RAG-based Q&A System: Hệ thống hỏi đáp video bài giảng dựa trên RAG

Nguyen Phuong Quyen, Nguyen Lam Bao Phuc, Ngo Pham Phuong Uyen
University of Information Technology, VNU-HCM

***Tóm tắt nội dung*—Báo cáo này trình bày quy trình thiết kế và hiện thực hóa hệ thống hỏi đáp (Q&A) tự động dành cho dữ liệu video bài giảng, ứng dụng kỹ thuật Retrieval-Augmented Generation (RAG) kết hợp với luồng xử lý tác vụ thông minh (Agentic Workflow). Hệ thống tích hợp các kỹ thuật xử lý dữ liệu tiên tiến bao gồm phân đoạn đệ quy (Recursive Chunking) kết hợp phân đoạn ngữ nghĩa (Semantic Chunking) để tối ưu hóa đầu vào. Quy trình truy hồi (Retrieval) sử dụng chiến lược tìm kiếm hỗn hợp (Hybrid Search) giữa BM25 và Vector Search, được tinh chỉnh bởi thuật toán Maximal Marginal Relevance (MMR) và mô hình tái xếp hạng Cross-Encoder để nâng cao độ chính xác. Kết quả thực nghiệm cho thấy hệ thống hoạt động ổn định, có khả năng trả lời chính xác các câu hỏi chuyên ngành và cung cấp trích dẫn nguồn (timestamp) minh bạch, đáp ứng tốt nhu cầu hỗ trợ tự học của sinh viên. Mã nguồn (Source code) của dự án đã được tải lên tại: `https://github.com/BPhucKHMT/Rag_QABot.git`**

***Index Terms*—Agentic RAG, LangGraph, Recursive Chunking, Semantic Chunking, Hybrid Search, Vector Search, Cross-Encoder**

I. Giới thiệu

Trong những năm gần đây, hình thức học trực tuyến và giảng dạy thông qua video đã trở nên phổ biến và đóng vai trò quan trọng trong giáo dục. Tuy nhiên, do đặc tính **phi cấu trúc và tuyến tính theo thời gian** của nội dung video, người học gặp nhiều khó khăn khi tìm kiếm, định vị và truy cập nhanh chóng đến một nội dung hoặc một khái niệm cụ thể nằm sâu trong một bài giảng kéo dài. Người học thường phải tốn nhiều thời gian tua lại hoặc dò tìm thủ công, làm giảm hiệu quả của quá trình tự học và ôn tập. Do đó, nhu cầu về một hệ thống hỗ trợ tìm kiếm thông minh, có khả năng "hiểu" nội dung video và chỉ dẫn chính xác đến vị trí kiến thức cần tìm là vô cùng cần thiết.

Để giải quyết bài toán trên, báo cáo này đề xuất một hệ thống hỏi đáp tự động cho video bài giảng dựa trên Retrieval-Augmented Generation (RAG) [1]. RAG là một kiến trúc kết hợp giữa mô hình truy hồi thông tin (retriever) và mô hình sinh ngôn ngữ (generator), trong đó các đoạn văn bản liên quan được truy xuất từ nguồn tri thức bên ngoài và cung cấp làm ngữ cảnh cho mô hình sinh nhằm tạo ra câu trả lời chính xác và có căn cứ. Dựa trên kiến trúc này, hệ thống khai thác các kỹ thuật xử lý ngôn ngữ tự nhiên và truy hồi thông tin để chuyển đổi dữ liệu video bài giảng phi cấu trúc thành dạng tri thức có thể truy vấn, từ đó hỗ trợ người học tiếp cận nhanh chóng và hiệu quả các nội dung kiến thức cần tìm.

