TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

VIỆN TRÍ TUỆ NHÂN TẠO

***

BÁO CÁO PROJECT
Retrieval Augmented Generation

Thành viên nhóm:

1. Lê Vũ Hiếu
2. Đàm Lê Minh Quân
3. Nguyễn Hoàng Tú

Hà Nội, 06/2025

Phần I: Lý thuyết về Retrieval-Augmented Generation (RAG)

I. Lý thuyết

1. Giới thiệu

1.1. Bối cảnh

Các Mô hình Ngôn ngữ Lớn (LLM), dù đã đạt được những thành tựu đáng kể, vẫn đối mặt với các hạn chế nhất định. Một trong những vấn đề chính là xu hướng tạo ra thông tin sai lệch (hallucinations), tức là đưa ra các câu trả lời không có căn cứ thực tế. Điều này phát sinh do LLM chỉ dựa vào kiến thức được học trong quá trình huấn luyện từ tập dữ liệu tĩnh, dẫn đến việc không thể truy cập hoặc cập nhật thông tin mới nhất sau thời điểm dữ liệu được thu thập.

Ngoài ra, LLM thường gặp khó khăn trong việc cung cấp thông tin chuyên sâu hoặc chi tiết từ các nguồn dữ liệu cụ thể, ví dụ như tài liệu nội bộ hoặc các lĩnh vực chuyên biệt. Để giải quyết những thách thức này và nâng cao độ tin cậy, tính minh bạch cũng như khả năng áp dụng của LLM trong các nhiệm vụ đòi hỏi tri thức chuyên sâu, kỹ thuật Retrieval-Augmented Generation (RAG) đã được phát triển.

1.2. RAG là gì?

Retrieval-Augmented Generation (RAG) là một khuôn khổ mạnh mẽ kết hợp khả năng sinh văn bản của Mô hình Ngôn ngữ Lớn (LLM) với khả năng truy xuất thông tin từ một kho dữ liệu bên ngoài. Thay vì chỉ dựa vào kiến thức nội tại đã học, một hệ thống RAG sẽ tìm kiếm và trích xuất các đoạn thông tin liên quan từ một cơ sở tri thức (knowledge base) và sau đó sử dụng các đoạn thông tin này làm ngữ cảnh bổ sung để LLM tạo ra câu trả lời.

1.3. Ưu điểm của RAG

Việc tích hợp cơ chế truy xuất mang lại nhiều ưu điểm vượt trội cho RAG so với việc sử dụng LLM độc lập:

- **Giảm thiểu hiện tượng "Hallucinations" và tăng tính chính xác:** RAG giúp LLM căn cứ câu trả lời vào thông tin thực tế được truy xuất từ cơ sở dữ liệu, giảm đáng kể khả năng tạo ra các thông tin sai lệch hoặc không có căn cứ.
- **Cập nhật kiến thức:** RAG cho phép hệ thống truy cập và sử dụng thông tin mới nhất từ cơ sở dữ liệu được cập nhật thường xuyên, vượt qua giới hạn về thời gian huấn luyện của LLM.
- **Minh bạch và khả năng truy vết nguồn gốc:** RAG có thể chỉ ra chính xác các nguồn tài liệu hoặc đoạn văn bản đã được sử dụng để tạo ra câu trả lời, tăng cường sự tin cậy và khả năng kiểm chứng.

- **Giảm chi phí và độ phức tạp trong huấn luyện:** Thay vì phải huấn luyện lại (fine-tuning) toàn bộ LLM mỗi khi có dữ liệu mới hoặc cần thêm kiến thức chuyên biệt, RAG chỉ cần cập nhật cơ sở tri thức và các embedding của nó, tiết kiệm đáng kể tài nguyên tính toán và thời gian.
- **Khả năng thích ứng với các miền tri thức chuyên biệt:** RAG cho phép LLM trả lời các câu hỏi về thông tin chuyên ngành hoặc nội bộ mà nó chưa từng được thấy trong quá trình huấn luyện ban đầu, chỉ bằng cách bổ sung cơ sở tri thức tương ứng.

