

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ
VIỆN TRÍ TUỆ NHÂN TẠO**

-----***-----

**BÁO CÁO MÔN HỌC KỸ THUẬT VÀ CÔNG NGHỆ DỮ LIỆU LỚN
ĐỀ TÀI
CHATBOT SỬ DỤNG CORRECTIVE RAG, ELASTICSEARCH,
LANGCHAIN**

Nhóm sinh viên thực hiện:

1. Nguyễn Gia Lộc
2. Đỗ Tiên Dũng
3. Vũ Minh Khải
4. Trần Đức Đăng Khôi

Giảng viên hướng dẫn: TS. Trần Hồng Việt

ThS.Ngô Minh Hương

HÀ NỘI, 12/2024

MỞ ĐẦU

Công nghệ Big data đã đạt đến đỉnh cao trong việc thực hiện các chức năng của nó. Trong tháng 8/2015 Big data đã vượt ra khỏi bảng xếp hạng những công nghệ mới nổi Cycle Hype của Gartner và tạo ra một tiếng vang lớn cho xu hướng công nghệ của thế giới. Big data chứa trong mình rất nhiều thông tin quý giá mà nếu mà trích xuất thành công, nó sẽ giúp rất nhiều trong nhiều lĩnh vực như y tế, giao thông, giáo dục, ...

Chính vì thế, các framework hỗ trợ xử lý Big Data cũng ngày càng được chú trọng phát triển. Một trong những công nghệ cốt lõi trong việc lưu trữ và tìm kiếm số lượng lớn dữ liệu là Elasticsearch – một công cụ mạnh mẽ cho việc quản lý, tìm kiếm và phân tích Big Data với khả năng xử lý thời gian thực và hỗ trợ mở rộng quy mô linh hoạt.

Từ đó, chúng em đã chọn đề tài: "**Chatbot sử dụng Corrective RAG, Elasticsearch, Langchain**" để làm báo kết thúc môn học của mình.

Báo cáo gồm 5 chương:

Chương 1: Mô hình LLM.

Chương 2: Kỹ thuật CRAG.

Chương 3: ElasticSearch và LangChain.

Chương 4: Ứng dụng và thiết kế chatbot.

Chương 5: Kết luận và hướng phát triển.

MỤC LỤC

MỞ ĐẦU

CHƯƠNG 1: MÔ HÌNH LLM

1.1 Mô hình ngôn ngữ lớn là gì ?

1.2 Các mô hình LLM nổi tiếng

1.3 LLM Chatbot

CHƯƠNG 2: KỸ THUẬT CRAG

2.1 Hiện tượng ảo giác trong LLM

2.2 Kỹ thuật RAG

2.3 Corrective RAG

CHƯƠNG 3: ELASTICSEARCH & LANGCHAIN

3.1 LangChain

3.2 ElasticSearch

CHƯƠNG 4: ỨNG DỤNG VÀ THIẾT KẾ CHATBOT

4.1 Luồng hoạt động ứng dụng

4.2 Xây dựng ứng dụng

4.3 So sánh kết quả

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

TÀI LIỆU THAM KHẢO

NHIỆM VỤ CỦA CÁC THÀNH VIÊN

CHƯƠNG 1: MÔ HÌNH LLM

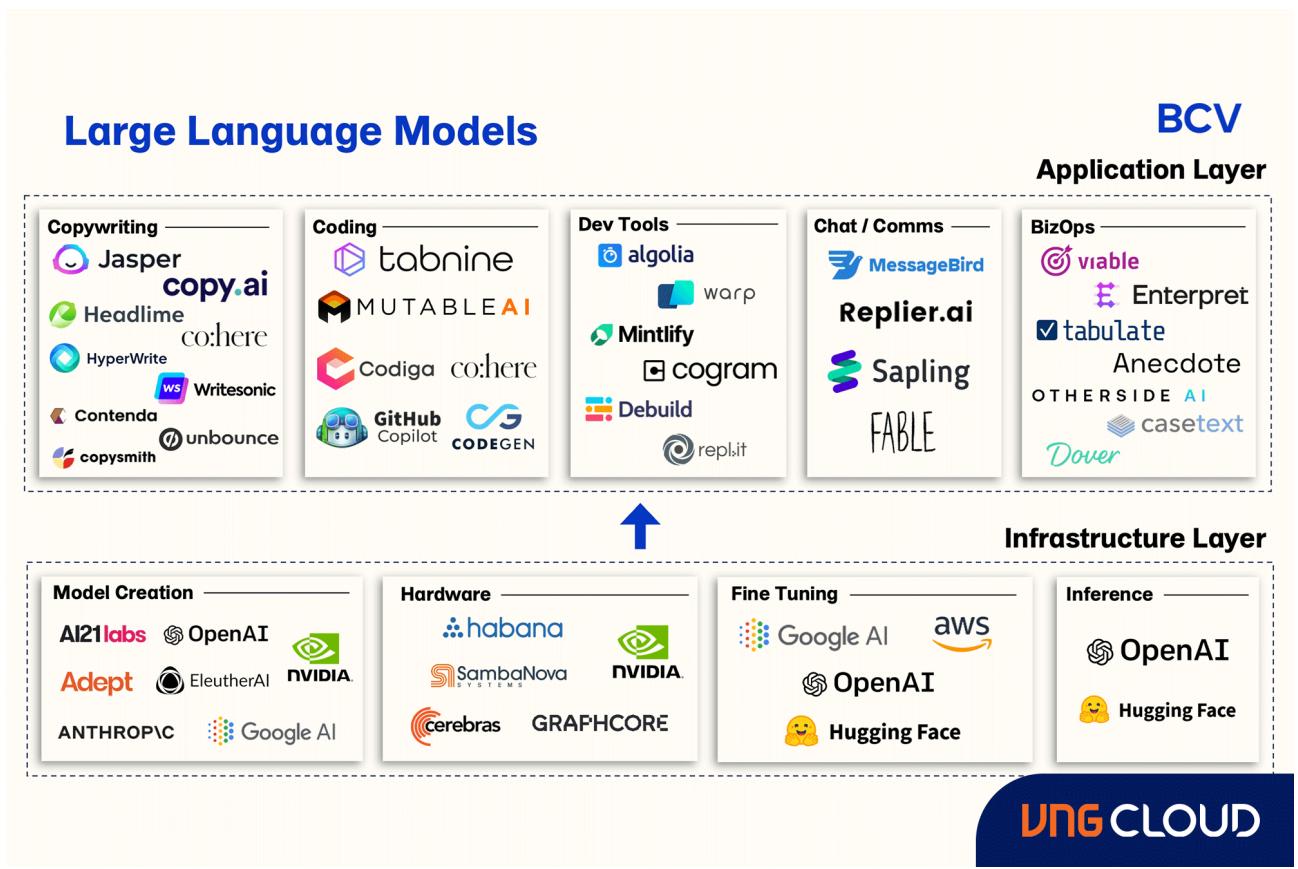
1.1 Mô hình ngôn ngữ lớn là gì ?

Mô hình Ngôn ngữ lớn (**Large Language Model - LLM**) là thuật toán deep learning mạnh mẽ với khả năng xử lý nhiều tác vụ trong natural language processing (NLP). Những mô hình này sử dụng transformer model và trải qua quá trình đào tạo chuyên sâu với các bộ dữ liệu khổng lồ. Quá trình này trang bị cho chúng khả năng nhận diện, dịch thuật, dự đoán hoặc tạo ra văn bản và các dạng nội dung khác.

Bên cạnh đó, **LLM** đôi khi được gọi là neural networks (NNs), lấy cảm hứng từ cấu trúc của bộ não con người. Những mạng lưới thần kinh này hoạt động thông qua các node được kết nối với nhau, giống như các tế bào thần kinh trong não của chúng ta.

Ngoài khả năng giảng dạy cho các ứng dụng trí tuệ nhân tạo (AI) ngôn ngữ của con người, **LLM** có thể được đào tạo để thực hiện tốt trong nhiều công việc, chẳng hạn như tìm hiểu cấu trúc protein hoặc viết code phần mềm, cùng nhiều tác vụ khác. Tương tự như bộ não con người, **LLM** phải trải qua quá trình đào tạo và sau đó là điều chỉnh để vượt trội trong các lĩnh vực như phân loại văn bản, trả lời câu hỏi, tóm tắt tài liệu và tạo nội dung. Khả năng giải quyết vấn đề của chúng có thể ứng dụng trong các lĩnh vực như y tế, tài chính và giải trí, nơi **LLM** phục vụ một loạt các ứng dụng NLP, bao gồm dịch thuật, chatbot, trợ lý AI, v.v.

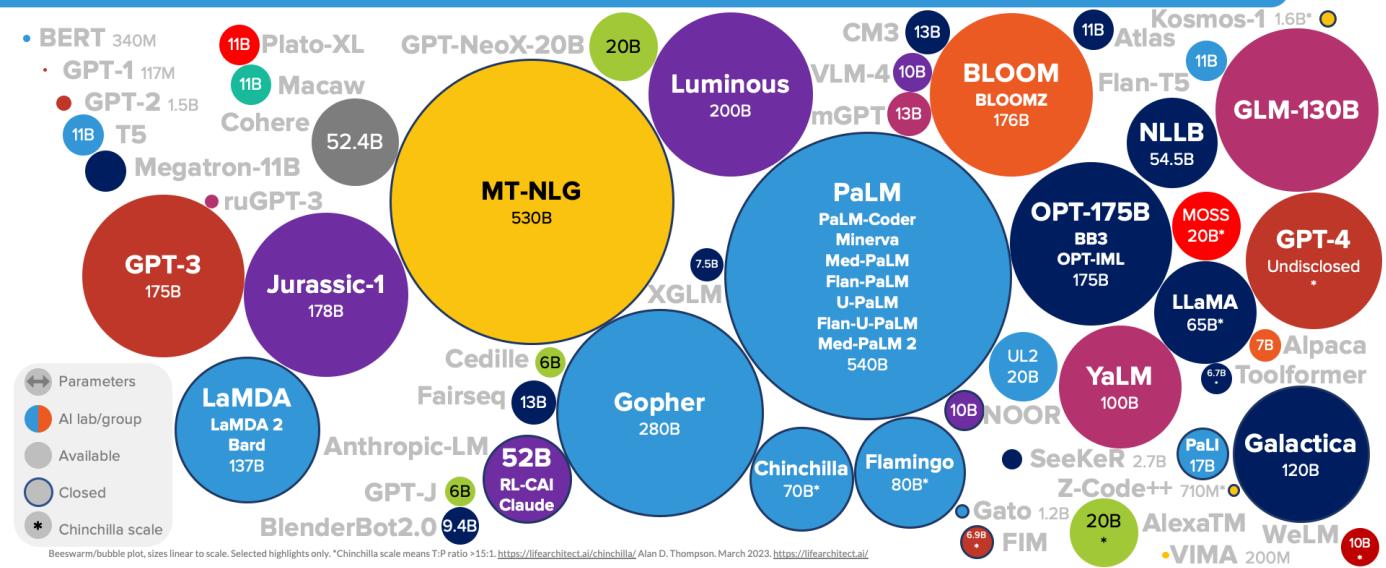
Đặc biệt, **LLM** có số lượng tham số lớn, giống như những kiến thức mà mô hình thu được khi học từ quá trình đào tạo. Những tham số này đại diện cho nguồn thông tin kiến thức của mô hình.