Hệ thống được xây dựng với các đặc điểm kỹ thuật chính sau:

- **Quy trình xử lý dữ liệu linh hoạt:** Hệ thống tích hợp cơ chế thu thập dữ liệu linh hoạt, ưu tiên trích xuất từ phụ đề gốc và có cơ chế dự phòng để tự động chuyển đổi âm thanh sang văn bản (ASR) khi cần thiết. Dữ liệu sau đó được chuẩn hóa và xử lý bằng các chiến lược phân đoạn (Chunking) nâng cao nhằm chuyển đổi dữ liệu thô thành dạng tri thức có khả năng truy vấn (queryable knowledge representation)
- **Cơ chế truy hồi và tái xếp hạng:** Để đảm bảo độ chính xác cao nhất, hệ thống áp dụng chiến lược Hybrid Search, kết hợp ưu điểm của tìm kiếm từ khóa và tìm kiếm vector. Kết quả truy hồi lại được tinh chỉnh qua lớp Cross-Encoder nhằm lọc nhiễu và định vị chính xác phân đoạn chứa câu trả lời
- **Điều phối thông minh:** Sử dụng kiến trúc *Agentic Workflow* (LangGraph) để điều phối luồng hội thoại, cho phép hệ thống phản hồi linh hoạt và cung cấp trích dẫn thời gian thực (timestamps), giúp người học truy cập ngay lập tức đến nội dung cần thiết trong video.

II. Phương pháp thực hiện

Hệ thống được thiết kế theo kiến trúc đa tầng (Multi-layered Architecture), phân tách rõ ràng giữa quy trình xử lý dữ liệu ngoại tuyến (Offline Pipeline) và luồng xử lý truy vấn thời gian thực (Online Inference Pipeline). Cách tiếp cận này đảm bảo hiệu năng cao cho hệ thống khi mở rộng quy mô dữ liệu.

A. Tổng quan hệ thống

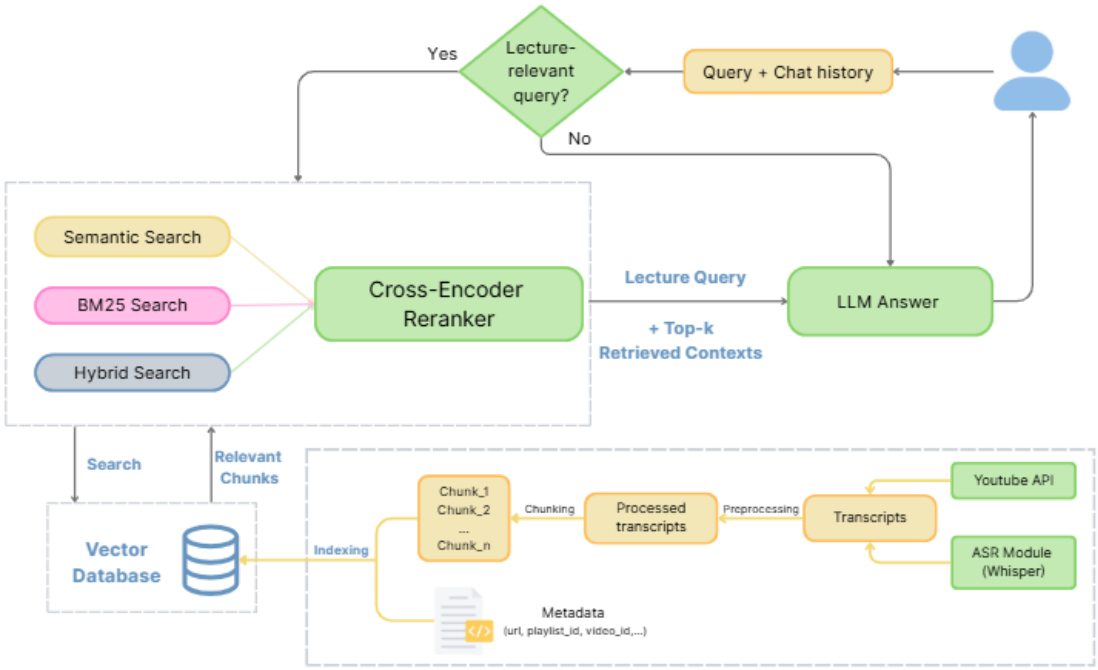
Kiến trúc hệ thống bao gồm ba phân hệ chính tương tác chặt chẽ với nhau (Hình 1):