2. Các thành phần chính và cơ chế hoạt động của RAG

Một hệ thống Retrieval-Augmented Generation (RAG) được xây dựng từ nhiều thành phần cốt lõi, hoạt động phối hợp để xử lý truy vấn của người dùng và tạo ra câu trả lời dựa trên thông tin được truy xuất. Các thành phần chính bao gồm cơ sở tri thức, mô hình embedding, mô hình truy xuất và mô hình ngôn ngữ lớn.

2.1. Cơ sở tri thức (Knowledge Base/Corpus)

- **Vai trò:** Là kho lưu trữ tập hợp các tài liệu nguồn bên ngoài LLM, chứa đựng tri thức mà hệ thống RAG có thể truy xuất.
- **Cách hoạt động:**
    - **Thu thập dữ liệu (Data Ingestion):** Quá trình này bao gồm việc thu thập các tài liệu từ nhiều nguồn khác nhau, có thể là văn bản, PDF, trang web, hoặc
    - Khi một đoạn văn bản hoặc một truy vấn (câu hỏi) được đưa vào mô hình embedding, nó sẽ được xử lý để tạo ra một chuỗi các số (vector).
    - Các vector này được thiết kế để nắm bắt ngữ nghĩa của văn bản; các văn bản có ý nghĩa tương tự sẽ được biểu diễn bởi các vector gần nhau trong không gian embedding.các định dạng dữ liệu khác.
    - **Tiền xử lý (Preprocessing):** Các tài liệu thô được làm sạch để loại bỏ nhiễu, chuẩn hóa định dạng, và xử lý các lỗi hoặc dữ liệu không nhất quán.
    - **Chia đoạn (Chunking):** Sau khi tiền xử lý, các tài liệu lớn thường được chia thành các đoạn văn (chunks) nhỏ hơn và có ý nghĩa hơn. Việc chia chunk là rất quan trọng để đảm bảo rằng mỗi đoạn văn có thể được đưa vào ngữ cảnh của LLM một cách hiệu quả, tránh việc các đoạn quá dài hoặc quá ngắn làm mất đi tính liên kết ngữ nghĩa. Các chiến lược chia chunk có thể bao gồm chia theo kích thước cố định, theo ngữ nghĩa (semantic chunking), hoặc dựa trên cấu trúc tài liệu (ví dụ: theo tiêu đề, đoạn văn).

2.2. Mô hình Embedding (Embedder/Encoder)

- **Vai trò:** Chuyển đổi các đoạn văn bản (từ cơ sở tri thức) và câu hỏi của người dùng thành các vector số (embeddings) trong một không gian đa chiều.
- **Cách hoạt động:**
    - Sau khi tạo ra, các embeddings của các đoạn văn bản từ cơ sở tri thức sẽ được lập chỉ mục (indexed) và lưu trữ trong một cơ sở dữ liệu vector (vector database) để cho phép tìm kiếm độ tương đồng một cách hiệu quả và nhanh chóng.

2.3. Mô hình Truy xuất (Retriever)

- **Vai trò:** Xác định và trích xuất các đoạn văn bản có liên quan nhất từ cơ sở tri thức dựa trên câu hỏi của người dùng.
- **Cách hoạt động:**
    - Đầu tiên, câu hỏi của người dùng được chuyển đổi thành một vector embedding bằng cách sử dụng cùng mô hình embedding đã dùng để mã hóa cơ sở tri thức.
    - Sau đó, retriever sẽ thực hiện tìm kiếm độ tương đồng (ví dụ: sử dụng độ tương đồng cosine) giữa embedding của câu hỏi và các embeddings của tất cả các đoạn văn bản trong cơ sở dữ liệu vector.
    - Các đoạn văn bản được xếp hạng dựa trên điểm tương đồng của chúng với câu hỏi. Retriever sẽ trả về một danh sách các đoạn văn bản có điểm số cao nhất (thường là top-K đoạn) để làm ngữ cảnh tiềm năng.
    - Các phương pháp truy xuất có thể là truy xuất thưa (sparse retrieval) dựa trên từ khóa (ví dụ: BM25), truy xuất dày đặc (dense retrieval) dựa trên embeddings, hoặc kết hợp cả hai (hybrid retrieval) để tận dụng ưu điểm của từng phương pháp.

