BÁO CÁO
Assignment End-to-End -NLP-System Building

Triệu Vũ Hoàn - 22022654

Nguyễn Trường Huy - 22022509

1. Giới thiệu

Bài tập này tập trung vào việc xây dựng một hệ thống Hỏi Đáp (Question Answering - QA) hoàn chỉnh từ đầu đến cuối, sử dụng kiến trúc Retrieval Augmented Generation (RAG) theo mô tả của Lewis et al. (2021). Mục tiêu là phát triển một hệ thống có khả năng trả lời các câu hỏi thực tế bằng cách đầu tiên truy xuất các tài liệu liên quan từ một kho tri thức đã xây dựng, sau đó sử dụng các tài liệu này để sinh ra câu trả lời.

Hệ thống được thiết kế để xử lý các câu hỏi bằng tiếng Việt, với trọng tâm kiến thức ban đầu hướng về Đại học Quốc gia Hà Nội (VNU) và các đơn vị thành viên. Quá trình xây dựng bao gồm các giai đoạn chính:

(1) Thu thập và tiền xử lý dữ liệu từ các nguồn web công khai

(2) Phân đoạn văn bản thông minh (semantic chunking) để tối ưu hóa cho việc truy xuất

(3) Xây dựng kho vector sử dụng FAISS để lưu trữ và truy vấn hiệu quả các đoạn văn bản đã được embedding

(4) Phát triển một module truy xuất đa bước (multi-hop retrieval) cho phép hệ thống đào sâu hoặc mở rộng tìm kiếm thông tin

(5) Tích hợp với các Mô hình Ngôn ngữ Lớn (LLM) opensource mạnh mẽ thông qua OpenRouter API để sinh câu trả lời dựa trên ngữ cảnh đã truy xuất.

Các công nghệ và thư viện chính được sử dụng bao gồm Python, requests và BeautifulSoup4 cho thu thập dữ liệu, NLTK và sentence-transformers cho xử lý và embedding văn bản, FAISS cho kho vector, và các mô hình LLM như Qwen series cho các tác vụ lập kế hoạch truy vấn và sinh câu trả lời. Báo cáo này sẽ trình bày chi tiết về thiết kế hệ thống, quá trình tạo dữ liệu, các thử nghiệm mô hình, kết quả đạt được và những phân tích, đánh giá liên quan.

Link github: https://github.com/huynt119/RAG_VNU

---

2. Tạo Dữ liệu (Data Creation)

2.1. Biên soạn Kho Tri Thức (Knowledge Resource)

- Phương pháp biên soạn: Kho tri thức của hệ thống được xây dựng bằng cách thu thập tự động nội dung từ các trang web công khai. Điểm khởi đầu cho quá trình thu thập là trang chủ của Đại học Quốc gia Hà Nội (https://www.vnu.edu.vn/home/?C1885).
- Tiêu chí lựa chọn tài liệu: Tập trung vào các thông tin liên quan đến Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội và các trường/khoa thành viên. Danh sách các tên miền được ưu tiên thu thập bao gồm: uet.vnu.edu.vn, vnu.edu.vn, ueb.edu.vn, hus.vnu.edu.vn, ussh.vnu.edu.vn, ulis.vnu.edu.vn, education.vnu.edu.vn, vju.ac.vn, is.vnu.edu.vn, hsb.edu.vn, law.vnu.edu.vn cùng với dữ liệu từ các trang tuyensinh, wiki,...
- Hệ thống crawler (*crawl.py*) được thiết kế để duyệt qua các liên kết trong các tên miền này, tải về và lưu trữ nội dung văn bản. Cơ chế lưu trạng thái (*crawl_state.json*) và sử dụng đa luồng cũng được triển khai để tối ưu hóa quá trình thu thập.
- Tổng thu thập được dữ liệu của hơn 10 nghìn trang web

2.2. Trích xuất và Làm sạch Dữ liệu Thô

- Công cụ và quy trình trích xuất: Dữ liệu HTML từ các URL hợp lệ được tải về bằng thư viện requests. Sau đó, BeautifulSoup4 được sử dụng để phân tích cây HTML. Nội dung văn bản chính được trích xuất từ các thẻ HTML thường chứa nội dung như <article>, <main>, các class như .entry-content, .content, .noidung… cùng với các file PDF được xử lý bởi pypdf. Các thành phần không mong muốn như menu điều hướng (nav), footer, script, sidebar đã được loại bỏ trước khi trích xuất text để giảm nhiễu.
- Quy trình làm sạch:
    - Loại bỏ khoảng trắng thừa ở đầu và cuối mỗi dòng.
    - Loại bỏ các dòng hoàn toàn trống hoặc các dòng trống liên tiếp (nhiều hơn 2 lần xuống dòng).
    - Loại bỏ các chú thích thường gặp
    - Loại bỏ các dòng chỉ chứa các ký tự lặp lại không mang ý nghĩa
- Mục tiêu là chuẩn bị dữ liệu văn bản sạch sẽ nhất có thể cho các bước xử lý tiếp theo.