- **Phân hệ dữ liệu (Data engine):** Thu thập metadata, trích xuất transcript và vector hóa nội dung video.
- **Phân hệ truy hồi (Retrieval Engine):** Thực hiện tìm kiếm hỗn hợp (Hybrid Search) và tái xếp hạng (ReRanking).
- **Phân hệ sinh nội dung và điều phối (Generation & Orchestration):** Sử dụng kiến trúc Agentic Workflow để điều phối luồng và sinh câu trả lời kèm trích dẫn.

B. Chi tiết các phân hệ thực thi

1. Phân hệ xử lý dữ liệu (Data Engine): Giai đoạn này chuyển đổi dữ liệu phi cấu trúc thành tri thức có thể truy vấn thông qua các bước:

- *Thu thập dữ liệu:* Sử dụng YouTube Data API để lấy metadata và phụ đề gốc. Với video thiếu phụ đề, mô hình `whisper-medium` được sử dụng để chuyển đổi âm thanh sang văn bản.
- *Tiền xử lý:* Áp dụng cơ chế Denoising thông qua mô hình Gemini để sửa lỗi chính tả và loại bỏ các câu thoại gây nhiễu (lời chào, quảng cáo).



Hình 1. Sơ đồ tổng quan về kiến trúc hệ thống và luồng xử lý dữ liệu

- *Phân đoạn dữ liệu (Chunking):* Kết hợp 2 phương pháp là *Recursive Chunking* (Phương pháp truy hồi) và *Semantic Chunking* (Phân đoạn ngữ nghĩa) bằng `text-embedding-3-large` để bảo toàn ngữ cảnh.

2. Phân hệ Truy hồi và Xếp hạng (Retrieval & Ranking): Về chiến lược, hệ thống sử dụng chiến lược tìm kiếm đa tầng để tối ưu độ chính xác và độ phủ:

- *Tìm kiếm hỗn hợp (Hybrid Search):* Kết hợp tuyến tính giữa Keyword Search (BM25) và Vector Search (mô hình `BAAI/bge-m3`).
- *Tái xếp hạng (Reranking):* Top các kết quả được đưa qua mô hình `bge-reranker-base` để đánh giá trực tiếp cặp câu (Query, Document).

Về cơ sở toán học, hệ thống áp dụng thuật toán BM25 và MMR để tối ưu hóa điểm số:

$$\text{Score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D)+k_1\cdot\left(1-b+b\cdot\frac{|D|}{\text{avgdl}}\right)} \quad (1)$$

Để tránh trùng lặp nội dung, thuật toán MMR được áp dụng để cân bằng giữa độ liên quan (Sim_1) và tính đa dạng (Sim_2):

$$\text{MMR} = \max_{D_i \in R \setminus S} \left[\lambda \cdot \text{Sim}_1(D_i,Q) - (1-\lambda) \max_{D_j \in S} \text{Sim}_2(D_i,D_j) \right] \quad (2)$$

3. Phân hệ Sinh nội dung và Điều phối thông minh: Việc kiểm soát luồng hội thoại được thực hiện qua kiến trúc **LangGraph** với Router Agent để định tuyến câu hỏi.

Để giảm thiểu ảo giác (Hallucination), hệ thống áp dụng kỹ thuật *Contextual Grounding* thông qua một System Prompt nghiêm ngặt bao gồm:

- **Role Definition**: Thiết lập vai trò trợ lý học tập chuyên sâu
- **Negative Constraints**: Các quy tắc cấm như: "Không được tự suy luận thông tin ngoài video", "Nếu không có thông tin trong ngữ cảnh thì trả lời không biết".
- **Citation Protocol** (Yêu cầu thẻ tham chiếu `[index]` tương ứng). Kết quả cuối cùng được ánh xạ ngược (reverse mapping) để tạo ra các liên kết sâu (deep links) chứa timestamp chính xác (ví dụ: `&t=120s`).