2.4. Mô hình Ngôn ngữ Lớn (Large Language Model - LLM)

- **Vai trò:** Sinh ra một câu trả lời mạch lạc, chính xác và được căn cứ vào thông tin thực tế từ các đoạn văn bản đã được truy xuất.
- **Cách hoạt động:**
    - LLM nhận một đầu vào kết hợp: câu hỏi gốc của người dùng và các đoạn văn bản liên quan được cung cấp bởi retriever.
    - Với ngữ cảnh bổ sung này, LLM sử dụng khả năng hiểu ngôn ngữ tự nhiên và sinh văn bản của mình để tổng hợp thông tin, đưa ra câu trả lời trực tiếp giải quyết truy vấn của người dùng, đồng thời giảm thiểu khả năng tạo ra thông tin sai lệch (hallucinations).

- Các chiến lược kỹ thuật prompt (prompt engineering) cũng được áp dụng để hướng dẫn LLM cách kết hợp ngữ cảnh và truy vấn để tạo ra phản hồi mong muốn.

2.5. Cơ chế kết hợp (Generation Workflow)

Cơ chế kết hợp là sự điều phối tổng thể của hệ thống RAG, đảm bảo tương tác liền mạch giữa các thành phần. Có thể phân chia thành hai luồng hoạt động chính:

- **Luồng xử lý tài liệu (Document Processing Flow):**

1. **Thu thập và Chuẩn bị:** Các tài liệu thô được thu thập từ các nguồn khác nhau.
2. **Tiền xử lý & Chia đoạn:** Tài liệu được làm sạch và chia thành các đoạn văn bản nhỏ hơn, có ý nghĩa.
3. **Tạo Embeddings:** Mỗi đoạn văn bản được chuyển đổi thành một vector số (embedding) bằng mô hình embedding.
4. **Lập chỉ mục và Lưu trữ:** Các embeddings này được lập chỉ mục và lưu trữ trong một cơ sở dữ liệu vector chuyên dụng, tạo thành kho tri thức có thể tìm kiếm.

- **Luồng hỏi đáp (Question Answering Flow):**

1. **Nhận truy vấn:** Hệ thống nhận câu hỏi từ người dùng.
2. **Mã hóa truy vấn:** Câu hỏi của người dùng được chuyển đổi thành vector embedding.
3. **Truy xuất thông tin:** Retriever sử dụng embedding của truy vấn để tìm kiếm các đoạn văn bản liên quan nhất từ cơ sở dữ liệu vector.
4. **Bổ sung ngữ cảnh:** Các đoạn văn bản được truy xuất được đưa vào làm ngữ cảnh bổ sung cho LLM, cùng với câu hỏi gốc.
5. **Tạo câu trả lời:** LLM xử lý cả câu hỏi và ngữ cảnh được bổ sung để tạo ra câu trả lời cuối cùng.
6. **Trình bày đầu ra:** Câu trả lời được trình bày cho người dùng, thường kèm theo thông tin về các nguồn tài liệu đã được sử dụng để tăng tính minh bạch và độ tin cậy.

3. Các kỹ thuật cải tiến RAG (Advanced RAG Techniques)

Mặc dù kiến trúc RAG cơ bản đã mang lại những cải thiện đáng kể cho các Mô hình Ngôn ngữ Lớn (LLM), vẫn có những thách thức trong việc đảm bảo chất lượng, tính chính xác và hiệu quả tối ưu. Các kỹ thuật RAG nâng cao tập trung vào việc tinh chỉnh từng giai đoạn của quy trình (xử lý dữ liệu, truy xuất, và tạo sinh) để khắc phục những hạn chế này.