2.3. Chú thích Dữ liệu (Annotation)

- Dữ liệu được chú thích cho Kiểm thử và Huấn luyện:
    - Loại dữ liệu: Các cặp câu hỏi - câu trả lời (Question-Answer pairs) liên quan đến nội dung đã thu thập, tập trung vào VNU và các đơn vị thành viên.
    - Số lượng: 242 cặp QA đã được chú thích thủ công với 180 câu cho tập train và 62 câu cho tập kiểm thử (test set).

- Quy trình quyết định và chú thích:
    - Hai phương pháp chính được kết hợp để tạo tập dữ liệu câu hỏi-trả lời (Q-A) cho quá trình huấn luyện và đánh giá mô hình.
        - Thứ nhất, dựa vào các đoạn văn bản đã thu thập được ở bước trước từ các nguồn tài liệu của ĐHQGHN để trích xuất thông tin và xây dựng các cặp Q-A. Trong đó, các câu hỏi được tạo ra sao cho câu trả lời có thể được tìm thấy trực tiếp trong các đoạn văn bản này.
        - Thứ hai, tham khảo các câu hỏi mà người dùng thường quan tâm hoặc hay hỏi nhất về ĐHQGHN (ví dụ: thông tin tuyển sinh, chương trình đào tạo, lịch sử hình thành) để tạo thêm các cặp Q-A, nhằm tăng tính bao phủ và thực tế của tập dữ liệu. Các câu hỏi này được tạo thủ công bởi các thành viên trong nhóm.

- Ước tính Chất lượng Chú thích (Inter-Annotator Agreement - IAA):
    - Để đảm bảo chất lượng của tập dữ liệu, tiến hành đánh giá chủ quan và đánh giá sự đồng thuận giữa những người chú thích.
    - Về đánh giá chủ quan, mỗi thành viên trong nhóm (2 người) độc lập xem xét từng cặp Q-A và đánh giá theo các tiêu chí sau:
        - Tính đúng đắn: Câu trả lời có chính xác và đầy đủ so với ngữ cảnh của câu hỏi không?
        - Tính rõ ràng: Câu hỏi và câu trả lời có dễ hiểu và không mơ hồ không?
        - Tính liên quan: Câu hỏi có liên quan đến nội dung của các tài liệu nguồn không?
    - Mỗi cặp Q-A được gán nhãn một trong ba mức độ: "tốt" (đạt tất cả các tiêu chí), "trung bình" (đạt phần lớn các tiêu chí nhưng có một vài điểm cần cải thiện), hoặc "không tốt" (không đạt nhiều tiêu chí)
    - Sau khi đánh giá chủ quan, tiến hành đánh giá sự đồng thuận giữa những người chú thích (Inter-Annotator Agreement - IAA). Sử dụng hệ số Kappa của Cohen để đo lường mức độ đồng thuận này. Hệ số Kappa tính đến khả năng xảy ra sự đồng thuận ngẫu nhiên, do đó đưa ra đánh giá khách quan hơn về mức độ đồng thuận thực tế. Kết quả IAA là 0,846, cho thấy mức độ đồng thuận cao giữa hai người chú thích.

3. Chi tiết Hệ thống và Mô hình (System and Model Details)

3.1. Kiến trúc Tổng quan Hệ thống RAG

Hệ thống RAG được phát triển theo kiến trúc module hóa, bao gồm các thành phần chính sau:

1. Input (Câu hỏi): Người dùng nhập câu hỏi bằng tiếng Việt.
2. Module Truy xuất Đa bước (Multi-hop Retriever):
    - Câu hỏi ban đầu được sử dụng để truy xuất một tập các đoạn văn bản (chunks) liên quan từ Kho Vector.
    - Một LLM "Planner" (triển khai qua OpenRouter) phân tích câu hỏi gốc và ngữ cảnh đã truy xuất. Nếu thông tin chưa đủ, Planner sẽ sinh ra một câu hỏi phụ hoặc một truy vấn cụ thể hơn để thực hiện lượt truy xuất tiếp theo.
    - Quá trình này có thể lặp lại nhiều lần (multi-hop), tích lũy thêm ngữ cảnh sau mỗi hop, cho đến khi Planner xác định đã đủ thông tin hoặc đạt số hop tối đa.
3. Module Sinh Câu trả lời (Generator):
    - Toàn bộ ngữ cảnh thu thập được từ các bước truy xuất được tổng hợp lại.
    - Một LLM "Generator" (triển khai qua OpenRouter) nhận câu hỏi gốc và toàn bộ ngữ cảnh này để sinh ra câu trả lời cuối cùng bằng tiếng Việt.