Hình 1: Các ứng dụng của mô hình ngôn ngữ lớn

1.2 Các mô hình LLM nổi tiếng

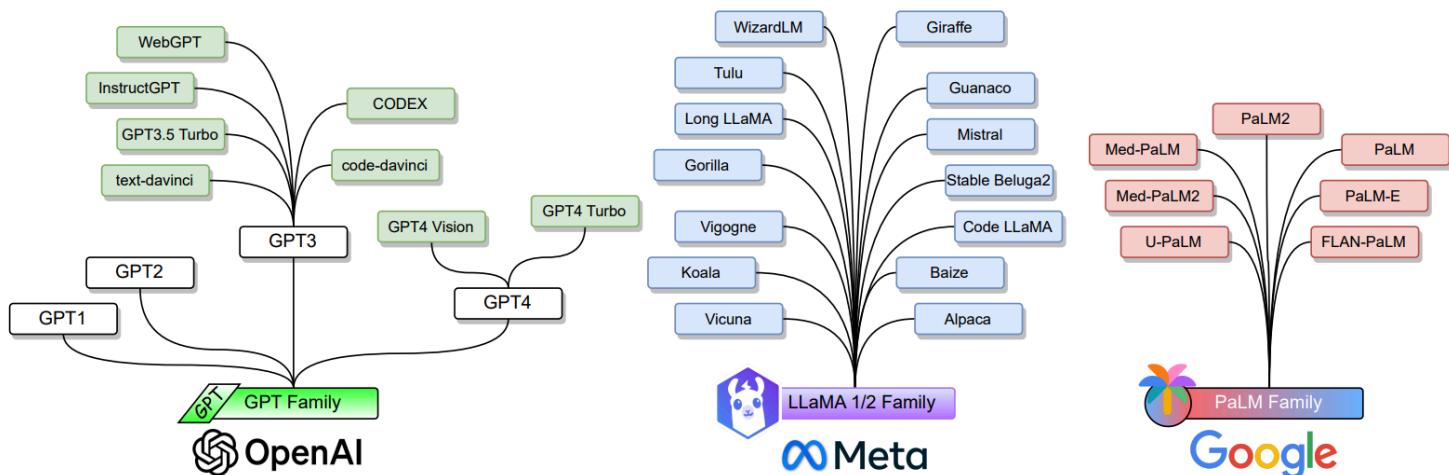
LANGUAGE MODEL SIZES TO MAR/2023



LifeArchitect.ai/models

Bảng 2: Thông số các mô hình ngôn ngữ nổi tiếng tính đến tháng 3 năm 2023

Các LLMs từ khoảng năm 2022 bắt đầu nổi lên và được chú ý nhiều hơn. Phần lớn đều dựa trên cấu trúc của Transformers để hình thành các pre-trained model. Có nhiều các công ty lớn tham gia vào cuộc đua xây dựng LLM model của riêng họ, nhằm cạnh tranh với GPT.



Hình 3: Các công ty lớn tham gia vào xây dựng LLMs và kết quả

1.3 LLM Chatbot

OPENAI (ChatGPT)

ChatGPT viết tắt của “Chat Generative Pre-training Transformer” là một công cụ trí tuệ nhân tạo được phát triển bởi OpenAI. Nó cung cấp một cách để tạo ra các cuộc trò chuyện tự động và thông minh. ChatGPT sử dụng một mô hình máy học sâu để học cách trả lời câu hỏi và tạo ra các cuộc trò chuyện tự động. Nó cũng có thể được sử dụng để tạo ra các hội thoại tự động và các ứng dụng trí tuệ nhân tạo khác. ChatGPT cũng có thể được sử dụng để tạo ra các hội thoại tự động và các ứng dụng trí tuệ nhân tạo khác. Nó cũng có thể được sử dụng để tạo ra các hội thoại tự động và các ứng dụng trí tuệ nhân tạo khác. ChatGPT cung cấp một cách để tạo ra các cuộc trò chuyện tự động và thông minh, và cũng có thể được sử dụng để tạo ra các ứng dụng trí tuệ nhân tạo khác. ChatGPT chính thức ra mắt vào 30/11/2022 và tạo nên cơn sốt thu hút trên toàn cầu.

1. **GPT-1 (2018)**: GPT đầu tiên được ra mắt vào năm 2018, đánh dấu bước đột phá trong việc sử dụng kiến trúc Transformer cho các tác vụ ngôn ngữ tự nhiên. GPT-1 sử dụng 117 triệu tham số và được huấn luyện trên một tập dữ liệu lớn từ internet.
2. **GPT-2 (2019)**: Tiếp nối thành công của GPT-1, GPT-2 được phát triển với 1.5 tỷ tham số. GPT-2 có khả năng tạo ra văn bản tự nhiên và mạch lạc, mở rộng khả năng của mô hình trong các tác vụ ngôn ngữ phức tạp hơn.
3. **GPT-3 (2020)**: GPT-3 đánh dấu bước nhảy vọt lớn với 175 tỷ tham số. Nó có khả năng hiểu và tạo văn bản với mức độ phức tạp và chi tiết cao hơn rất nhiều, mở ra nhiều ứng dụng thực tiễn trong các lĩnh vực khác nhau.
4. **GPT-4 (2023)**: GPT-4, nền tảng của ChatGPT hiện tại, tiếp tục cải thiện độ chính xác và khả năng hiểu ngữ cảnh. Với khả năng xử lý ngôn ngữ tự nhiên vượt trội, GPT-4 đã trở thành công cụ hữu ích trong nhiều lĩnh vực, từ hỗ trợ khách hàng đến sáng tạo nội dung.

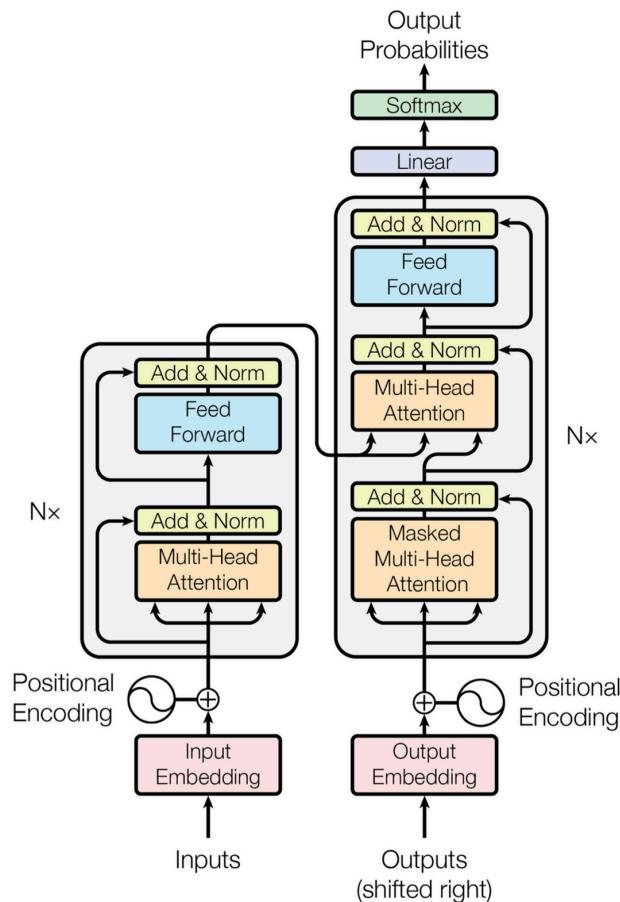
Cấu trúc của ChatGPT

ChatGPT được xây dựng trên kiến trúc Transformer, một mô hình mạng nơ-ron được giới thiệu bởi Vaswani và các cộng sự vào năm 2017. Transformer sử dụng cơ chế tự tập trung (self-attention) để xử lý thông tin, giúp mô hình hiểu ngữ cảnh và các mối quan hệ trong văn bản.

Các thành phần chính của Transformer:

- **Encoder**
 - **Self-Attention**
 - Cho phép mô hình tập trung vào các phần khác nhau của chuỗi đầu vào.
 - Tạo mối liên hệ giữa các thành phần khác nhau của chuỗi đầu vào

- **Feed-Forward Network**
 - Áp dụng một mạng kết nối đầy đủ (fully connected) lên từng vị trí trong chuỗi một cách độc lập.
- **Decoder**
 - **Masked Self-Attention:**
 - Giống Self-Attention nhưng có thêm mask để ngăn mô hình nhìn vào các thành phần phía trước của chuỗi đầu ra, từ đó mô hình phải dựa dự đoán
 - **Encoder-Decoder Attention:**
 - Sử dụng Result của Encoder kết hợp với Masked Self-Attention để tạo mối liên hệ giữa encoder và decoder
 - **Feed-Forward Network**
 - Áp dụng một mạng kết nối đầy đủ (fully connected) lên từng vị trí trong chuỗi một cách độc lập.



Hình 4: Mô hình Transformers

CHƯƠNG 2: KỸ THUẬT CRAG

2.1 Hiện tượng ảo giác trong các mô hình LLM

- Trong các mô hình ngôn ngữ lớn (LLM), "**ảo giác**" (**hallucination**) là hiện tượng khi mô hình tạo ra thông tin sai lệch, không có cơ sở trong dữ liệu huấn luyện hoặc dữ liệu đầu vào. Điều này có thể bao gồm việc bịa đặt nội dung, hiểu sai ngữ cảnh, hoặc tự mâu thuẫn. Hiện tượng này chủ yếu bắt nguồn từ dữ liệu huấn luyện không đồng nhất, thiếu cơ chế kiểm chứng thông tin, và cách mô hình dự đoán từ ngữ dựa trên xác suất. Đây là thách thức lớn khi áp dụng LLM vào các lĩnh vực yêu cầu độ chính xác cao.

Hiện tượng "ảo giác" trong các mô hình ngôn ngữ lớn (LLM) được phân loại thành "ảo giác về tính chính xác" và "ảo giác về tính trung thành".

2.1.1 Ảo giác về tính chính xác

Ảo giác về tính chính xác trong các mô hình ngôn ngữ lớn (LLM) là hiện tượng mô hình cung cấp thông tin nghe có vẻ đúng và đáng tin cậy, nhưng thực tế lại sai lệch hoặc không có cơ sở. Điều này thường khiến người dùng nhầm lẫn và đánh giá sai về độ chính xác của mô hình. Hiện tượng này xuất phát từ khả năng ngôn ngữ tự nhiên của LLM, giúp tạo ra nội dung trôi chảy nhưng không đảm bảo tính xác thực do thiếu cơ chế kiểm chứng thông tin.

Bảng dưới phân loại các loại "ảo giác về tính chính xác" trong các mô hình ngôn ngữ lớn (LLM) cùng với ví dụ:

Phân loại	Đầu vào của User	Đầu ra của Model	Giải thích
Mâu thuẫn thực tế	Hãy cho tôi biết ai là người đầu tiên bay ra ngoài vũ trụ.	Neil Armstrong là người đầu tiên bay ra ngoài vũ trụ.	Câu trả lời của LLM mâu thuẫn với thực tế vì Neil Armstrong là người đầu tiên bước lên mặt trăng, Yuri Gagarin là người đầu tiên bay ra ngoài vũ trụ .
Bịa đặt thực tế	Ai là người phát minh ra tàu vũ trụ Apollo 11?	Tàu vũ trụ Apollo 11 được phát minh bởi nhà khoa học không gian nổi tiếng	Phản hồi của LLM là một tuyên bố bịa đặt vì Apollo 11 không phải là phát minh

		<p style="color: red;">Jonathan McKenzie vào năm 1965.</p>	<p>của một cá nhân cụ thể, mà là thành quả của hàng nghìn kỹ sư, nhà khoa học làm việc trong dự án của NASA.</p> <p>"Jonathan McKenzie" có thể là một cái tên do mô hình tự bịa ra, vì không có người nổi tiếng nào với vai trò đó trong lịch sử Apollo.</p>
--	--	--	---

2.1.2 Ảo giác về tính trung thành

Đây là những trường hợp khi mô hình tạo ra nội dung không trung thành hoặc không nhất quán với nội dung nguồn được cung cấp.