3.1. Chiến lược chunking nâng cao (Advanced chunking strategies)

- **Vấn đề:** Việc chia tài liệu thành các đoạn văn (chunks) với kích thước cố định, mặc dù đơn giản, có thể gây ra các vấn đề như làm mất ngữ cảnh (khi thông tin quan trọng bị cắt ngang giữa hai chunk) hoặc chứa quá nhiều thông tin không liên quan.
- **Giải pháp:**
    - **Semantic chunking (Chia đoạn theo ngữ nghĩa):** Thay vì chỉ cắt theo độ dài, phương pháp này cố gắng chia tài liệu thành các đoạn dựa trên ranh giới ngữ nghĩa hoặc các đơn vị thông tin logic. Điều này giúp đảm bảo rằng mỗi chunk đại diện cho một ý tưởng hoàn chỉnh, mạch lạc, từ đó cải thiện chất lượng của ngữ cảnh được đưa vào LLM.
    - **Parent document retriever (Truy xuất tài liệu gốc/cha):** Kỹ thuật này liên quan đến việc tạo và lưu trữ hai loại chunk: các chunk nhỏ hơn để phục vụ mục đích truy xuất (để tăng cường độ chính xác khi tìm kiếm tương đồng) và các "tài liệu cha" lớn hơn chứa ngữ cảnh rộng hơn. Khi một chunk nhỏ được truy xuất, toàn bộ tài liệu cha liên quan sẽ được đưa vào LLM để cung cấp ngữ cảnh đầy đủ hơn, giải quyết vấn đề mất ngữ cảnh khi chunk quá nhỏ.

3.2. Cải tiến truy xuất (Enhanced Retrieval)

- **Vấn đề:** Các phương pháp truy xuất ban đầu, đặc biệt là tìm kiếm độ tương đồng vector đơn thuần, có thể không đủ để xác định tất cả các tài liệu liên quan hoặc có thể trả về các kết quả không tối ưu.
- **Giải pháp:**
    - **Hybrid search (Tìm kiếm kết hợp):** Đây là chiến lược kết hợp các ưu điểm của nhiều phương pháp truy xuất khác nhau. Phổ biến nhất là kết hợp truy xuất thưa (sparse retrieval), dựa trên từ khóa và tần suất xuất hiện (ví dụ: BM25), với truy xuất dày đặc (dense retrieval), dựa trên embeddings và độ tương đồng ngữ nghĩa. Việc kết hợp này giúp hệ thống vừa bắt được các từ khóa chính xác, vừa hiểu được ý nghĩa sâu sắc của truy vấn, từ đó tăng cường độ phủ và độ chính xác của kết quả truy xuất.
    - **Query transformation (Biến đổi truy vấn):** Phương pháp này bao gồm việc sửa đổi hoặc mở rộng truy vấn ban đầu của người dùng để cải thiện hiệu quả truy xuất. Ví dụ, một LLM có thể được sử dụng để mở rộng truy vấn với các từ đồng nghĩa hoặc các câu hỏi liên quan, hoặc để tạo ra một câu trả lời giả định (Hypothetical Document Embeddings - HyDE). Sau đó, embedding của câu trả lời giả định này được sử dụng để tìm kiếm các tài liệu thực tế tương tự, giúp tìm ra các nguồn liên quan hơn về mặt ngữ nghĩa.

- **Re-ranking (Sắp xếp lại thứ hạng):**
    - **Vai trò:** Tinh lọc và sắp xếp lại thứ tự ưu tiên của các đoạn văn bản đã được truy xuất ban đầu, đảm bảo rằng những thông tin liên quan nhất được đặt ở vị trí cao nhất.
    - **Cách hoạt động:** Sau khi mô hình truy xuất ban đầu trả về một tập hợp lớn các đoạn văn bản tiềm năng, một mô hình Re-ranker riêng biệt, thường là một cross-encoder, sẽ được sử dụng. Mô hình này nhận từng cặp (truy vấn, đoạn văn bản) và tính toán lại điểm số mức độ liên quan của chúng một cách chính xác hơn, bằng cách phân tích tương tác ngữ nghĩa giữa truy vấn và đoạn văn bản. Dựa trên các điểm số mới này, danh sách các đoạn văn bản được sắp xếp lại, và chỉ một số lượng nhỏ các đoạn có điểm cao nhất sẽ được chọn để đưa vào LLM. Điều này giúp loại bỏ thông tin nhiễu và cung cấp ngữ cảnh chất lượng cao hơn cho LLM.