3.2. Module Xử lý và Phân đoạn Văn bản (Text Processing and Chunking)

Quá trình xử lý văn bản bao gồm việc làm sạch dữ liệu (như mô tả ở Mục 2.2) và sau đó là phân đoạn văn bản thành các đơn vị nhỏ hơn (chunks). Trung tâm của việc tạo ra các biểu diễn ngữ nghĩa cho văn bản, phục vụ cả quá trình phân đoạn và truy xuất, là mô hình embedding chuyên biệt cho tiếng Việt.

- Mô hình Embedding sử dụng:
    - Nguồn gốc và Phát triển: AITeamVN/Vietnamese_Embedding là một mô hình embedding được phát triển bởi AITeamVN (Nguyễn Nho Trung, Nguyễn Nhật Quang). Điểm đặc biệt của mô hình này là nó được tinh chỉnh (fine-tuned) từ mô hình nền tảng BAAI/bge-m3, một mô hình embedding đa ngôn ngữ mạnh mẽ, với mục tiêu tăng cường đáng kể khả năng truy xuất (retrieval capabilities) spécifiquement cho ngôn ngữ tiếng Việt.
    - Quá trình Tinh chỉnh (Fine-tuning):
        - Dữ liệu: Mô hình được huấn luyện trên một tập dữ liệu lớn gồm khoảng 300,000 bộ ba (triplets) tiếng Việt. Mỗi bộ ba bao gồm một câu truy vấn (query), một tài liệu dương tính (positive document - liên quan đến query) và một tài liệu âm tính (negative document - không liên quan đến query). Phương pháp huấn luyện dựa trên triplets này (contrastive learning) giúp mô hình học cách kéo các biểu diễn của query và positive document lại gần nhau trong không gian vector, đồng thời đẩy xa biểu diễn của negative document.
        - Độ dài Chuỗi Tối đa (Maximum Sequence Length): Quá trình huấn luyện được thực hiện với độ dài chuỗi tối đa là 2048 tokens. Điều này cho phép mô hình xử lý hiệu quả các đoạn văn bản tương đối dài mà không bị cắt bớt thông tin quan trọng.
    - Thông số Kỹ thuật của Mô hình:
        - Loại Mô hình (Model Type): Sentence Transformer.
        - Mô hình Cơ sở (Base model): BAAI/bge-m3.
        - Độ dài Chuỗi Tối đa khi sử dụng: 2048 tokens
        - Chiều của Vector Đầu ra (Output Dimensionality): 1024 chiều.
        - Hàm Tương đồng Khuyến nghị (Similarity Function): Tích vô hướng (Dot product Similarity).
    - Trong Hệ thống RAG được phát triển:

        - Semantic Chunking AITeamVN/Vietnamese_Embedding được sử dụng để chuyển đổi từng câu trong tài liệu đã làm sạch thành các vector 1024 chiều. Độ tương đồng (sử dụng tích vô hướng hoặc cosine similarity sau chuẩn hóa) giữa các vector câu liên tiếp được tính toán. Nếu độ tương đồng này giảm xuống dưới ngưỡng SIMILARITY_THRESHOLD, một điểm ngắt chunk sẽ được xác định. Việc này giúp tạo ra các chunks mạch lạc về mặt ngữ nghĩa.
        - Xây dựng Kho Vector và Truy xuất
            - Tất cả các chunks văn bản đã được tạo ra từ bước semantic chunking sẽ được mã hóa thành các vector 1024 chiều sử dụng AITeamVN/Vietnamese_Embedding.
            - Các vector này sau đó được lưu trữ trong một FAISS index.
            - Vector câu hỏi sau đó được sử dụng để truy vấn FAISS index nhằm tìm ra k chunks có vector tương đồng nhất
    - Lý do lựa chọn
        - Chuyên biệt cho Tiếng Việt: Việc fine-tuning trên một lượng lớn dữ liệu tiếng Việt giúp mô hình nắm bắt tốt hơn các đặc điểm ngữ nghĩa và cấu trúc của ngôn ngữ, điều cần thiết cho việc truy xuất thông tin chính xác.
        - Hiệu năng Truy xuất Tốt: Kết quả đánh giá công khai cho thấy mô hình có khả năng truy xuất tốt, vượt trội so với mô hình nền tảng đa ngôn ngữ và cạnh tranh với các mô hình tiếng Việt khác.
        - Hỗ trợ Chuỗi Dài: Khả năng xử lý chuỗi lên đến 2048 tokens là một lợi thế khi làm việc với các tài liệu có thể chứa các câu hoặc đoạn văn dài.
- Phương pháp Phân đoạn (Semantic Chunking):
    - Dựa vào độ tương đồng ngữ nghĩa giữa các câu được tính toán từ vector embedding của AITeamVN/Vietnamese_Embedding.
    - Kích thước của mỗi chunk cũng được kiểm soát bằng số lượng token (sử dụng tokenizer cl100k_base từ tiktoken), với MAX_CHUNK_TOKENS và MIN_CHUNK_TOKENS được chọn qua grid search đánh giá trên tập train để đảm bảo chunk không quá dài hoặc quá ngắn.

3.3. Module Xây dựng Kho Vector và Truy xuất (Vector Store & Retrieval)

- Kho Vector: Sử dụng thư viện FAISS để xây dựng và lưu trữ index.
- Model Embedding: Cùng một model được sử dụng trong bước chunking . Các chunk văn bản được encode thành vector và nạp vào FAISS index.

3.4. Module Sinh Ngôn ngữ sử dụng Mô hình Ngôn ngữ Lớn (LLM) qua OpenRouter

Hệ thống sử dụng nền tảng OpenRouter để truy cập các Mô hình Ngôn ngữ Lớn (LLM) cho hai nhiệm vụ chính: lập kế hoạch truy vấn (Planner) và sinh câu trả lời cuối cùng (Generator). Mô hình LLM chủ đạo được lựa chọn cho các tác vụ này là một biến thể của dòng Qwen3.

- Nền tảng truy cập LLM: OpenRouter API
- Dòng Mô hình LLM sử dụng: Qwen3
    - Nhà phát triển: Alibaba Cloud.
    - Tổng quan về Qwen3: Qwen3 là một dòng mô hình ngôn ngữ lớn (LLM) tiên tiến được phát triển bởi Alibaba Cloud, đại diện cho những cải tiến đáng kể so với các thế hệ trước đó như Qwen2 và Qwen1.5. Các mô hình trong series Qwen3 được thiết kế để cung cấp hiệu năng hàng đầu trên một loạt các tác vụ xử lý ngôn ngữ tự nhiên và đa phương thức.
    - Biến thể cụ thể được sử dụng: Hệ thống RAG này tận dụng biến thể qwen/qwen3-30b-a3b thông qua OpenRouter. Mô hình bao gồm chế độ suy luận, đặt nó vào nhóm các LLM có năng lực mạnh mẽ. Mô hình được công khai đầy đủ mã nguồn trên HuggingFace.
    - Kiến trúc Mô hình:

- Decoder-Only Transformer: Giống như hầu hết các LLM hiện đại, Qwen3 có khả năng cao dựa trên kiến trúc Transformer chỉ sử dụng bộ giải mã (decoder-only).
- Tối ưu hóa Kiến trúc: Các mô hình Qwen3 có thể tích hợp các cải tiến kiến trúc phổ biến như:
    - Grouped Query Attention (GQA): Để tăng hiệu quả tính toán và giảm yêu cầu bộ nhớ trong quá trình suy luận, đặc biệt quan trọng cho các mô hình lớn.
    - SwiGLU Activation Function: Một hàm kích hoạt cải thiện hiệu năng so với các hàm truyền thống như ReLU.
    - Rope embeding
    - RMSNorm (Root Mean Square Layer Normalization): Để cải thiện sự ổn định trong quá trình huấn luyện.

- Cửa sổ Ngữ cảnh (Context Window): Một trong những điểm cải tiến quan trọng của Qwen3 có thể là cửa sổ ngữ cảnh được mở rộng đáng kể (ví dụ: 32K, 64K, hoặc thậm chí 128K tokens. Điều này rất có lợi cho các ứng dụng RAG, vì nó cho phép LLM xử lý một lượng lớn thông tin truy xuất cùng một lúc.
- Từ vựng (Vocabulary): Từ vựng được tối ưu hóa để hỗ trợ hiệu quả nhiều ngôn ngữ, bao gồm cả tiếng Việt, mặc dù trọng tâm huấn luyện ban đầu là tiếng Trung và tiếng Anh.