Bảng này phân loại các loại ảo giác về tính trung thành thành ba loại:

- Mâu thuẫn hướng dẫn: LLM bỏ qua các hướng dẫn cụ thể được người dùng đưa ra.
- Mâu thuẫn ngữ cảnh: Đầu ra của mô hình bao gồm thông tin không có trong ngữ cảnh được cung cấp hoặc mâu thuẫn với nó.
- Mâu thuẫn logic: Đầu ra của mô hình chứa một lỗi logic mặc dù ban đầu là đúng.

Phân loại	Đầu vào của User	Đầu ra của Model	Giải thích
Mâu thuẫn hướng dẫn	Dịch câu hỏi sau đây sang tiếng Việt: "Thủ đô của Mỹ là gì?"	Thủ đô của Mỹ là Washington D.C.	Người dùng đã cung cấp rõ ràng một hướng dẫn để dịch câu hỏi sang tiếng Việt. Thay vì dịch như được yêu cầu, LLM đã chọn trả lời câu hỏi.
Mâu thuẫn ngữ cảnh	Tóm tắt đoạn văn sau về sông Amazon: Amazon là	Sông Amazon bắt nguồn từ Brazil và chảy ra Đại Tây	Đoạn văn gốc chỉ rõ rằng sông Amazon bắt nguồn từ dãy

	một con sông lớn ở Nam Mỹ, được coi là con sông có lưu vực lớn nhất thế giới. Nó bắt nguồn từ dãy Andes ở Peru, chảy qua nhiều quốc gia trước khi đổ ra Đại Tây Dương	Dương.	Andes ở Peru , nhưng mô hình lại tóm tắt thành bắt nguồn từ Brazil , vốn là một quốc gia mà sông Amazon đi qua, không phải nơi nó bắt đầu.
Mẫu thuẫn logic	Giải phương trình $2x + 9 = 11$ từng bước một.	Bước 1: Trừ 9 ở cả hai vế được: $2x = 2$. Bước 2: Chia cả hai vế cho 2 được $x = 2$. Vậy kết quả của phương trình là $x = 2$.	Bước đầu tiên đúng khi cô lập biến x để được $2x = 2$. Tuy nhiên, bước tiếp theo lại chia 2 cho 2 một cách không chính xác để có kết quả $x = 2$, điều này không nhất quán với lý luận trước đó.

2.1.3 Nguyên nhân gây ra ảo giác trong LLM

1. Nguồn dữ liệu sai sót

- Huấn luyện trên dữ liệu sai có thể dẫn đến các thông tin sai lệch.

Ví dụ: Một LLM trích dẫn Robert Oppenheimer là nhà phát minh duy nhất của bom nguyên tử do thông tin sai lệch lặp đi lặp lại trong dữ liệu huấn luyện.

2. Giới hạn kiến thức

- Thiếu thông tin cập nhật dẫn đến giới hạn trong các lĩnh vực chuyên biệt.

Ví dụ: Một LLM cung cấp thông tin cũ về các quốc gia đăng cai Olympic gần đây do kiến thức tĩnh từ dữ liệu huấn luyện.

3. Vấn đề trong quá trình huấn luyện

- Lỗi kiến trúc: Biểu diễn đơn hướng có thể hạn chế sự hiểu biết ngữ cảnh.

Ví dụ: Một LLM tạo ra một câu chuyện một chiều mà không xem xét tất cả ngữ cảnh, dẫn đến nội dung phiến diện hoặc thiên lệch.

- Thiên lệch phơi nhiễm: Sự khác biệt giữa huấn luyện và suy luận có thể gây ra các lỗi lan truyền.

Ví dụ: Trong quá trình suy luận, một LLM tiếp tục tạo ra các lỗi dựa trên một từ sai mà nó đã tạo ra.

4. Vấn đề căn chỉnh

- Căn chỉnh khả năng: Căn chỉnh LLM với các khả năng vượt quá dữ liệu huấn luyện có thể dẫn đến lỗi.

Ví dụ: Một LLM tạo ra nội dung trong một lĩnh vực chuyên biệt mà không có dữ liệu cần thiết, dẫn đến các thông tin bịa đặt.

- Căn chỉnh niềm tin: Các đầu ra lệch khỏi niềm tin nội bộ của LLM, dẫn đến sự không chính xác.

Ví dụ: Một LLM chiều lòng ý kiến người dùng, tạo ra nội dung mà nó "biết" là sai.

5. Vấn đề trong quá trình suy luận

- Vấn đề decoding: Sự ngẫu nhiên trong quá trình lấy mẫu có thể dẫn đến các đầu ra không thường xuyên nhưng không chính xác.

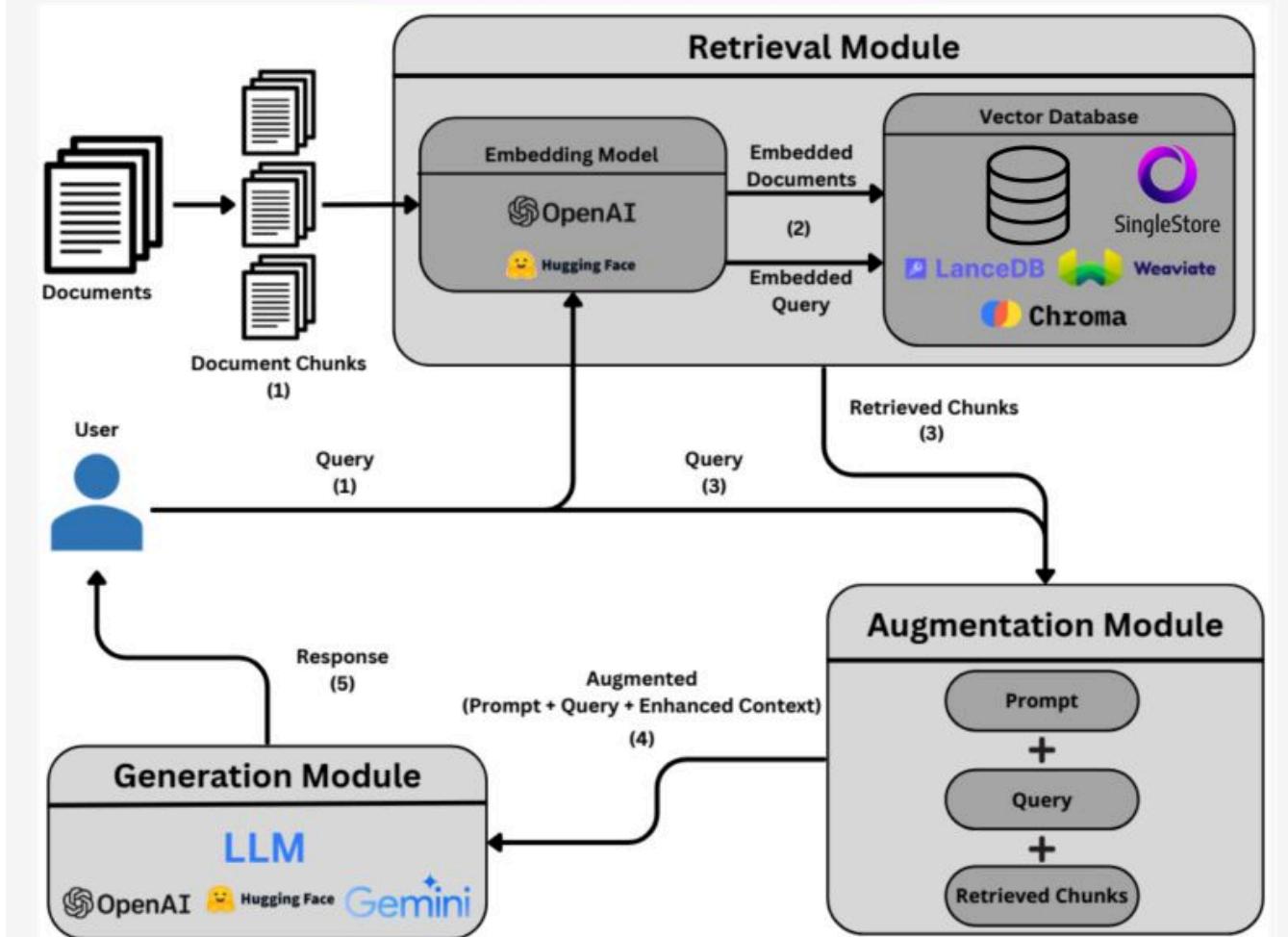
Ví dụ: Một LLM chọn các từ có xác suất thấp trong quá trình tạo ra, dẫn đến nội dung bất ngờ hoặc không liên quan.

- Biểu diễn giải mã không hoàn hảo: Quá phụ thuộc vào nội dung đã tạo ra một phần và giới hạn softmax.

Ví dụ: Một LLM tập trung quá nhiều vào các từ gần đây hoặc không nắm bắt được mối quan hệ từ phức tạp, dẫn đến các lỗi về độ trung thực.

2.2 Kỹ thuật RAG

Systematic RAG Workflow



Để khắc phục vấn đề ảo giác, một phương pháp ban đầu mang tên **Retrieval-Augmented Generation (RAG)** đã được phát triển. Phương pháp này hoạt động bằng cách tạo chỉ mục cho từng đoạn văn trong tài liệu. Khi nhận được một truy vấn, hệ thống sẽ truy xuất các đoạn văn liên quan nhất và cung cấp chúng cho mô hình ngôn ngữ lớn như ChatGPT hoặc GPT-4. Việc chỉ cung cấp các đoạn văn có chọn lọc, thay vì toàn bộ tài liệu, giúp tránh tình trạng quá tải thông tin cho mô hình và cải thiện đáng kể chất lượng phản hồi.

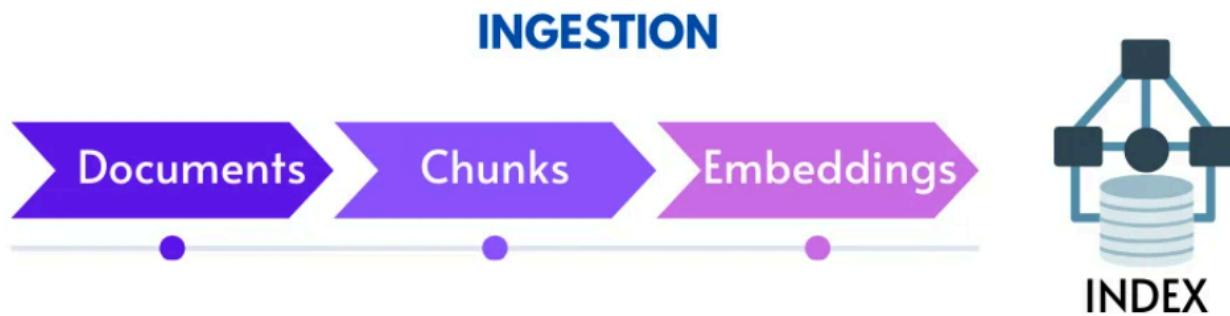
➤ **Tổng quan RAG có 3 bước:**

Bước 1: Ingestion (Biến đổi dữ liệu)

Bước 2: Retrieval (Truy xuất thông tin)

Bước 3: Synthesis/Response Generation (Tổng hợp và sinh câu trả lời)

2.2.1 Ingestion



Ingestion là quá trình biến đổi dữ liệu, gồm:

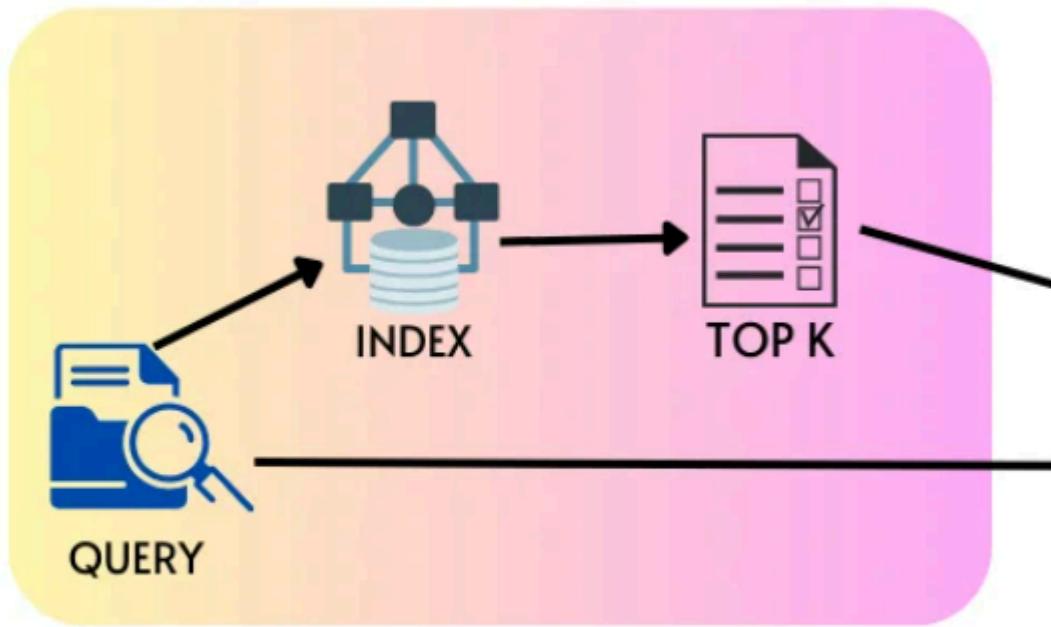
- Thu thập dữ liệu
- Tiền xử lý dữ liệu
- Lập chỉ mục và lưu trữ vào database

Chunking (phân đoạn): quá trình chia các prompts hoặc tài liệu thành các phần nhỏ hơn, gọi là **chunks** hoặc **segments**, sao cho mỗi phần vẫn mang ý nghĩa rõ ràng. Các chunks này thường được chia dựa trên kích thước cố định, chẳng hạn như số ký tự, câu hoặc đoạn văn cụ thể.

Embeddings: Cả câu truy vấn của user và các tài liệu trong kho tri thức được chuyển đổi thành một định dạng chung, cụ thể là không gian vector, nhằm so sánh mức độ liên quan giữa chúng.

Indexing: Các vector embedding được gán chỉ mục và lưu trữ trong cơ sở dữ liệu vector (vector database) để hỗ trợ việc tìm kiếm nhanh chóng và hiệu quả.

2.2.2 Retrieval



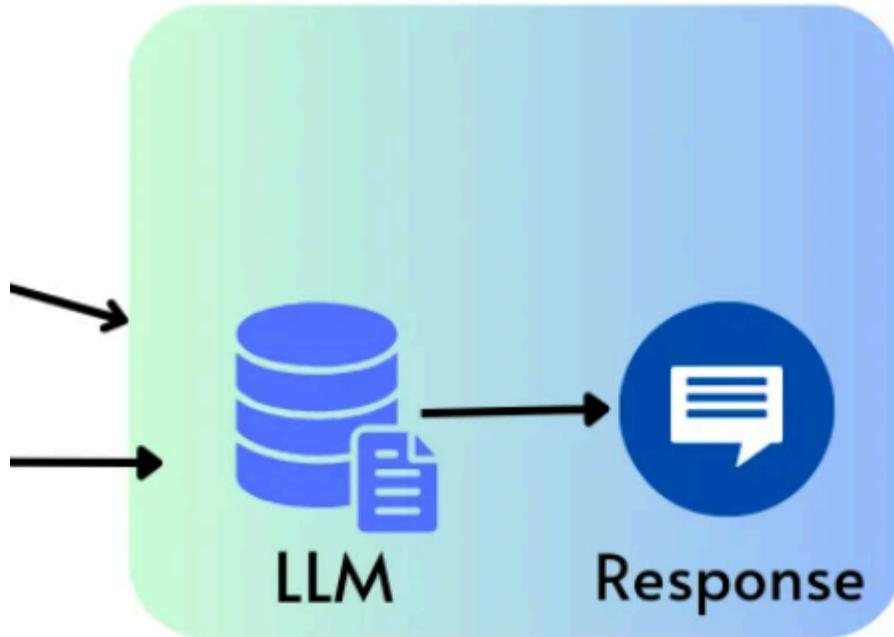
RETRIEVAL

Khi người dùng đặt một câu hỏi:

- Đầu vào (input) sẽ được chuyển đổi thành một vector (query vector) thông qua embedding model đã được huấn luyện ở phần trước.
- Query vector này sẽ được so khớp với tất cả các vectors trong cơ sở dữ liệu vector (vector database) để tìm ra những vector gần nhất, có khả năng chứa câu trả lời cho câu hỏi của người dùng.

2.2.3 Augmentation & Generation

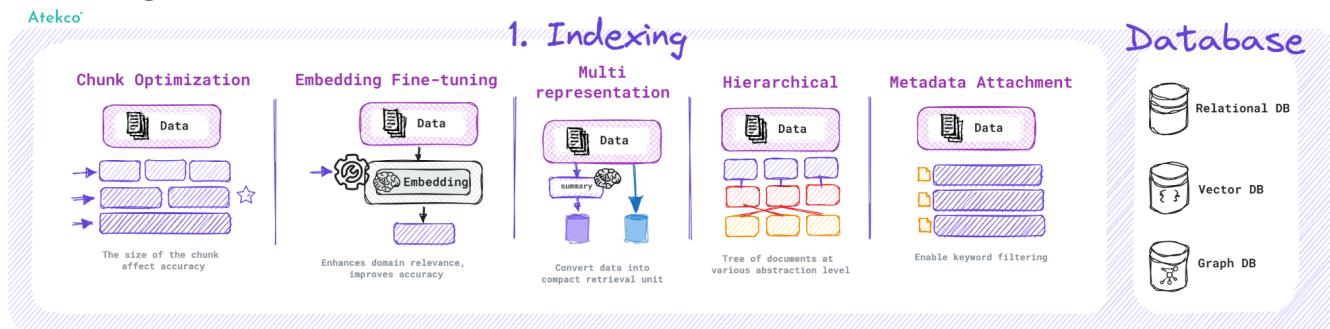
Mô hình LLM thu thập câu hỏi của người dùng cùng với các thông tin liên quan từ vector database. Quá trình này kết hợp câu hỏi và dữ liệu đã được nhận diện để tạo ra câu trả lời phù hợp.



SYNTHESIS

2.2.4 Một số vấn đề và kỹ thuật nâng cao

1. Indexing



- **Chunk Optimization:** Tối ưu hóa kích thước và cấu trúc của các đoạn văn bản (chunk) nhằm đảm bảo chúng không quá lớn cũng không quá nhỏ, giúp duy trì ngữ cảnh cần thiết mà không vượt quá giới hạn độ dài của mô hình ngôn ngữ lớn (LLM). Điều này giúp cải thiện hiệu quả xử lý và đảm bảo rằng mô hình có thể xử lý thông tin một cách chính xác mà không bị cắt xén hoặc mất mát nghĩa quan trọng.
- **Embedding Fine-tuning:** Tinh chỉnh embedding model giúp cải thiện khả năng hiểu nghĩa của dữ liệu được tạo chỉ mục, từ đó nâng cao khả năng khớp chính xác nội dung truy xuất với yêu cầu của người dùng. Việc này giúp mô hình hiểu rõ hơn về mối quan hệ giữa các đoạn văn bản, cải thiện chất lượng tìm kiếm và trả lời câu hỏi.
- **Multi-Representation:** Phương pháp này cho phép chuyển đổi tài liệu thành các đơn vị truy xuất nhỏ gọn, như tóm tắt nội dung, giúp cải thiện độ chính xác và tốc độ của quá

trình truy xuất khi người dùng yêu cầu thông tin cụ thể từ một tài liệu lớn. Việc này giúp giảm thiểu thời gian tìm kiếm và đảm bảo rằng người dùng nhận được câu trả lời chính xác và nhanh chóng.

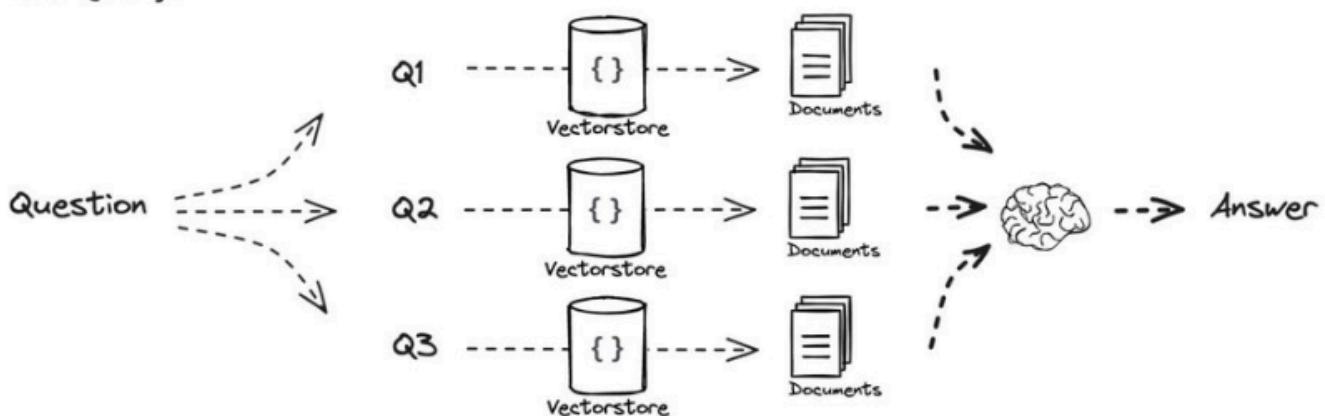
- **Hierarchical Indexing:** Áp dụng mô hình phân cấp như RAPTOR giúp tổ chức dữ liệu thành các cấp độ tổng hợp từ chi tiết đến tổng quát, từ đó cải thiện khả năng truy xuất thông tin dựa trên ngữ cảnh rộng hơn và chính xác hơn. Phương pháp này cho phép hệ thống lựa chọn các thông tin phù hợp ở các cấp độ khác nhau, giúp đáp ứng yêu cầu người dùng một cách nhanh chóng và hiệu quả, đồng thời tối ưu hóa quá trình tìm kiếm trong các tài liệu lớn.
- **Metadata Attachment:** Thêm **metadata** vào từng chunk hoặc dữ liệu giúp tăng cường khả năng phân tích và phân loại thông tin, từ đó cho phép truy xuất dữ liệu một cách có hệ thống hơn. Việc này giúp mô hình dễ dàng nhận diện và lựa chọn thông tin phù hợp trong nhiều tình huống cụ thể, nâng cao tính chính xác và hiệu quả khi người dùng yêu cầu thông tin từ kho dữ liệu.

2. Pre-retrieval

➤ **Vấn đề:** Truy vấn của người dùng là một thách thức trong RAG. Nếu người dùng đưa ra một truy vấn không rõ ràng, các tài liệu được truy xuất cũng sẽ trở nên mơ hồ nếu sử dụng phương pháp tìm kiếm dựa trên độ tương đồng khoảng cách.

Multi-Query: Sử dụng cách tiếp cận chia nhỏ để trị. Viết lại câu hỏi của người dùng từ nhiều góc độ khác nhau, sau đó truy xuất các tài liệu phù hợp cho mỗi phiên bản câu hỏi đã được thay đổi. Với mỗi câu hỏi đã viết lại, thu thập các tài liệu có mức độ liên quan cao nhất và kết hợp chúng lại thành một tài liệu duy nhất, sử dụng làm ngữ cảnh để trả lời câu hỏi.