3.3. Tối ưu hóa Tạo sinh (Generation Optimization)

- **Vấn đề:** Ngay cả khi có ngữ cảnh chất lượng cao, LLM vẫn cần được hướng dẫn để sử dụng thông tin đó một cách hiệu quả, tránh vượt quá giới hạn token hoặc tạo ra câu trả lời không theo định dạng mong muốn.
- **Giải pháp:**
    - **Contextual compression (Nén ngữ cảnh):** Khi số lượng hoặc độ dài của các đoạn văn bản được truy xuất vẫn còn lớn, vượt quá giới hạn token của LLM, các kỹ thuật nén ngữ cảnh có thể được áp dụng. Điều này bao gồm việc tóm tắt các đoạn văn bản hoặc trích xuất những thông tin cốt lõi nhất từ chúng trước khi đưa vào LLM. Mục tiêu là giảm lượng thông tin đầu vào cho LLM mà vẫn giữ được nội dung quan trọng.
    - **Advanced prompt engineering (Kỹ thuật prompt nâng cao):** Thiết kế prompt đóng vai trò quan trọng trong việc hướng dẫn LLM sử dụng ngữ cảnh một cách hiệu quả và tạo ra câu trả lời mong muốn. Các kỹ thuật như Chain-of-Thought (CoT) prompting có thể được sử dụng để khuyến khích LLM suy nghĩ từng bước trước khi đưa ra câu trả lời cuối cùng, hoặc few-shot prompting (cung cấp các ví dụ mẫu) giúp LLM hiểu rõ hơn định dạng và phong cách trả lời mong muốn.

3.4. Huấn luyện đồng thời (Joint training)

- **Mô tả:** Đây là một phương pháp tối ưu hóa RAG ở cấp độ mô hình và huấn luyện, vượt xa việc chỉ ghép nối các thành phần đã được huấn luyện trước.
- **Cách hoạt động:** Mô hình truy xuất (retriever) và mô hình tạo sinh (generator) được huấn luyện cùng lúc, end-to-end. Thay vì huấn luyện độc

lập, quá trình huấn luyện chung cho phép retriever học cách tìm kiếm các tài liệu không chỉ liên quan mà còn tối ưu cho việc sinh ra câu trả lời của generator. Đồng thời, generator học cách sử dụng các tài liệu được truy xuất hiệu quả hơn. Việc huấn luyện đồng thời giúp các thành phần RAG "hợp tác" với nhau một cách tối ưu hơn, vượt qua giới hạn của việc ghép nối các mô hình được huấn luyện riêng biệt.

- **Ý nghĩa:** Huấn luyện đồng thời đại diện cho một hướng đi quan trọng trong việc tối ưu hóa RAG về mặt học thuật và phát triển mô hình, mặc dù nó đòi hỏi tài nguyên tính toán và độ phức tạp cao hơn so với RAG cơ bản.

Phần II: Thực nghiệm và Đánh giá Ứng dụng RAG

1. Thực nghiệm:

1.1. Mục tiêu:

- Mục tiêu của phần thực nghiệm này là xây dựng và triển khai một hệ thống Hỏi-Đáp (Question Answering) dựa trên kiến trúc RAG (Retrieval-Augmented Generation) nâng cao. Hệ thống được thiết kế để vượt qua các hạn chế của một pipeline RAG cơ bản, tập trung vào việc cải thiện độ chính xác, sự liên quan của thông tin truy xuất và chất lượng của câu trả lời được tạo ra.

Các kỹ thuật nâng cao được tích hợp:

- Parent Document Retrieval
- Hybrid Search & Re-ranking
- Contextual Compression & Advanced Prompting

1.2. Luồng hoạt động tổng thể của hệ thống

Khi người dùng gửi một câu hỏi, hệ thống sẽ thực hiện chuỗi các bước sau:

- **Input:** Nhận câu hỏi từ người dùng.
- **Hybrid Search:** EnsembleRetriever đồng thời thực hiện tìm kiếm từ khóa (BM25) và tìm kiếm ngữ nghĩa (FAISS) trên các **chunks con** để lấy ra một danh sách ứng viên.
- **Parent Retrieval:** Hệ thống xác định các **chunks cha** tương ứng với các chunks con được tìm thấy.
- **Re-ranking:** CohereRerank sắp xếp lại danh sách các chunks cha, chọn ra 8 chunks liên quan nhất.
- **Context Structuring:** Hàm reorder_documents sắp xếp lại 8 chunks này để chống lại hiệu ứng "Lost in the Middle".