Models	Layers	Heads (Q / KV)	Tie Embedding	Context Length
Qwen3-0.6B	28	16 / 8	Yes	32K
Qwen3-1.7B	28	16 / 8	Yes	32K
Qwen3-4B	36	32 / 8	Yes	128K
Qwen3-8B	36	32 / 8	No	128K
Qwen3-14B	40	40 / 8	No	128K
Qwen3-32B	64	64 / 8	No	128K

Kiến trúc dòng model dense

Models	Layers	Heads (Q / KV)	# Experts (Total / Activated)	Context Length
Qwen3-30B-A3B	48	32 / 4	128 / 8	128K
Qwen3-235B-A22B	94	64 / 4	128 / 8	128K

Kiến trúc dòng model MOE

- Huấn luyện Mô hình:
    - Dữ liệu Tiền huấn luyện (Pre-training Data): Qwen3 được tiền huấn luyện trên một tập dữ liệu khổng lồ, có thể lên tới hàng nghìn tỷ (trillions) tokens. Dữ liệu này bao gồm một phổ rộng các nguồn: văn bản từ web, sách, mã nguồn, và có thể cả các dữ liệu chuyên ngành khác nhau. Sự đa dạng và quy mô của dữ liệu tiền huấn luyện là nền tảng cho kiến thức tổng quát rộng lớn và khả năng hiểu ngôn ngữ tự nhiên của mô hình.
    - Tinh chỉnh theo Hướng dẫn (Instruction Fine-Tuning - SFT): Sau giai đoạn tiền huấn luyện, các mô hình Qwen3 trải qua quá trình tinh chỉnh theo hướng dẫn một cách nghiêm ngặt. Quá trình này sử dụng các tập dữ liệu chất lượng cao, đa dạng, bao gồm các cặp (hướng dẫn, phản hồi) để dạy mô hình cách tuân theo chỉ dẫn của người dùng và thực hiện các tác vụ cụ thể.

        - Tinh chỉnh Đồng chỉnh (Alignment Tuning): Để tăng cường tính hữu ích, trung thực và vô hại, Qwen3 đã được tinh chỉnh đồng chỉnh bằng các kỹ thuật tiên tiến như Reinforcement Learning from Human Feedback (RLHF) hoặc Direct Preference Optimization (DPO). Những kỹ thuật này giúp mô hình học hỏi từ sở thích của con người và tạo ra các phản hồi phù hợp hơn với mong đợi.

    - Khả năng Nổi bật của Qwen3 (Liên quan đến ứng dụng RAG):
        - Tuân theo Hướng dẫn Phức tạp: Nhờ quá trình SFT và alignment, Qwen3 có khả năng hiểu và tuân theo các hướng dẫn chi tiết và phức tạp, điều này rất quan trọng cho cả vai trò Planner và Generator trong hệ thống RAG.
        - Khả năng Lập luận (Reasoning): Các mô hình lớn như Qwen3 30B thường thể hiện khả năng lập luận ở một mức độ nhất định, giúp Planner LLM "suy nghĩ" về việc liệu có cần thêm thông tin hay không và loại thông tin nào cần được truy xuất thêm.
        - Sinh Văn bản Mạch lạc và Sáng tạo: Qwen3 có thể sinh ra văn bản tự nhiên, mạch lạc và có tính sáng tạo (khi được yêu cầu), đồng thời vẫn bám sát ngữ cảnh được cung cấp.
        - Hỗ trợ Đa ngôn ngữ: Mặc dù trọng tâm có thể không phải là tiếng Việt, khả năng xử lý đa ngôn ngữ của Qwen3 cho phép nó hiểu và sinh văn bản tiếng Việt ở mức độ tốt, đặc biệt khi được cung cấp prompt và ngữ cảnh bằng tiếng Việt.
        - Hiểu biết Ngữ cảnh Dài: Với cửa sổ ngữ cảnh lớn, Qwen3 có thể duy trì và sử dụng thông tin từ các đoạn văn bản dài, rất quan trọng khi tổng hợp thông tin từ nhiều chunks truy xuất.
        - Khả năng sử dụng Công cụ (Tool Use) / Agent: Một số phiên bản Qwen có thể được huấn luyện để sử dụng các công cụ hoặc hoạt động như một agent. Mặc dù trong hệ thống RAG này, việc "sử dụng công cụ" của Planner LLM là sinh ra truy vấn cho retriever, nền tảng huấn luyện này có thể đóng góp vào khả năng đưa ra quyết định của nó.

