.
- Cấu trúc của khóa luận.

Chương 2: Cơ sở lý thuyết:

- Tổng quan về Xử lý Ngôn ngữ Tự nhiên (NLP): Các khái niệm cơ bản, kỹ thuật tiền xử lý văn bản.

- Kỹ thuật Embedding: Word embeddings, sentence embeddings, document embeddings (tập trung vào Sentence Transformers).
- Vector Database: Khái niệm, các loại hình, kỹ thuật tìm kiếm vector. Tập trung vào Qdrant (collections, points, payloads, similarity search).
- Large Language Models (LLMs): Kiến trúc (Transformer), các mô hình phổ biến (ví dụ: Gemini).
- Retrieval-Augmented Generation (RAG): Kiến trúc chi tiết (Retriever, Generator), các biến thể, ưu nhược điểm.
- Tổng quan về môn học Cơ sở dữ liệu: Các chủ đề chính cần chatbot hỗ trợ.

Chương 3: Phân tích và thiết kế hệ thống:

- Phân tích yêu cầu: Yêu cầu chức năng (hỏi đáp, quản lý feedback cơ bản,...) và phi chức năng (hiệu năng chấp nhận được, bảo mật cơ bản, khả năng mở rộng).
- Thiết kế kiến trúc tổng thể: Xác định các module chính (Data Ingestion, Embedding, Vector Retrieval, Generation, Postprocessing, API, Frontend), luồng dữ liệu và tương tác giữa các module.
- Thiết kế cơ sở tri thức:

  + Chiến lược tiền xử lý và phân đoạn (Chunking): Mô tả cách xử lý các loại tài liệu (PDF, text, SQL), chiến lược chunking được chọn (vd: RecursiveCharacterTextSplitter) và lý do.

  + Lựa chọn và thiết kế cấu trúc lưu trữ: Chỉ tập trung vào Vector Store (Qdrant), định dạng dữ liệu (points) và metadata (payloads) lưu kèm embeddings.
- Thiết kế luồng xử lý RAG: Chi tiết hóa quy trình: Tiền xử lý câu hỏi -> Tạo embedding cho câu hỏi -> Truy vấn Qdrant để tìm ngữ cảnh liên quan -> Chuẩn bị prompt với ngữ cảnh -> Gọi LLM (Gemini) để sinh câu trả lời -> Hậu xử lý cơ bản (định dạng, trích nguồn nếu có).
- Thiết kế API: Xác định các endpoints chính (vd: /chat, /feedback), định dạng request/response.
- Thiết kế giao diện người dùng (UI/UX): Phác thảo giao diện chat đơn giản, luồng tương tác cơ bản.

Chương 4: Triển khai hệ thống:

- Lựa chọn công nghệ: Ngôn ngữ (Python), framework backend (FastAPI), thư viện RAG/NLP (Langchain, Sentence Transformers), xử lý tài liệu (PyPDF, Pytesseract, BeautifulSoup4), vector DB (Qdrant), LLM (Gemini API), Frontend (HTML, CSS, JavaScript – Cơ bản).
- Xây dựng module Data Ingestion: Đọc, phân đoạn theo chiến lược đã thiết kế, và làm sạch tài liệu CSDL.
- Xây dựng module Embedding: Sử dụng Sentence Transformers để tạo vector embeddings cho các đoạn tài liệu và lưu vào Qdrant.
- Xây dựng module Vector Retrieval: Triển khai cơ chế tìm kiếm vector hiệu quả trên Qdrant dựa trên embedding của câu hỏi (similarity search).
- Xây dựng module Generation: Tích hợp Gemini API (sử dụng Langchain) để sinh câu trả lời dựa trên prompt chứa ngữ cảnh truy xuất được.
- Xây dựng module Postprocessing: Định dạng câu trả lời, hiển thị nguồn tham khảo (nếu metadata cho phép).
- Xây dựng Backend API: Triển khai các API đã thiết kế bằng FastAPI.
- Xây dựng Frontend: Triển khai giao diện người dùng web-based tương tác với Backend API.
- Tích hợp hệ thống: Kết nối các module, API và Frontend.
- Đóng gói và triển khai: Sử dụng Docker và Docker Compose nếu có thời gian.