- **Prompt Injection:** Các chunks đã được tối ưu hóa được chèn vào Advanced Prompt cùng với câu hỏi của người dùng.
- **Generation:** Toàn bộ prompt được gửi đến mô hình ChatOllama (llama3).
- **Output:** LLM tạo ra câu trả lời dựa trên ngữ cảnh và các chỉ dẫn đã nhận được. Câu trả lời và các nguồn tham khảo (chunks cha đã được re-rank) được hiển thị cho người dùng.

2. Kết quả

Để kiểm chứng hiệu quả của hệ thống Advanced-RAG, nhóm em đã tạo ra một bộ câu hỏi dựa trên một bộ tài liệu xác định, cụ thể ở đây là tài liệu giáo trình Kinh Tế Chính Trị VNU. Việc lựa chọn tài liệu này nhằm mục đích thử thách khả năng của hệ thống trong việc xử lý ngôn ngữ học thuật, các khái niệm trừu tượng và các mối quan hệ logic phức tạp.

Hiệu suất của hệ thống được đo lường tự động thông qua một bộ các chỉ số (metrics) tiêu chuẩn trong đánh giá RAG, bao gồm:

- **Context Precision** (Độ chính xác của Ngữ cảnh)
- **Context Relevancy** (Sự liên quan của Ngữ cảnh)
- **Context Recall** (Độ phủ của Ngữ cảnh)
- **Faithfulness** (Mức độ Trung thực)
- **Answer Relevancy** (Sự liên quan của Câu trả lời)

2.1. Thông số kết quả

Metric	Score
Context Precision	0.9912
Context Relevancy	1.0
Context Recall	0.9298
Faithfulness	0.797
Answer Relevancy	0.7801

2.2. Nhận xét & đánh giá

2.2.1. Giai đoạn hệ thống tìm kiếm thông tin trong tài liệu

Gồm các chỉ số Context Precision, Context Relevancy, và Context Recall.

- **Context Relevancy (1.0) & Context Precision (0.9912):**
    - Ý nghĩa: Các chỉ số này đo lường xem ngữ cảnh (các chunks) mà hệ thống truy xuất được có thực sự liên quan đến câu hỏi hay không.
    - Nhận xét: Điểm số gần như tuyệt đối cho thấy khâu truy xuất của hệ thống hoạt động cực kỳ hiệu quả. Sự kết hợp giữa Hybrid Search (BM25 + FAISS) đã đảm bảo không bỏ sót các thông tin quan trọng (cả về từ khóa và ngữ nghĩa), trong khi Cohere Re-ranker đã làm tốt nhiệm vụ lọc bỏ các chunks nhiễu, chỉ giữ lại những thông tin chất lượng nhất.
- **Context Recall (0.9298):**
    - Ý nghĩa: Chỉ số này đo lường xem hệ thống có truy xuất được tất cả các thông tin cần thiết để trả lời câu hỏi một cách trọn vẹn hay không.
    - Nhận xét: Điểm số rất cao (93%) cho thấy hệ thống gần như luôn tìm thấy đầy đủ các mẩu thông tin liên quan trong tài liệu. Việc sử dụng chiến lược Parent Document Retrieval đã phát huy tác dụng, vì các "chunks cha" lớn chứa đựng ngữ cảnh rộng, làm tăng khả năng bao hàm hết các ý cần thiết. Điểm số chưa đạt 1.0 có thể do một vài trường hợp thông tin cần thiết nằm rải rác ở nhiều "chunks cha" khác nhau và một trong số chúng không được re-ranker xếp hạng đủ cao.

Giai đoạn Retrieval là điểm sáng của hệ thống, hoạt động với hiệu suất gần như hoàn hảo. Hệ thống đã chứng tỏ khả năng xác định và trích xuất đúng và đủ thông tin liên quan từ kho kiến thức.