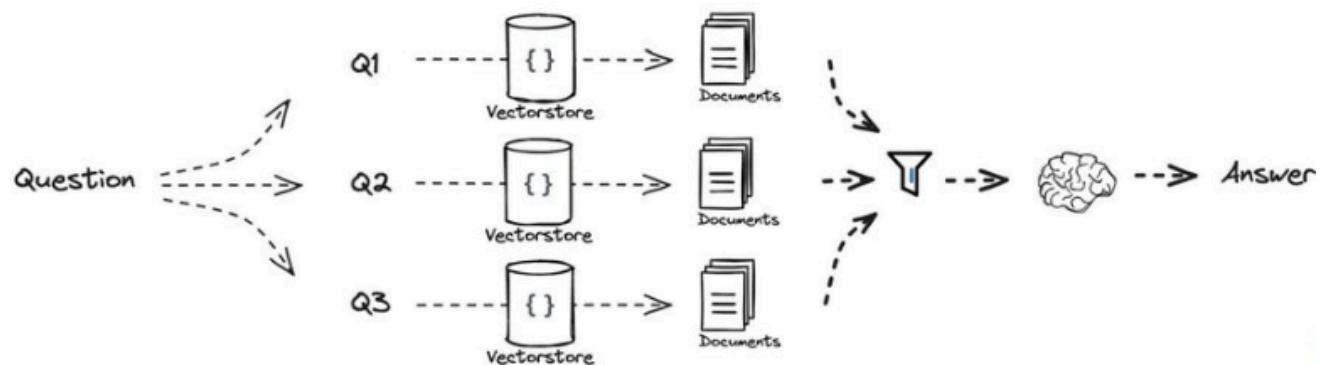
Multi Query:



➤ **Vấn đề:** Một truy vấn tìm kiếm đơn lẻ của người dùng có thể không phản ánh đầy đủ phạm vi mà người dùng quan tâm, hoặc không cung cấp được kết quả toàn diện.

Rag-Fusion: Xây dựng trên nền của Multi Query. Viết lại câu hỏi của người dùng từ nhiều góc độ khác nhau, truy xuất tài liệu cho mỗi phiên bản câu hỏi đã được thay đổi, và kết hợp thứ hạng của các danh sách kết quả tìm kiếm từ các truy vấn khác nhau để tạo ra một thứ hạng duy nhất và thống nhất bằng phương pháp **Reciprocal Rank Fusion (RRF)**.

RAG-Fusion:

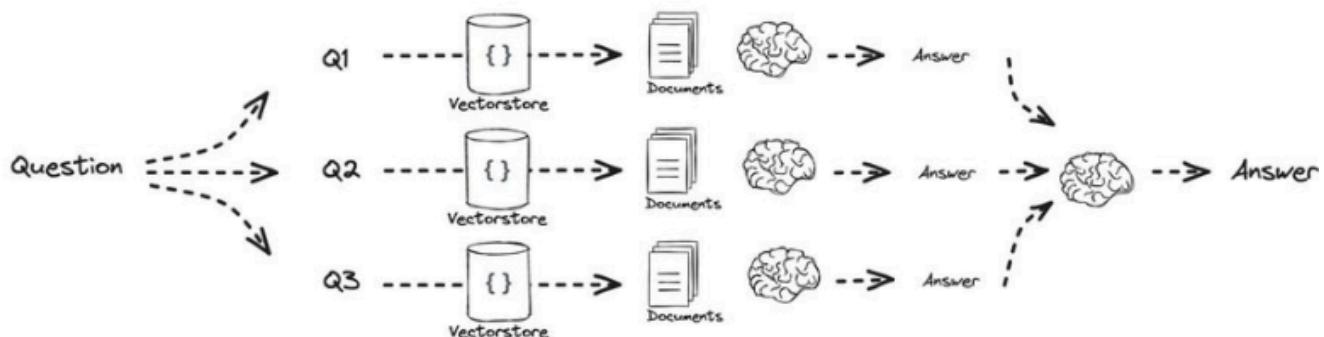


➤ **Vấn đề:** Một số câu hỏi phức tạp không thể được giải quyết hiệu quả chỉ với một bước truy xuất duy nhất, vì các thông tin nền tảng hoặc tiền đề cần thiết có thể không được thu thập đầy đủ.

Decomposition:

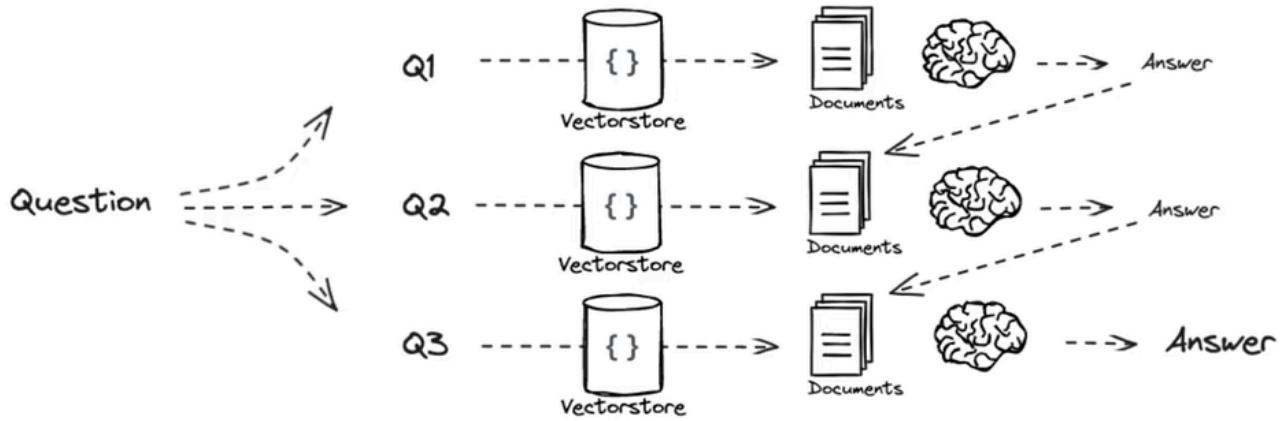
Phân tách một câu hỏi thành một tập hợp các vấn đề/câu hỏi nhỏ hơn, có thể được giải quyết song song (hợp nhất mỗi câu trả lời thành câu trả lời cuối cùng) như Least-to-Most prompting.

Decomposition:



IR-CoT:

Giải quyết tuần tự (sử dụng câu trả lời từ câu hỏi đầu tiên + truy xuất để trả lời câu hỏi thứ hai) (IR-CoT)



➤ **Vấn đề:** Một số câu hỏi phức tạp của người dùng cần kiến thức về các khái niệm cao cấp hoặc nguyên lý cơ bản để trả lời một cách hiệu quả nhất.

Step-back question:

Đầu tiên, yêu cầu LLM đặt một câu hỏi chung về các khái niệm hoặc nguyên tắc cấp cao, và truy xuất các thông tin liên quan về chúng. Sử dụng cơ sở này để giúp trả lời câu hỏi của người dùng.

Knowledge QA Final-Answer Prompt

You are an expert of world knowledge. I am going to ask you a question. Your response should be comprehensive and not contradicted with the following context if they are relevant. Otherwise, ignore them if they are not relevant.

<Passage from original retrieval augmentation>
<Passage from step-back retrieval augmentation>

Original Question: <Original Question>
Answer:

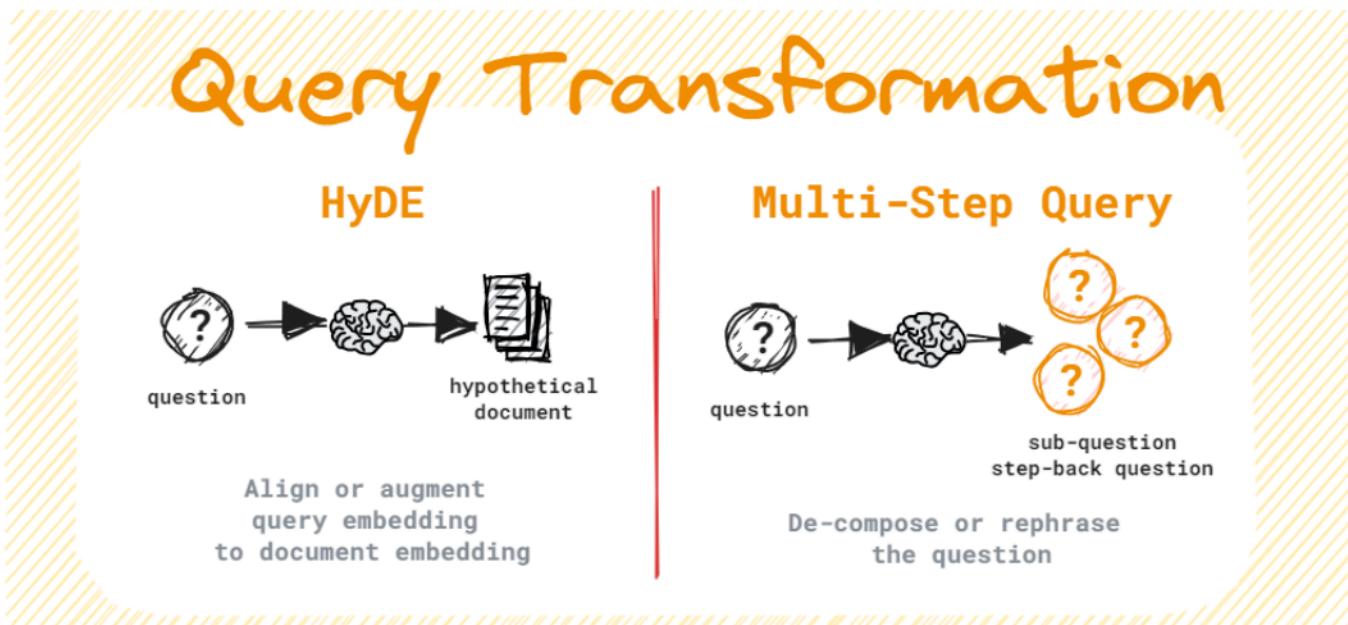
Table 12: Prompt of querying the model for final answer with additional contexts from original and step-back retrieval augmentations in TimeQA and SituatedQA

dataset	Original Question	Step-back Question
MuSiQue	at year saw the creation of the region where the county of Hertfordshire is located?	which region is the county of Hertfordshire located?
MuSiQue	Jan Šindel's was born in what country?	what is Jan Šindel's personal history?
MuSiQue	When was the abolishment of the studio that distributed The Game?	which studio distributed The Game?
MuSiQue	What city is the person who broadened the doctrine of philosophy of language from?	who broadened the doctrine of philosophy of language

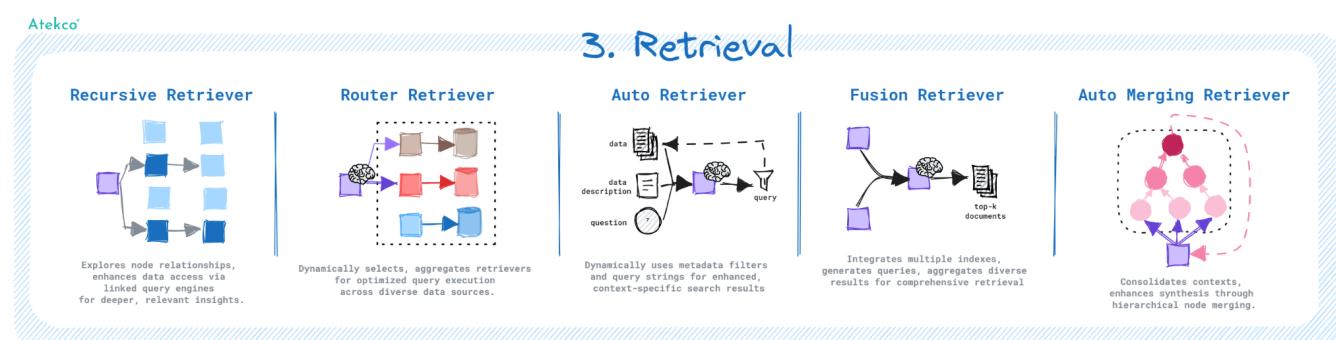
➤ **Vấn đề:** Các tài liệu thường rất khác biệt so với các câu hỏi đầu vào của người dùng. Tuy nhiên, nhiều phương pháp retrieval lại embed tài liệu và câu hỏi vào cùng một không gian embedding để tìm kiếm sự tương đồng.