- Chương 5: Đánh giá và kết quả:

- Xây dựng bộ dữ liệu đánh giá: Tập hợp các câu hỏi kiểm thử đa dạng về chủ đề và độ khó.
- Phương pháp đánh giá:
- Đánh giá định lượng:

  + Đo lường Answer Relevancy (Độ liên quan của câu trả lời với câu hỏi) thông qua đánh giá thủ công.

  + Đo lường Faithfulness (Câu trả lời có dựa trên ngữ cảnh truy xuất không) thông qua đánh giá thủ công (nếu khả thi trong thời gian).

  + Đo lường Tốc độ phản hồi trung bình của hệ thống..
- Đánh giá định tính: Thu thập phản hồi từ nhóm nhỏ người dùng thử nghiệm (sinh viên, khoảng 5-10 người) thông qua khảo sát online đơn giản về: tính dễ sử dụng, tính hữu ích, độ chính xác cảm nhận, tốc độ và góp ý cải thiện.
- Thực hiện đánh giá: Cho người dùng tương tác với chatbot, ghi nhận kết quả định lượng và thu thập phản hồi định tính.

- Phân tích kết quả: Tổng hợp, phân tích dữ liệu từ các chỉ số đo lường và kết quả khảo sát. So sánh với mục tiêu đề ra.
- Thảo luận: Bàn luận về kết quả đạt được, ưu điểm, nhược điểm của hệ thống RAG với vector retrieval (Qdrant), và các vấn đề gặp phải. Nêu rõ hạn chế của việc đánh giá đơn giản hóa.

- Chương 6: Kết luận và hướng phát triển:

- Tóm tắt các kết quả chính đã đạt được của đề tài.
- Đóng góp của đề tài.
- Hạn chế của hệ thống và của nghiên cứu (bao gồm hạn chế từ việc đánh giá đơn giản hóa).
- Đề xuất các hướng phát triển trong tương lai:

  + Cải thiện độ chính xác (tinh chỉnh embedding, prompt, LLM).

  + Cải tiến chiến lược phân đoạn (Chunking): Nghiên cứu semantic chunking.

  + Mở rộng tính năng chatbot (tạo câu hỏi ôn tập, tóm tắt, quản lý lịch sử hội thoại chi tiết hơn).

  + Tối ưu hóa hiệu năng và mở rộng cơ sở tri thức.

  + Thực hiện đánh giá toàn diện hơn với các chỉ số RAG đầy đủ.

  + Khám phá các vector database khác hoặc self-hosting Qdrant.

3. Phương pháp thực hiện:

- Nghiên cứu lý thuyết: Đọc và tổng hợp kiến thức về RAG, LLMs, NLP, vector databases (Qdrant), và các công nghệ liên quan (Langchain, Gemini API,...). Nghiên cứu tài liệu môn Cơ sở dữ liệu.
- Phân tích và thiết kế hệ thống: Áp dụng các phương pháp luận phát triển phần mềm, mô hình hóa hệ thống.
- Lập trình và phát triển: Sử dụng Python, FastAPI, Langchain, Qdrant, Gemini API,...
- Thử nghiệm và kiểm thử: Unit testing, integration testing, system testing cho các thành phần cốt lõi.
- Đánh giá thực nghiệm: Xây dựng quy trình đánh giá, thu thập dữ liệu thông qua kiểm tra thủ công (relevancy, faithfulness) và khảo sát người dùng, sử dụng các chỉ số RAG cơ bản và tốc độ phản hồi.

4. Bố cục đề tài:

Đề tài dự kiến được chia thành 6 chương như đã trình bày chi tiết trong Mục 2 (Nội dung thực hiện).