- Vai trò của Qwen3 trong Hệ thống RAG
    - Planner LLM
        - Nhiệm vụ: Planner LLM nhận câu hỏi gốc của người dùng, ngữ cảnh đã truy xuất được từ các hop trước. Dựa trên khả năng hiểu hướng dẫn và lập luận, Planner LLM quyết định:
            - Nếu thông tin hiện tại đủ, báo hiệu dừng.
            - Nếu thiếu thông tin, sinh ra một câu hỏi/truy vấn mới, cụ thể và rõ ràng, để hướng dẫn lượt truy xuất tiếp theo.

        - Thiết kế Prompt : Prompt được thiết kế để khai thác khả năng phân tích và đánh giá của Qwen3, yêu cầu nó hoạt động như một "trợ lý nghiên cứu thông minh".
    - Generator LLM
        - Nhiệm vụ: Sau khi quá trình multi-hop kết thúc, Generator LLM nhận câu hỏi gốc và toàn bộ ngữ cảnh đã tích lũy. Nó sử dụng khả năng sinh văn bản mạch lạc và bám sát ngữ cảnh của Qwen3 để tạo ra câu trả lời cuối cùng.
        - Thiết kế Prompt: Prompt cho Generator rất nghiêm ngặt, yêu cầu Qwen3: chỉ sử dụng thông tin trong ngữ cảnh, trả lời bằng tiếng Việt, không suy diễn, và xử lý trường hợp không có thông tin một cách nhất quán.

3.5. Các Biến thể Hệ thống đã Thử nghiệm

- Biến thể 1: Closed-book LLM
    - Mô tả: Không dùng RAG.

- Biến thể 2: Single-hop RAG
    - Mô tả: Hệ thống chỉ thực hiện một lượt truy xuất duy nhất (max_hops=1). Toàn bộ ngữ cảnh từ lượt này được đưa trực tiếp cho Generator LLM để sinh câu trả lời.

- Biến thể 3: Multi-hop RAG (Hệ thống chính được đề xuất)
    - Mô tả: Hệ thống thực hiện nhiều lượt truy xuất (max_hops=3) với sự điều hướng của Planner LLM.
    - Lý do: Kỳ vọng rằng việc cho phép hệ thống "suy nghĩ" và tìm kiếm theo nhiều bước sẽ giúp giải quyết các câu hỏi phức tạp hơn, cần tổng hợp thông tin từ nhiều nguồn hoặc cần làm rõ ý trước khi trả lời. Điều này có thể cải thiện độ chính xác và tính đầy đủ của câu trả lời.

4. Kết quả Thử nghiệm (Experimental Results)

4.1. Thiết lập Thử nghiệm

- Tập dữ liệu đánh giá:
    - Tập kiểm thử (test set) do nhóm tự xây dựng và chú thích (mô tả ở Mục 2.3).
    - Tập kiểm thử "unseen" do giảng viên cung cấp.
- Các độ đo (Metrics): Theo hướng dẫn của bài tập và tham chiếu từ SQuAD, các độ đo token-based sau đã được sử dụng:
    - Exact Match (EM): Tỷ lệ phần trăm các câu trả lời dự đoán khớp hoàn toàn với một trong các câu trả lời tham chiếu.
    - F1-score: Trung bình điều hòa của Precision và Recall ở mức token, đo lường sự trùng khớp về từ ngữ giữa câu trả lời dự đoán và câu trả lời tham chiếu.
    - Answer Recall (AR): Tỷ lệ các token trong câu trả lời tham khảo được tìm thấy trong câu trả lời dự đoán.

4.2. Kết quả trên Tập Kiểm thử Tự Xây dựng

Biến thể Hệ thống	Exact Match (EM)	F1-score	Recall
Single-hop RAG (Baseline)	**0**	0.29	0.31
Multi-hop RAG (max_hops=3)	**0,1**	**0.4**	**0.45**

- Multi-hop RAG (max_hops=3) cho thấy hiệu năng tốt hơn rõ rệt về F1-score và Recall so với biến thể baseline Single-hop RAG. Điều này cho thấy mô hình multi-hop có khả năng kết hợp thông tin từ nhiều nguồn tốt hơn, dẫn đến độ bao phủ và độ chính xác cân bằng hơn.
- Hầu hết các câu trả lời của mô hình không khớp hoàn toàn với đáp án tham khảo. Nguyên nhân là do các đáp án tham khảo chỉ mang tính chất xác nhận thông tin và cùng một câu có thể được diễn đạt theo nhiều cách khác nhau
- Multi-hop có tiềm năng tốt hơn, đặc biệt cho các tác vụ yêu cầu tích hợp nhiều thông tin từ các đoạn khác nhau.