2.2.2. Giai đoạn LLM (Llama 3) sử dụng ngữ cảnh được cung cấp để tạo ra câu trả lời cuối cùng

Gồm các chỉ số: Faithfulness và Answer Relevancy.

- **Faithfulness (0.797):**
    - Ý nghĩa: Đo lường mức độ câu trả lời được tạo ra có bám sát và hoàn toàn dựa trên ngữ cảnh đã cung cấp, tránh "ảo giác" (hallucination).
    - Nhận xét: Điểm số ở mức khá tốt (gần 80%). Điều này cho thấy Advanced Prompt đã có hiệu quả trong việc hướng dẫn LLM tuân thủ quy tắc "chỉ trả lời dựa trên tài liệu". Tuy nhiên, vẫn còn khoảng 20% trường hợp LLM có thể diễn giải mở rộng hoặc thêm thắt các chi tiết nhỏ không có trong ngữ cảnh. Đây là một hạn chế cố hữu của các mô hình ngôn ngữ hiện tại, nhưng kết quả này vẫn ở mức có thể tạm chấp nhận được.
- **Answer Relevancy (0.7801):**

- Ý nghĩa: Là chỉ số quan trọng nhất, đo lường xem câu trả lời cuối cùng có thực sự đáp ứng đúng và trọn vẹn câu hỏi của người dùng hay không.
- Nhận xét: Điểm số rất cao (78.01%) là một kết quả khá tích cực và là minh chứng cho sự thành công của toàn bộ hệ thống. Nó cho thấy rằng công sức tối ưu hóa ở các khâu truy xuất trước đó đã không bị lãng phí và đã được chuyển hóa thành công thành một câu trả lời chất lượng cho người dùng cuối.

Qua đây, giai đoạn tạo sinh của hệ thống đã đạt được một sự cân bằng hiệu quả giữa tính hữu dụng và độ tin cậy, với kết quả tổng thể khá là tích cực. Điểm số cao của Answer Relevancy cho thấy mô hình LLM, khi được hỗ trợ bởi một ngữ cảnh chất lượng, đã có khả năng tổng hợp và đưa ra câu trả lời đáp ứng đúng nhu cầu của người dùng. Tuy nhiên, với điểm số của Faithfulness hệ thống vẫn cần khắc phục để đạt được sự trung thực tuyệt đối, giảm thiểu hoàn toàn việc diễn giải thông tin ngoài ngữ cảnh được cung cấp.

3. Kết luận chung và hướng phát triển

3.1. Kết luận

Hệ thống Advanced-RAG được xây dựng đã đạt được hiệu quả vượt trội và là một giải pháp thành công trong việc giải quyết bài toán hỏi-đáp trên tài liệu phức tạp. Việc tích hợp một cách có chủ đích các kỹ thuật nâng cao đã tạo ra một pipeline mạnh mẽ, có khả năng vượt qua nhiều thách thức cố hữu của các hệ thống RAG cơ bản. Mặc dù còn một vài điểm yếu cần khắc phục, nhưng hệ thống đã cho thấy những điểm mạnh nổi trội của nó như truy xuất tốt, khả năng tạo sinh hiệu quả. Hệ thống đã chứng tỏ rằng việc đầu tư vào một kiến trúc RAG nâng cao mang lại lợi ích rõ rệt, biến một mô hình LLM tầm trung thành một công cụ hỏi-đáp mạnh mẽ, cung cấp giá trị thực tiễn to lớn cho người sử dụng.

3.2. Hướng phát triển

- Tối ưu hóa prompt nghiêm ngặt hơn với những chỉ dẫn mạnh mẽ hơn nữa, hoặc Triển khai bước kiểm tra chéo
- Tinh chỉnh và tối ưu hóa pipeline như điều chỉnh các tham số, nâng cấp mô hình LLM.

Phần III: Phụ lục

1. Thành viên nhóm

Tên	MSV
Lê Vũ Hiếu	23020365

Đàm Lê Minh Quân	23020416
Nguyễn Hoàng Tú	23020428

2. Tài liệu tham khảo

- Retrieval-Augmented Generation for Large Language Models: A Survey
- Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks
- What Is Retrieval-Augmented Generation, aka RAG?