- Chương 1: Tổng quan
- Chương 2: Cơ sở lý thuyết
- Chương 3: Phân tích và thiết kế hệ thống
- Chương 4: Triển khai hệ thống
- Chương 5: Đánh giá và kết quả
- Chương 6: Kết luận và hướng phát triển

5. Tài liệu tham khảo:

- [1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel và D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Advances in Neural Information Processing Systems, vol. 33, Dec. 2020. [Online]. Available: https://arxiv.org/pdf/2005.11401.pdf. [Đã truy cập: 05-05-2025].
- [2] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang và H. Wang, "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv:2312.10997, Dec. 2023. [Online]. Available: https://arxiv.org/pdf/2312.10997.pdf. [Đã truy cập: 05-05-2025].
- [3] A. Zhang, B. Singh và C. Kumar, "A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions," arXiv:2410.12837, Oct. 2024. [Online]. Available: https://arxiv.org/pdf/2410.12837.pdf. [Đã truy cập: 05-05-2025].
- [4] D. Lin, "Revolutionizing Retrieval-Augmented Generation with Enhanced PDF Structure Recognition," arXiv:2401.12599, Jan. 2024. [Online]. Available: https://arxiv.org/pdf/2401.12599.pdf. [Đã truy cập: 05-05-2025].
- [5] W. Yeo, K. Kim, S. Jeong, J. Baek và S. Hwang, "UniversalRAG: Retrieval-Augmented Generation over Multiple Corpora with Diverse Modalities and Granularities," arXiv:2504.20734, Apr. 2025. [Online]. Available: https://arxiv.org/pdf/2504.20734.pdf. [Đã truy cập: 05-05-2025].
- [6] A. A. Khan, M. T. Hasan, K. K. Kemell, J. Rasku và P. Abrahamsson, "Developing Retrieval Augmented Generation (RAG) based LLM Systems from PDFs: An Experience Report," arXiv:2410.15944, Oct. 2024. [Online]. Available: https://arxiv.org/pdf/2410.15944.pdf. [Đã truy cập: 05-05-2025].

- [7] L. Thanh Hương, "Xử lý ngôn ngữ tự nhiên (Natural Language Processing)," Slide Lecture, Viện CNTT&TT – ĐHBK Hà Nội, 2022. [Online]. Available: https://users.soict.hust.edu.vn/huonglt/UNLP/1_introduction.pdf. [Đã truy cập: 05-05-2025].
- [8] N. Q. Đức, L. H. Sơn, N. D. Nhân, N. D. N. Minh, L. T. Hương và D. V. Sang, "Towards Comprehensive Vietnamese Retrieval-Augmented Generation and Large Language Models," arXiv:2403.01616, Mar. 2024. [Online]. Available: https://arxiv.org/pdf/2403.01616.pdf. [Đã truy cập: 05-05-2025].
- [9] Z. Jiang, F. F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan và G. Neubig, "Active Retrieval Augmented Generation," in Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP), Abu Dhabi, Dec. 2023, pp. 7969–7992. [Online]. Available: https://aclanthology.org/2023.emnlp-main.495.pdf. [Đã truy cập: 05-05-2025].
- [10] T. N. B. Nguyen, V. D. Thao, T. P. Quoc và T. T. V. Tran, "Vietnamese Legal Information Retrieval in Question-Answering System," arXiv:2409.13699, Sep. 2024. [Online]. Available: https://arxiv.org/pdf/2409.13699.pdf. [Đã truy cập: 05-05-2025].

6. Kế hoạch thực hiện đề tài:

Tuần	Từ ngày - đến ngày	Công việc thực hiện	Ghi chú
1	Từ ngày 07/4/2025 đến ngày 13/4/2025	- Nghiên cứu sâu lại lý thuyết RAG, Embedding (Sentence Transformers), Vector DB (Qdrant), LLM API (Gemini). - Hoàn thiện kế hoạch chi tiết, xác định rõ phạm vi cuối cùng. - Tìm hiểu các tiêu chuẩn trình bày khóa luận. - Thiết lập môi trường phát triển cơ bản (Python, Git, VS Code).	Nguyễn Đại Hoàng Phúc
2	Từ ngày 14/4/2025 đến ngày 20/4/2025	- Thu thập tài liệu môn Cơ sở dữ liệu (giáo trình, slide, bài tập,...). - Bắt đầu tiền xử lý dữ liệu: Trích xuất text từ PDF (PyPDF/Tesseract), xử lý file text, HTML.	Nguyễn Đại Hoàng Phúc