C. Hạ tầng Kỹ thuật (Tech Stack)

1. Backend (Hệ thống phía Server): Tầng xử lý trung tâm được phát triển bằng ngôn ngữ Python, sử dụng các framework sau:

- **FastAPI:** Framework web hiện đại, hiệu năng cao được sử dụng để xây dựng API Server. FastAPI hỗ trợ xử lý bất đồng bộ (async/await), giúp hệ thống có thể phục vụ nhiều yêu cầu đồng thời mà không bị chặn (non-blocking).
- **LangChain & LangGraph:** Đóng vai trò là bộ khung (framework) để điều phối các mô hình ngôn ngữ lớn (LLM). LangGraph được sử dụng đặc biệt để xây dựng đồ thị trạng thái (State Graph) cho Agent, quản lý logic định tuyến giữa việc trả lời trực tiếp và truy hồi thông tin.
- **Motor:** Driver MongoDB bất đồng bộ (Async driver), cho phép Backend thực hiện các thao tác đọc/ghi lịch sử chat vào cơ sở dữ liệu mà không làm gián đoạn luồng xử lý chính.

2. Frontend (Giao diện người dùng): Giao diện tương tác được xây dựng bằng **Streamlit**, một framework tối ưu cho các ứng dụng Khoa học Dữ liệu:

- **Quản lý phiên (Session Management):** Sử dụng `st.session_state` để lưu trữ ngữ cảnh hội thoại và trạng thái của người dùng trong suốt quá trình tương tác.
- **Render trích dẫn:** Streamlit được tùy biến để render mã HTML/Markdown, cho phép hiển thị các liên kết video (Deep links) có thể nhấp được, đưa người dùng đến chính xác mốc thời gian (timestamp) trong video bài giảng.

3. Lưu trữ (Database): Hệ thống sử dụng mô hình lưu trữ lai (Hybrid Storage):

- **ChromaDB (Vector Database):** Cơ sở dữ liệu vector chuyên dụng để lưu trữ các embeddings (biểu diễn vector) của phân đoạn văn bản. ChromaDB sử dụng thuật toán HNSW để tối ưu hóa tốc độ tìm kiếm tương đồng.
- **MongoDB (NoSQL Database):** Lưu trữ toàn bộ lịch sử hội thoại, metadata của người dùng và logs hệ thống. Cấu trúc linh hoạt của JSON/BSON phù hợp để lưu trữ các đoạn hội thoại có độ dài thay đổi.

III. Dữ liệu và độ đo đánh giá

Để kiểm chứng hiệu quả của hệ thống, chúng tôi tiến hành đánh giá dựa trên khung kiểm thử tự động chuyên biệt cho RAG, tập trung vào khả năng truy hồi chính xác và chất lượng nội dung sinh ra.

A. Thiết lập đánh giá (Experimental Setup)

Quy trình đánh giá được thực hiện trên tập dữ liệu và công cụ sau:

- **Bộ dữ liệu kiểm thử (Dataset):** Chúng tôi xây dựng một tập dữ liệu kiểm thử chuyên biệt (Test Set) bao gồm các cặp câu hỏi - câu trả lời chuẩn (ground-truth answers), được trích xuất và gán nhãn thủ công từ các video bài giảng về Machine Learning và Deep Learning. Các câu hỏi được thiết kế đa dạng, bao gồm định nghĩa khái niệm, giải thích thuật toán và so sánh các phương pháp, nhằm phản ánh các kịch bản truy vấn thực tế trong hệ thống hỏi–đáp dựa trên video.
- **Công cụ đánh giá (Evaluation Framework):** Hệ thống được đánh giá bằng thư viện **RAGAS** (Retrieval-Augmented Generation Assessment) [2]. Đây là một framework tiêu chuẩn sử dụng LLM làm bộ đánh giá (LLM-as-a-judge) để đo lường chất lượng pipeline thông qua bộ chỉ số *RAG Triad*, giúp giảm thiểu sự phụ thuộc vào việc đánh giá thủ công từ con người.