HyDE - Hypothetical Document Embeddings:

Sử dụng một mô hình LLM để chuyển đổi câu hỏi thành các tài liệu giả định trả lời cho câu hỏi. Sau đó ta sử dụng các tài liệu giả định đã được embed để retrieve các tài liệu thực dựa trên giả thuyết rằng tìm kiếm sự tương đồng giữa các tài liệu với nhau có thể tạo ra các kết quả phù hợp hơn.

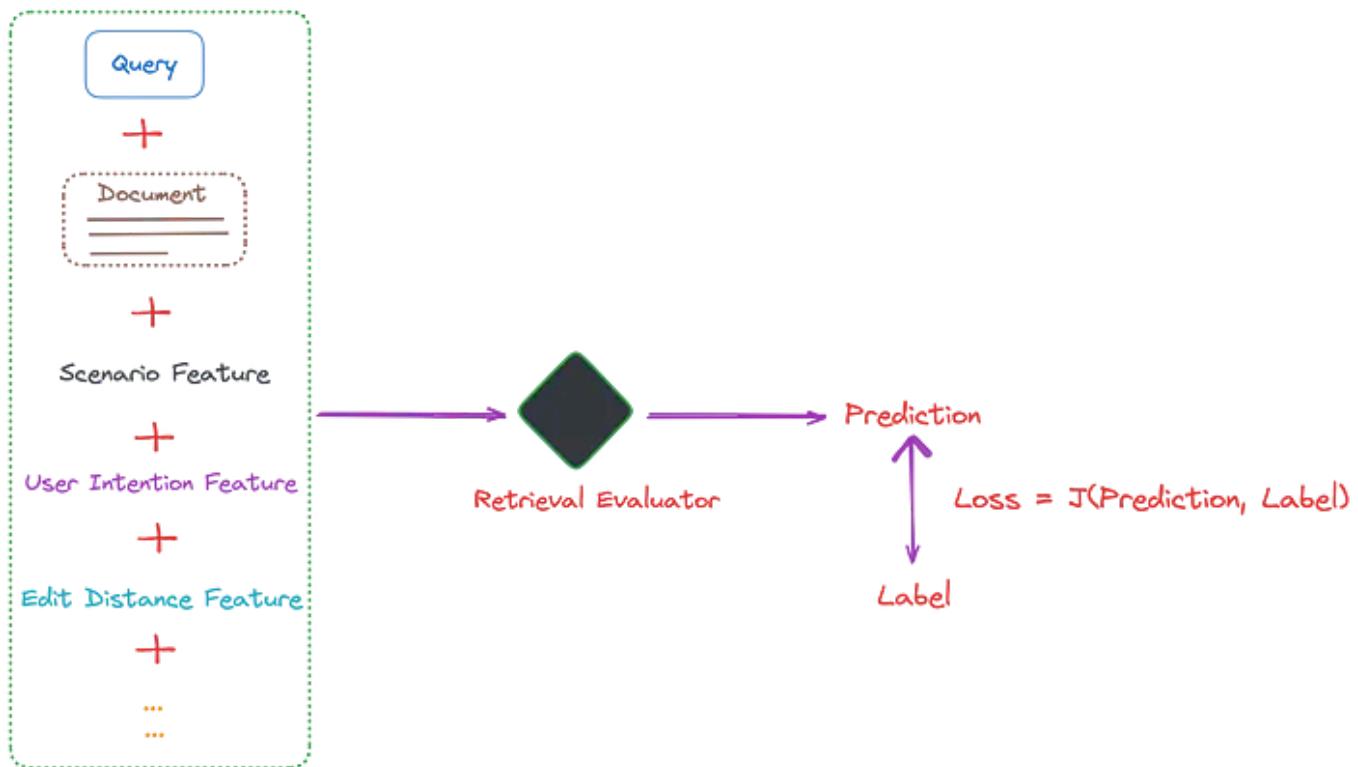


3. Retrieval



- **Recursive Retriever:** Là kỹ thuật cho phép truy xuất sâu vào các dữ liệu liên quan và thực hiện truy vấn thêm dữ liệu dựa trên kết quả truy vấn trước đó. Kỹ thuật này hữu ích trong các tình huống cần khám phá thông tin chi tiết hoặc chuyên sâu.
- **Router Retriever:** Là kỹ thuật sử dụng LLM để đưa ra quyết định động về nguồn dữ liệu hoặc công cụ truy vấn dữ liệu phù hợp cho mỗi truy vấn cụ thể, trong đó hệ thống đóng vai trò như một **router**, chọn lọc và định hướng các truy vấn (queries) đến những nguồn thông tin hoặc mô-đun cụ thể để tối ưu hóa quá trình truy xuất.

- **Auto Retriever:** Phương pháp tự động truy vấn cơ sở dữ liệu bằng cách sử dụng LLM để xác định metadata để thực hiện filter hoặc tạo câu truy vấn phù hợp để truy xuất.
- **Fusion Retriever:** Kết hợp kết quả từ nhiều truy vấn và index khác nhau, giúp tối ưu hóa việc truy xuất thông tin và đảm bảo kết quả thu được là toàn diện và không bị trùng lặp, mang lại cái nhìn đa chiều cho quá trình truy xuất.
- **Auto Merging Retriever:** Khi có nhiều phân đoạn dữ liệu con được truy vấn, kỹ thuật này sẽ chuyển chúng thành phân đoạn dữ liệu cha, cho phép tập hợp các ngữ cảnh nhỏ lẻ thành một ngữ cảnh lớn hơn, hỗ trợ quá trình tổng hợp thông tin. Kỹ thuật này giúp cải thiện độ liên quan và tính toàn vẹn của ngữ cảnh.



2.3 Corrective RAG

Corrective RAG là một sự cải tiến so với phiên bản RAG truyền thống. Hầu hết các phương pháp RAG truyền thống không xem xét đến tính liên quan của tài liệu với câu hỏi mà chỉ đơn giản là kết hợp các tài liệu đã tìm kiếm. Điều này có thể đưa ra các thông tin không liên quan, làm cản trở mô hình và có thể gây ra hiện tượng ảo giác.

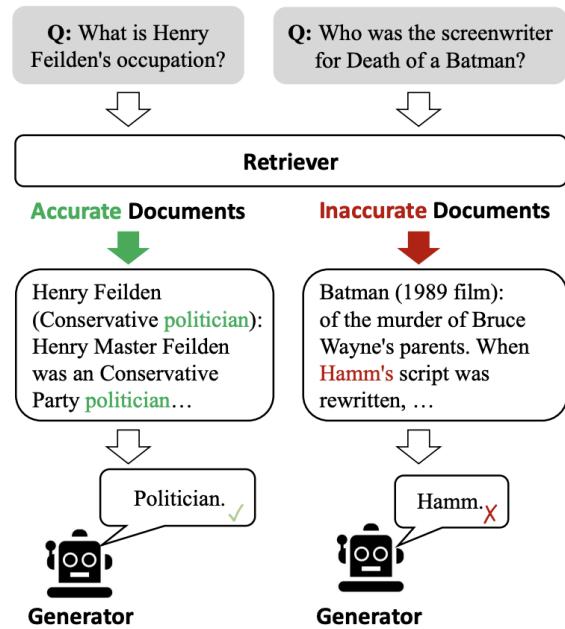
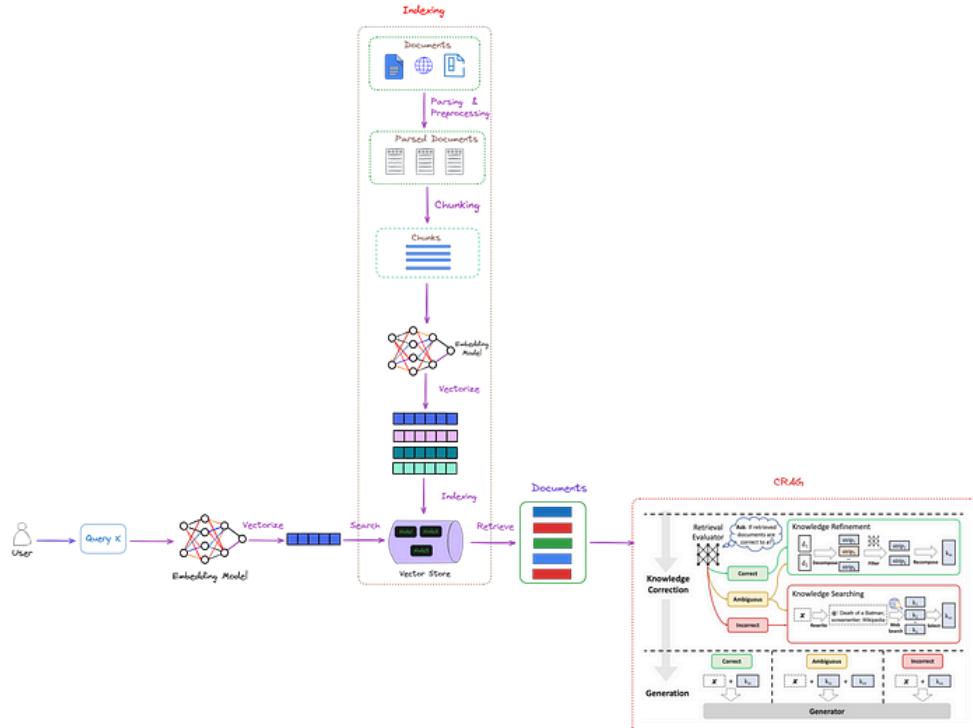


Figure 1: The examples show that a low-quality retriever is prone to introducing a substantial amount of irrelevant information, impeding the generators from acquiring accurate knowledge and potentially misleading them.

2.3.1 Ý tưởng

CRAG (Corrective Retrieval Augmented Generation) được thiết kế với một bộ đánh giá tìm kiếm gọn nhẹ để đánh giá chất lượng tổng thể của các tài liệu được truy xuất cho các truy vấn cụ thể. Sử dụng web search như một công cụ bổ sung để cải thiện kết quả truy xuất.



CRAG cải tiến RAG truyền thống bằng cách sử dụng một bộ đánh giá tìm kiếm để đánh giá mối quan hệ giữa các tài liệu được truy xuất và truy vấn.

Khi đánh giá, có 3 kết quả có thể xảy ra:

- Đúng: tài liệu được truy xuất chứa nội dung cần thiết theo yêu cầu truy vấn, sau đó sử dụng một thuật toán tinh chỉnh kiến thức để viết lại các tài liệu được truy xuất.
- Sai: Các tài liệu được truy xuất không liên quan đến yêu cầu truy vấn. Khi đó, sử dụng một công cụ tìm kiếm để tìm dữ liệu trên internet thay thế.
- Mơ hồ: Các tài liệu truy xuất có thể gần nhau nhưng không đủ để cung cấp một câu trả lời. Trong trường hợp này, thông tin bổ sung cần được lấy thông qua tìm kiếm web. Vì vậy, cả thuật toán tinh chỉnh kiến thức và công cụ tìm kiếm được sử dụng.

Cuối cùng, thông tin đã được xử lý được chuyển tiếp đến LLM để tạo phản hồi. Hình dưới mô tả chính xác thuật toán mà chúng ta sẽ sử dụng.