		- Nghiên cứu và lựa chọn chiến lược phân đoạn (Chunking Strategy). - Đăng ký và cấu hình API keys (Qdrant, Google Gemini).	
3	Từ ngày 21/4/2025 đến ngày 27/4/2025	- Hoàn thiện việc tiền xử lý và làm sạch dữ liệu. - Xây dựng module Data Ingestion với chức năng đọc, xử lý và phân đoạn tài liệu theo chiến lược đã chọn. - Kiểm thử module Ingestion với các loại tài liệu khác nhau. - Tạo cấu trúc thư mục dự án chuẩn.	Nguyễn Đại Hoàng Phúc
4	Từ ngày 28/4/2025 đến ngày 04/5/2025	- Nghiên cứu và lựa chọn mô hình Sentence Transformer phù hợp. - Xây dựng module Embedding: Tạo vector embeddings cho các đoạn tài liệu. - Tích hợp Qdrant: Cấu hình collection, tải embeddings và metadata lên Qdrant. - Kiểm thử quá trình tạo và lưu trữ embeddings.	Nguyễn Đại Hoàng Phúc
5	Từ ngày 05/5/2025 đến ngày 11/5/2025	- Xây dựng module Vector Retrieval: Tạo embedding cho câu hỏi người dùng, thực hiện tìm kiếm trên Qdrant. - Lấy ra các đoạn ngữ cảnh (context) liên quan nhất. - Kiểm thử hiệu quả của việc truy xuất ngữ cảnh.	Nguyễn Đại Hoàng Phúc

6	Từ ngày 12/5/2025 đến ngày 18/5/2025	- Xây dựng module Generation: Tích hợp Google Gemini API (sử dụng Langchain). - Thiết kế và thử nghiệm các prompt template để kết hợp câu hỏi và ngữ cảnh truy xuất được. - Sinh câu trả lời từ LLM. - Xây dựng module Postprocessing cơ bản (định dạng câu trả lời). - Kiểm thử luồng RAG hoàn chỉnh đầu tiên.	Nguyễn Đại Hoàng Phúc
7	Từ ngày 19/5/2025 đến ngày 25/5/2025	- Thiết kế kiến trúc API (các endpoints, request/response models). - Xây dựng Backend API sử dụng FastAPI để đóng gói luồng RAG. - Triển khai các endpoint chính (ví dụ: /chat). - Viết các unit test cơ bản cho API.	Nguyễn Đại Hoàng Phúc
8	Từ ngày 26/5/2025 đến ngày 01/6/2025	- Thiết kế giao diện người dùng (UI) web đơn giản. - Xây dựng Frontend sử dụng HTML, CSS, JavaScript. - Tích hợp Frontend với Backend API để gửi câu hỏi và hiển thị câu trả lời. - Kiểm thử luồng tương tác người dùng.	Nguyễn Đại Hoàng Phúc
9	Từ ngày 02/6/2025 đến ngày 08/6/2025	- Tích hợp toàn bộ hệ thống. - Triển khai với (docker nếu có thời gian) - Đánh giá hệ thống. - Phân tích kết quả, xác định ưu/nhược điểm.	Nguyễn Đại Hoàng Phúc

		- Thực hiện các tinh chỉnh nhỏ và sửa lỗi dựa trên kết quả đánh giá.	
10	Từ ngày 09/6/2025 đến ngày 15/6/2025	- Hoàn thiện bản thảo cuối cùng của khóa luận tốt nghiệp (viết báo cáo, bổ sung kết quả đánh giá, thảo luận, kết luận, hướng phát triển). - Định dạng tài liệu theo đúng quy định. - Chuẩn bị slide thuyết trình và demo sản phẩm. - Duyệt lại toàn bộ khóa luận, kiểm tra lỗi chính tả, định dạng. - Hoàn tất nộp khóa luận tốt nghiệp.	Nguyễn Đại Hoàng Phúc

Trà Vinh, ngày tháng 4 năm 2025

GIẢNG VIÊN HƯỚNG DẪN **SINH VIÊN THỰC HIỆN**

TS. Nguyễn Bảo Ân **Nguyễn Đại Hoàng Phúc**