B. Các chỉ số đo lường (Metrics)

Hiệu năng của hệ thống được đo lường thông qua bốn chỉ số cốt lõi được đề xuất trong RAGAS:

1) **Faithfulness (Độ trung thực):** Đo lường mức độ mà câu trả lời được sinh ra có thể được suy ra hoàn toàn từ các ngữ cảnh đã truy hồi. Chỉ số này phản ánh mức độ "ảo giác" (hallucination) của mô hình, trong đó điểm số cao cho thấy câu trả lời tuân thủ chặt chẽ thông tin có trong video và không đưa ra các khẳng định không được hỗ trợ bởi ngữ cảnh.
2) **Answer Relevancy (Độ liên quan của câu trả lời):** Đánh giá mức độ trực tiếp và phù hợp của câu trả lời đối với câu hỏi đầu vào. Chỉ số này phạt các câu trả lời lan man, lạc đề hoặc cung cấp thông tin không cần thiết, ngay cả khi các thông tin đó là đúng về mặt ngữ cảnh.
3) **Context Precision (Độ chính xác ngữ cảnh):** Đo lường chất lượng của bộ truy hồi bằng cách đánh giá tỷ lệ các đoạn ngữ cảnh (chunks) thực sự liên quan đến câu hỏi trong tập kết quả truy hồi top-k. Chỉ số này phản ánh khả năng loại bỏ nhiễu của retriever, và được kỳ vọng sẽ được cải thiện thông qua việc áp dụng cơ chế reranking bằng Cross-Encoder so với tìm kiếm vector thuần túy.
4) **Context Recall (Độ bao phủ ngữ cảnh):** Đo lường mức độ mà các ngữ cảnh được truy xuất bao phủ đầy đủ thông tin cần thiết để suy ra câu trả lời chuẩn (ground-truth answer). Context Recall cao cho thấy bộ truy hồi có khả năng thu thập đầy đủ các thông tin quan trọng liên quan đến câu hỏi, hạn chế việc bỏ sót nội dung cần thiết trong quá trình sinh câu trả lời.

IV. Kết quả đánh giá

Quá trình đánh giá được thực hiện bằng cách kết hợp và so sánh nhiều cấu hình khác nhau tại các giai đoạn chính của hệ thống, bao gồm *Phân đoạn văn bản (chunking)*, *Truy hồi thông tin (Search and Retrieval)* và *Sinh câu trả lời* sử dụng mô hình Gemini. Mục tiêu của thí nghiệm là xác định cấu hình tối ưu cho toàn bộ pipeline hỏi đáp dựa trên RAG.

Kết quả đánh giá trên tập dữ liệu kiểm thử cho thấy hiệu năng của hệ thống được cải thiện đáng kể khi áp dụng chiến lược *Hybrid Search*, đặc biệt trong việc nâng cao chất lượng ngữ cảnh được truy hồi.

Bảng I
KẾT QUẢ ĐÁNH GIÁ HIỆU NĂNG HỆ THỐNG TRÊN TẬP DỮ LIỆU KIỂM THỬ

Config	Faithfulness	Answer Relevancy	Context Precision	Context Recall
Semantic_Semantic	0.90	0.80	0.89	0.90
Semantic_BM25	0.80	0.68	0.80	0.83
Semantic_Hybrid	**0.93**	**0.94**	**1.00**	**0.90**
Recursive_Semantic	1.00	0.73	0.80	0.90
Recursive_BM25	0.96	0.82	0.90	0.90
Recursive_Hybrid	**0.87**	**0.83**	**0.90**	**0.90**

Dựa trên các kết quả thu được, có thể nhận thấy rằng ở giai đoạn *Phân đoạn văn bản*, phương pháp *Semantic Chunking* nhìn chung mang lại hiệu quả vượt trội so với *Recursive Chunking*, thể hiện qua các chỉ số *Answer Relevancy* và *Context Precision*.