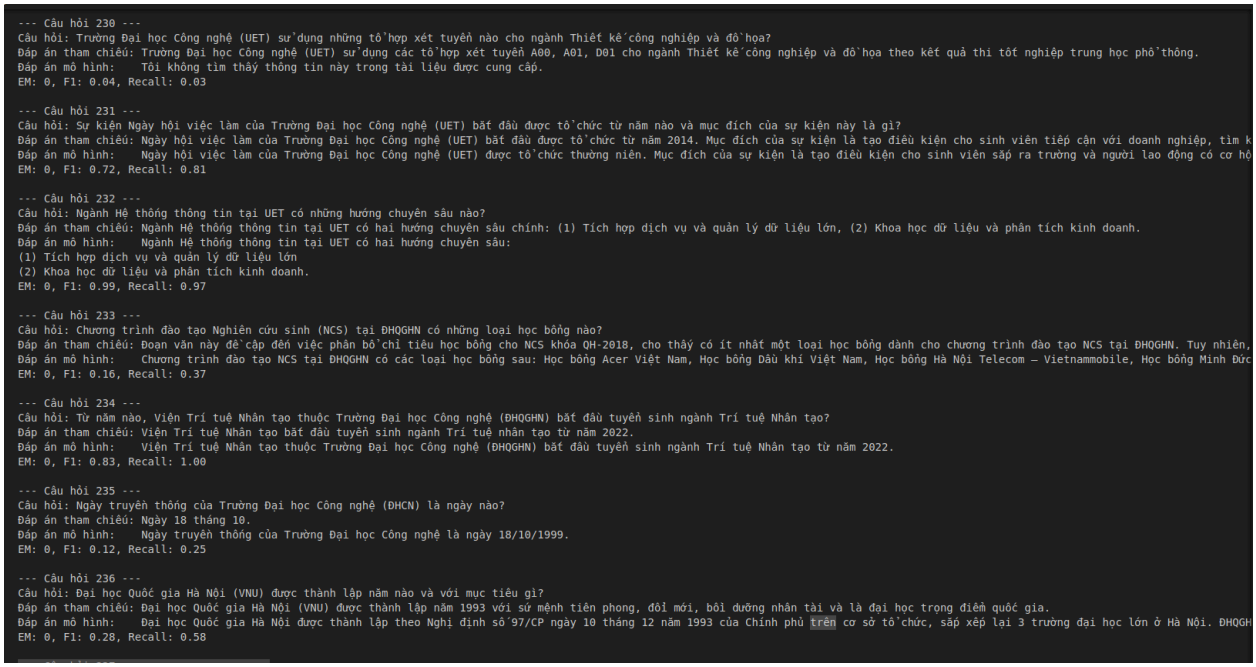
4.3. Kết quả trên Tập "Unseen Test Set"

- Hiện vẫn chưa có tập kiểm thử ẩn để thử nghiệm mô hình

---

5. Phân tích Kết quả (Analysis)

5.1. Một vài mẫu Q-A



5.1. So sánh RAG với Closed-book LLM

Phương pháp	Exact Match (EM)	F1-score	Recall
LLM Closed-book	**0**	0.11	0.13
Multi-hop RAG (max_hops=3)	**0**	**0.40**	**0.45**

- Thiết lập so sánh: "Để đánh giá hiệu quả của chiến lược retrieve-and-augment, các câu hỏi trong tập kiểm thử đã được chạy trực tiếp với mô hình Generator LLM (qwen/qwen3-30b-a3b) ở chế độ closed-book (không cung cấp bất kỳ ngữ cảnh truy xuất nào)."

- Kết quả so sánh:
  Phân tích: "Kết quả cho thấy rõ ràng ưu thế của hệ thống RAG. LLM closed-book gặp nhiều khó khăn trong việc trả lời chính xác các câu hỏi yêu cầu kiến thức chuyên biệt về VNU mà không có trong dữ liệu huấn

luyện của nó, dẫn đến tỷ lệ EM và F1 thấp đáng kể. Hệ thống RAG, nhờ khả năng truy xuất thông tin liên quan, đã cải thiện mạnh mẽ cả hai chỉ số, đồng thời giảm thiểu hiện tượng 'ảo giác' (hallucination) của LLM khi phải tự suy diễn thông tin."