Algorithm 1: CRAG Inference

Require : E (Retrieval Evaluator), W (Query Rewriter), G (Generator)

Input : x (Input question), $D = \{d_1, d_2, \dots, d_k\}$ (Retrieved documents)

Output : y (Generated response)

1 $score_i = E$ evaluates the relevance of each pair (x, d_i) , $d_i \in D$

2 **Confidence** = Calculate and give a final judgment based on $\{score_1, score_2, \dots, score_k\}$
// **Confidence** has 3 optional values: [CORRECT], [INCORRECT] or [AMBIGUOUS]

3 **if** *Confidence* == [CORRECT] **then**

4 Internal_Knowledge = Knowledge_Refine(x, D)

5 k = Internal_Knowledge

6 **else if** *Confidence* == [INCORRECT] **then**

7 External_Knowledge = Web_Search(W Rewrites x for searching)

8 k = External_Knowledge

9 **else if** *Confidence* == [AMBIGUOUS] **then**

10 Internal_Knowledge = Knowledge_Refine(x, D)

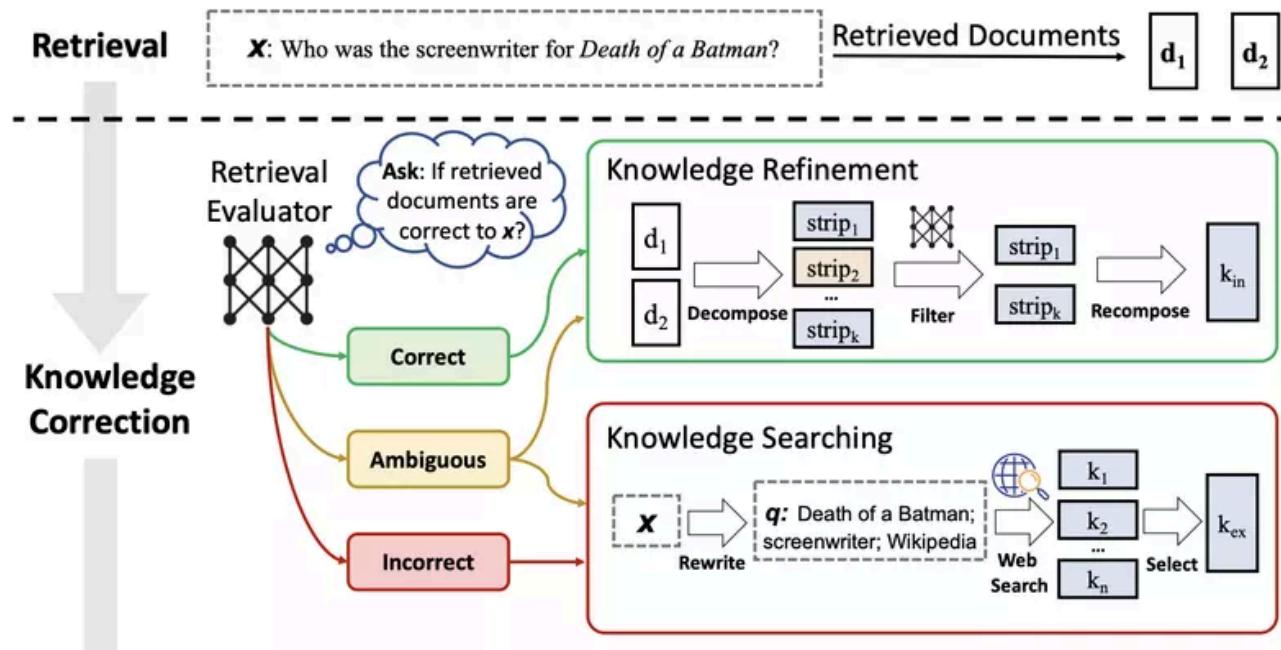
11 External_Knowledge = Web_Search(W Rewrites x for searching)

12 k = Internal_Knowledge + External_Knowledge

13 **end**

14 G predicts y given x and k

2.3.2 Triển khai



CRAg sử dụng một mô hình **T5-large** làm bộ đánh giá tìm kiếm. Đối với mỗi truy vấn, thông thường 10 tài liệu được truy xuất. Sau đó, truy vấn được nối tiếp với mỗi tài liệu một cách riêng biệt để làm đầu vào dự đoán mức độ liên quan của chúng. Trong quá trình tinh chỉnh, gán nhãn 1 cho các mẫu tích cực và -1 cho các mẫu tiêu cực. Trong quá trình suy luận, bộ đánh giá gán điểm liên quan cho mỗi tài liệu, dao động từ -1 đến 1.

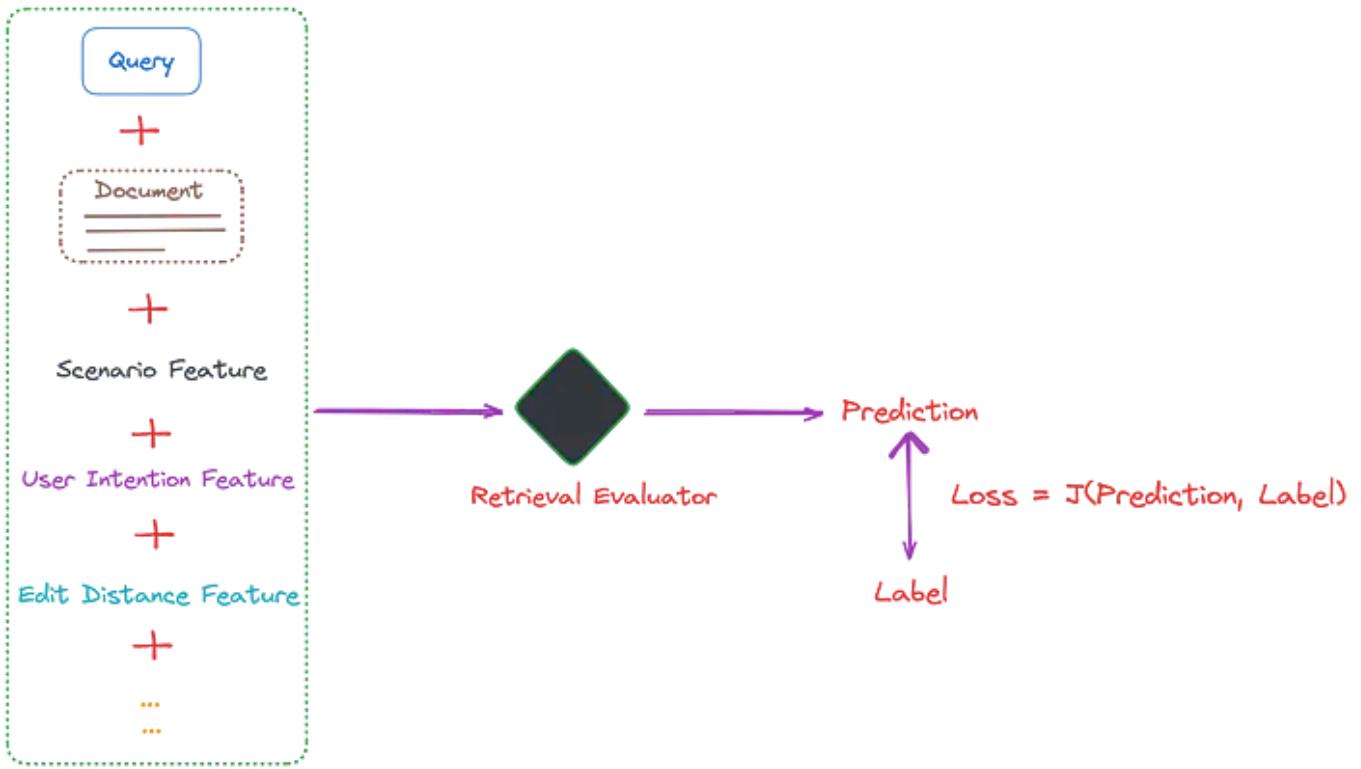
CRAg đã phát triển một phương pháp "**phân rã rồi tái cấu trúc**" (**decompose-then-recompose**) để trích xuất kiến thức từ các tài liệu có liên quan. Quy trình này nhằm mục đích rút ra những kiến thức quan trọng nhất từ tài liệu.

Đầu tiên, các quy tắc heuristic được áp dụng để phân tách từng tài liệu thành các dải kiến thức, với mục tiêu thu được kết quả chi tiết.

- Nếu tài liệu được truy xuất chỉ gồm một hoặc hai câu, nó được xem là một đơn vị độc lập.
- Nếu không, tài liệu được chia thành các đơn vị nhỏ hơn, thường bao gồm vài câu, tùy thuộc vào tổng độ dài của tài liệu. Mỗi đơn vị được kỳ vọng chứa một phần thông tin độc lập

Tiếp theo, bộ đánh giá tìm kiếm được sử dụng để tính toán similarity score cho mỗi dải kiến thức. Những dải có điểm liên quan thấp sẽ được lọc bỏ. Các dải kiến thức còn lại có liên quan sau đó được tái tổ hợp nhằm hình thành kiến thức nội bộ.

2.3.3 Cải tiến Bộ truy xuất:



- Bộ đánh giá truy xuất có thể được xem như một mô hình phân loại điểm số. Mô hình này được sử dụng để xác định mức độ liên quan của truy vấn và tài liệu, tương tự như mô hình reranking trong RAG.
- Các mô hình đánh giá liên quan này có thể được cải tiến bằng cách tích hợp thêm nhiều đặc điểm phù hợp với các tình huống thực tế.
- Bằng cách thêm các scenario feature vào dữ liệu đào tạo của bộ đánh giá truy xuất, nó có thể đánh giá tốt hơn mức độ liên quan của các tài liệu được truy xuất. Các đặc điểm khác, như ý định người dùng (user intention) và khoảng cách chỉnh sửa (edit distance) cũng có thể được tích hợp.

Điểm số và Ngưỡng (threshold) của Bộ đánh giá truy xuất:

- Ngưỡng là khác nhau cho các loại dữ liệu khác nhau. Ngoài ra, ta thấy rằng các ngưỡng cho các trường hợp mơ hồ và không chính xác thường xung quanh -0.9, cho thấy hầu hết kiến thức được truy xuất liên quan đến truy vấn. Có thể không nên loại bỏ hoàn toàn kiến thức này và chỉ dựa vào tìm kiếm web.
- Trong các ứng dụng thực tế, chúng ta cần điều chỉnh các tham số trên theo bài toán cụ thể và nhu cầu thực tế.

2.3.4 Đánh giá

Kết quả CRAG so với các phiên bản khác của RAG.