Ở giai đoạn *Truy hồi thông tin*, chiến lược *Hybrid Search* cho thấy hiệu năng ổn định và vượt trội hơn so với các phương pháp tìm kiếm đơn lẻ như *Vector Search (Semantic)* hoặc *Keyword Search (BM25)*. Sự cải thiện này được phản ánh rõ ràng thông qua hai chỉ số quan trọng là *Context Precision* và *Context Recall*, cho thấy Hybrid Search vừa đảm bảo độ chính xác, vừa duy trì độ bao phủ ngữ cảnh cần thiết cho mô hình sinh.

Bên cạnh kết quả định lượng, đánh giá định tính cho thấy hệ thống có khả năng định vị chính xác các mốc thời gian (timestamp) tương ứng với nội dung trả lời, ngay cả đối với các truy vấn phức tạp. Điều này giúp người dùng truy cập trực tiếp đến đoạn kiến thức liên quan trong video bài giảng, từ đó tiết kiệm đáng kể thời gian tra cứu so với phương pháp tìm kiếm thủ công.

V. KẾT LUẬN (CONCLUSION)

A. Tổng kết đóng góp

Đồ án đã thiết kế và hiện thực hóa thành công một hệ thống hỏi đáp video bài giảng toàn diện (End-to-End). Các đóng góp chính bao gồm:

- Xây dựng quy trình xử lý dữ liệu mạnh mẽ với chiến lược phân đoạn lai (Hybrid Chunking) và làm sạch dữ liệu tự động.
- Tối ưu hóa độ chính xác truy hồi thông qua cơ chế tìm kiếm đa tầng (Hybrid Search + Cross-Encoder Reranking).
- Triển khai luồng tác vụ thông minh (Agentic Workflow) giúp hệ thống phản hồi linh hoạt và cung cấp trích dẫn nguồn minh bạch (Deep linking).
- Hệ thống hỗ trợ tự động hóa toàn trình, cho phép người dùng nhập danh sách phát (playlist) YouTube bất kỳ để thực thi quy trình thu thập dữ liệu, trích xuất văn bản và đánh chỉ mục (indexing) vào cơ sở dữ liệu vector.

B. Hạn chế hiện tại

Mặc dù đạt được những kết quả khả quan, hệ thống vẫn tồn tại một số hạn chế:

- **Ngôn ngữ:** Hiện tại hệ thống tập trung tối ưu cho tiếng Việt, khả năng xử lý các video đa ngôn ngữ hoặc chuyển ngữ (Code-switching) còn hạn chế.
- **Độ trễ (Latency):** Việc sử dụng mô hình Cross-Encoder để tái xếp hạng làm tăng thời gian phản hồi, đặc biệt khi số lượng tài liệu truy hồi lớn.
- **Phạm vi tri thức:** Hệ thống hiện chỉ giới hạn trong các môn học Khoa học Máy tính (ML/DL/Python) và chưa có cơ chế tự động cập nhật kiến thức thời gian thực khi có video mới.

C. Hướng phát triển

Trong tương lai, nhóm nghiên cứu đề xuất các hướng cải tiến sau:

- **Multimodal RAG:** Tích hợp khả năng xử lý đa phương thức để hiểu nội dung từ cả slide bài giảng (OCR) và hình ảnh minh họa, không chỉ dựa vào phụ đề.
- **Mở rộng Domain:** Tinh chỉnh pipeline để hỗ trợ các lĩnh vực khác và bổ sung hỗ trợ tiếng Anh toàn diện.
- **Tối ưu hóa tốc độ:** Nghiên cứu các kỹ thuật lượng tử hóa (Quantization) hoặc mô hình Reranking nhẹ hơn để giảm độ trễ hệ thống.

TÀI LIỆU

[1] P. Lewis, E. Perez, A. Piktus, *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[2] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, "Ragas: Automated evaluation of retrieval augmented generation," *arXiv preprint arXiv:2309.15217*, 2023.