5.2. Một vài so sánh với các biến thể

- Ví dụ 1: Multi-hop cải thiện so với Single-hop
    - Câu hỏi: "Phòng ban nào của Trường Đại học Công nghệ (UET) chịu trách nhiệm chính trong việc liên hệ và phát triển hợp tác?"
    - Trả lời (Single-hop RAG): ""Phòng Hợp tác phát triển chịu trách nhiệm chính trong việc liên hệ và phát triển hợp tác."
    - Trả lời (Multi-hop RAG, max_hops=3): "Phòng Khoa học Công nghệ và Hợp tác phát triển của Trường Đại học Công nghệ (UET) chịu trách nhiệm chính trong việc liên hệ và phát triển hợp tác."
    - Phân tích: "Trong trường hợp này, lượt truy xuất đầu tiên (single-hop) chưa tìm thấy thông tin chính xác. Planner LLM trong hệ thống multi-hop đã nhận thấy điều đó và tạo thêm truy vấn giúp truy xuất thêm ngữ cảnh cần thiết để hoàn thiện câu trả lời."

- Ví dụ 2: RAG so với Closed-book LLM
    - Câu hỏi: "Trường Đại học Công nghệ (ĐHQGHN) đã đạt được thành tích gì tại Olympic Toán học Sinh viên và Học sinh năm 2023?"
    - Trả lời (Multi-hop RAG): "Đội tuyển Toán học Trường Đại học Công nghệ (ĐHQGHN) đạt top 4 kỳ thi Olympic Toán học Sinh viên và Học sinh năm 2023 với thành tích 04 giải Nhất, 06 giải Nhì và 01 giải Ba."
    - Trả lời (Closed-book LLM): "Tôi không có thông tin về thành tích cụ thể tại kỳ thi Olympic Toán học Sinh viên và Học sinh năm 2023."
    - LLM không có khả năng truy xuất thông tin chi tiết và cập nhật như kết quả thi đấu. RAG, bằng cách truy xuất từ kho tri thức, đã đưa ra được câu trả lời gần khớp tuyệt đối với thông tin tham chiếu, minh chứng cho khả năng truy xuất hiệu quả hơn.

---

6. Kết luận

Hệ thống RAG được phát triển đã cho thấy khả năng ứng dụng trong việc trả lời câu hỏi dựa trên một kho tri thức tùy chỉnh. Qua các thử nghiệm, kiến trúc multi-hop RAG, với sự hỗ trợ của Planner LLM, đã thể hiện tiềm năng cải thiện chất lượng câu trả lời so với kiến trúc single-hop RAG cơ bản, đặc biệt đối với các câu hỏi phức tạp cần tổng hợp thông tin. Việc kết hợp truy xuất thông tin với khả năng của các LLM mạnh mẽ cũng chứng minh hiệu quả vượt trội so với việc sử dụng LLM ở chế độ closed-book cho các tác vụ yêu cầu kiến thức chuyên biệt.

- Điểm mạnh:
    - Khả năng tùy chỉnh kho tri thức.
    - Sử dụng semantic chunking giúp cải thiện tính mạch lạc của ngữ cảnh truy xuất.
    - Kiến trúc multi-hop linh hoạt, có tiềm năng xử lý câu hỏi phức tạp.

- Hạn chế:
    - Chất lượng câu trả lời phụ thuộc nhiều vào chất lượng dữ liệu thu thập, model embedding và các LLM được sử dụng (đặc biệt là Planner LLM).
    - Độ trễ có thể tăng lên với nhiều hop truy xuất.
    - Giới hạn của API

- Hướng phát triển tương lai:
    - Cải thiện logic của Planner LLM để đưa ra các câu hỏi theo dõi thông minh hơn và có cơ chế dừng sớm hiệu quả hơn.

    - Thử nghiệm với các model embedding và LLM tiên tiến hơn.
    - Mở rộng và cập nhật liên tục kho tri thức.
    - Tối ưu hóa tốc độ và hiệu quả của quá trình truy xuất.
    - Fine-tuning các LLM (Planner và Generator) trên dữ liệu chuyên biệt của VNU để cải thiện hơn nữa hiệu năng.

---

7. Tài liệu Tham khảo

- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33, 9459-9474.
- FAISS: https://github.com/facebookresearch/faiss
- Sentence Transformers: https://www.sbert.net
- Hugging Face Transformers: https://huggingface.co/transformers
- OpenRouter: https://openrouter.ai