Method	PopQA (Accuracy)	Bio (FactScore)	Pub (Accuracy)	ARC (Accuracy)
<i>LMs trained with propriety data</i>				
LLaMA2-c _{13B}	20.0	55.9	49.4	38.4
Ret-LLaMA2-c _{13B}	51.8	79.9	52.1	37.9
ChatGPT	29.3	71.8	70.1	75.3
Ret-ChatGPT	50.8	-	54.7	75.3
Perplexity.ai	-	71.2	-	-
<i>Baselines without retrieval</i>				
LLaMA2 _{7B}	14.7	44.5	34.2	21.8
Alpaca _{7B}	23.6	45.8	49.8	45.0
LLaMA2 _{13B}	14.7	53.4	29.4	29.4
Alpaca _{13B}	24.4	50.2	55.5	54.9
CoVE _{65B}	-	71.2	-	-
<i>Baselines with retrieval</i>				
LLaMA2 _{7B}	38.2	78.0	30.0	48.0
Alpaca _{7B}	46.7	76.6	40.2	48.0
SAIL	-	-	69.2	48.4
LLaMA2 _{13B}	45.7	77.5	30.2	26.0
Alpaca _{13B}	46.1	77.7	51.1	57.6
<i>LLaMA2-hf-7b</i>				
RAG	50.5	44.9	48.9	43.4
CRAG	54.9	47.7	59.5	53.7
Self-RAG*	29.0	32.2	0.7	23.9
Self-CRAG	49.0	69.1	0.6	27.9
<i>SelfRAG-LLaMA2-7b</i>				
RAG	52.8	59.2	39.0	53.2
CRAG	59.8	74.1	75.6	68.6
Self-RAG	54.9	81.2	72.4	67.3
Self-CRAG	61.8	86.2	74.8	67.2

CHƯƠNG 3: Elasticsearch và Langchain

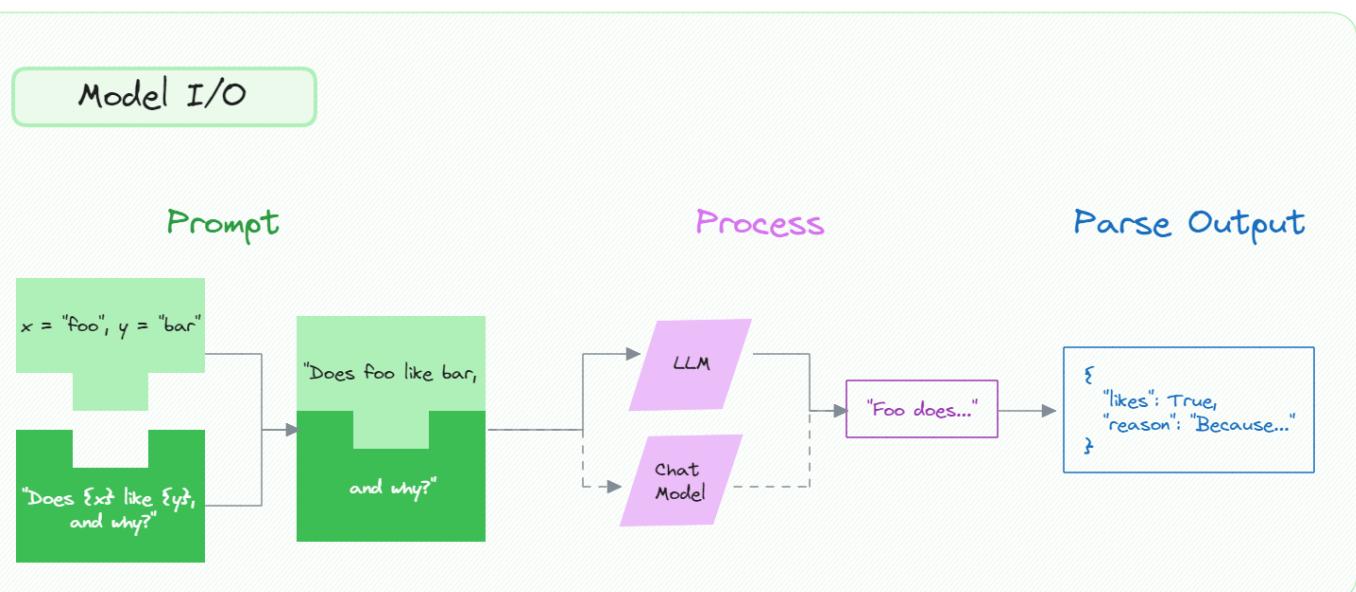
3.1 LangChain

LangChain là công cụ được tạo ra nhằm giúp xâu chuỗi, điều phối và tương tác có cấu trúc với các LLM. So với các phương pháp truyền thống, việc sử dụng LangChain giúp tích hợp LLM vào hệ thống dễ dàng, cho phép tương tác động với nhiều nguồn dữ liệu khác nhau để mang lại trải nghiệm phong phú, thay vì chỉ gọi API tiêu chuẩn. Bên cạnh đó, LangChain còn giúp developer xây dựng các agent có khả năng suy luận và chia nhỏ vấn đề, đưa context và bộ nhớ vào quá trình xử lý task,...

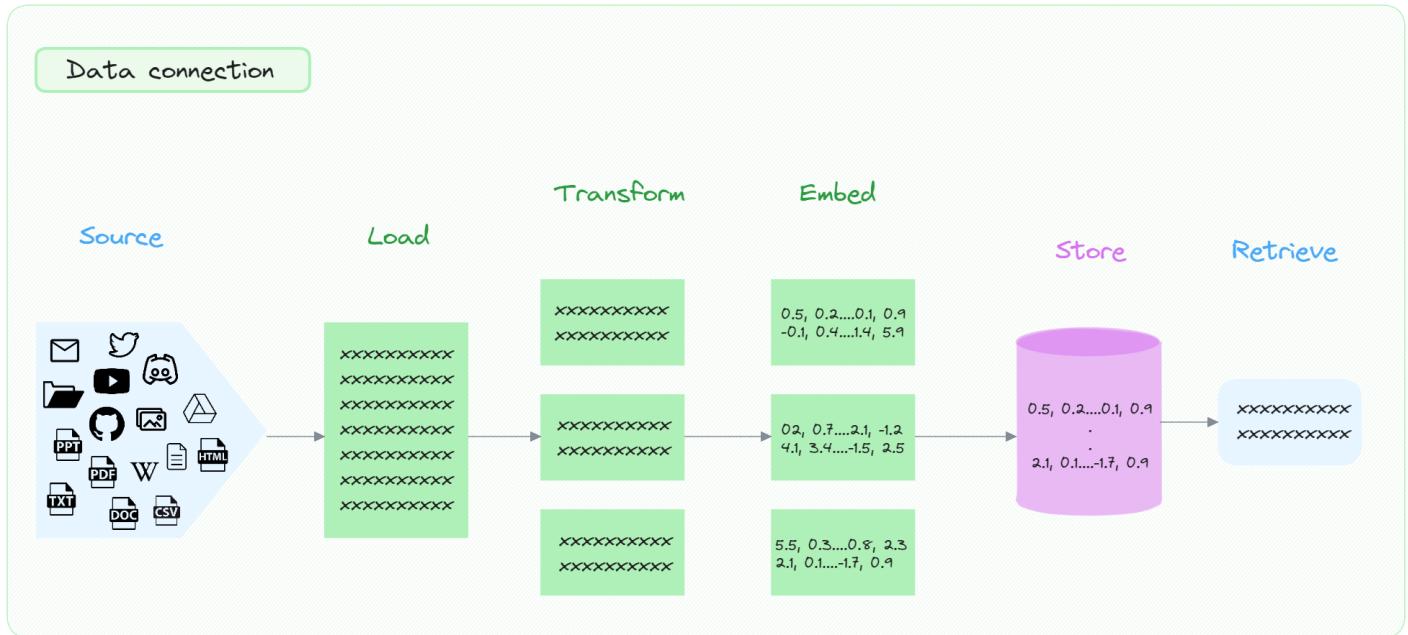
LangChain framework cung cấp các module cho phép bạn quản lý nhiều khía cạnh cụ thể trong tương tác với LLM. Các module này bao gồm:

- **Model I/O:** là thành phần nền tảng để giao tiếp với các language model. Module này cung cấp giao diện để thực hiện tương tác liền mạch với bất kỳ language model nào. Trong đó:
 - Đầu vào được định nghĩa bằng Prompts - tập hợp các hướng dẫn hay input mà user cung cấp giúp language model hiểu được ngữ cảnh và tạo câu trả lời mạch lạc.
 - Phần xử lý là language model được LangChain hỗ trợ tích hợp với 2 loại chính là LLMs (nhận một chuỗi đầu vào → trả chuỗi kết quả) và Chat models (nhận danh sách Chat Message → trả một Chat Message).
 - Ngoài ra, module này còn kết hợp các output parser để chuyển đổi kết quả đầu ra từ văn bản thô thành thông tin có tổ chức và có cấu trúc.
 - Chỉ với module Model I/O, người dùng đã có thể thực hiện tương tác và tích hợp LLM vào ứng dụng của mình một cách đơn giản và thuận tiện.

Chi với module Model I/O, người dùng đã có thể thực hiện tương tác và tích hợp LLM vào ứng dụng của mình một cách đơn giản và thuận tiện.



- **Retrieval:** là thành phần giúp tạo điều kiện thuận lợi cho việc tích hợp dữ liệu chuyên biệt của người dùng vào giai đoạn generation của language model bằng cách sử dụng Retrieval Augmented Generation (RAG). Langchain hỗ trợ các chức năng như document loaders, document transformers, text embedding models và nhiều thuật toán truy xuất cũng như cách lưu trữ dữ liệu dưới dạng vector, đảm bảo việc sử dụng dữ liệu hiệu quả và phù hợp ngữ cảnh.



- **Chains:** là thành phần giúp thiết kế những xử lý phức tạp bằng cách kết nối các LLM với nhau hoặc với các thành phần khác, bao gồm các chuỗi khác. Cách tiếp cận này vừa đơn giản vừa hiệu quả, giúp phát triển các ứng dụng phức tạp, nâng cao khả năng bảo trì. Một minh họa chuỗi đơn giản như: chuỗi lấy thông tin đầu vào của người dùng, chuyển thành định dạng của PromptTemplate, chuyển đến xử lý với LLM, và tổng hợp lại kết quả.
- **Agents:** là thành phần giúp ứng dụng sử dụng language model làm công cụ suy luận để quyết định thứ tự của hành động một cách linh hoạt. LangChain cung cấp nhiều loại Agents kết hợp với đa dạng Tools giúp agent tương tác với các tiện ích chung, với Chains hay với các agent khác, tạo thành một framework mạnh mẽ để xử lý các tác vụ phức tạp.
- **Memory:** là thành phần đóng vai trò then chốt đối với các ứng dụng có giao diện đối thoại, cho phép đưa luồng hội thoại trước đó vào LLM, giúp duy trì ngữ cảnh tương tác của người dùng. LangChain cung cấp rất nhiều tiện ích để tích hợp memory vào

hệ thống, hỗ trợ các hoạt động cơ bản như đọc và viết, đảm bảo hệ thống có thể truy cập vào các tin nhắn trong quá khứ hoặc duy trì cập nhật liên tục.

- **Callbacks:** là thành phần giúp đăng ký các event để khai thác các giai đoạn khác nhau của LLM, hữu ích trong các tác vụ như logging, monitoring, streaming,... Các event này được kích hoạt bởi các CallbackHandler tương ứng với từng sự kiện đã đăng ký. Ngoài ra, LangChain cũng cung cấp một số built-in handler như StdOutCallbackHandler giúp ghi lại tất cả sự kiện trong quá trình thực thi.

3.2 ElasticSearch

Elasticsearch là một công cụ tìm kiếm và phân tích dữ liệu mã nguồn mở được xây dựng trên nền tảng Apache Lucene. Công nghệ cung cấp khả năng tìm kiếm nhanh chóng và phân tích dữ liệu phức tạp thông qua giao diện RESTful và JSON.

Elasticsearch được sử dụng rộng rãi trong các ứng dụng tìm kiếm, phân tích log và giám sát hệ thống. Nền tảng có thể xử lý lượng dữ liệu lớn và cung cấp khả năng mở rộng tốt, cho phép người dùng lưu trữ, tìm kiếm, phân tích dữ liệu một cách hiệu quả theo thời gian thực.

3.2.1 Xác định đối tượng sử dụng Elasticsearch

- Các doanh nghiệp và tổ chức

Các doanh nghiệp sử dụng Elasticsearch để tìm kiếm và phân tích dữ liệu trong ứng dụng web, di động hoặc nội bộ. Nền tảng cũng được sử dụng để xây dựng các hệ thống phân tích dữ liệu thời gian thực.

- Nhà phát triển phần mềm

Elasticsearch là một công cụ mạnh mẽ cho các nhà phát triển phần mềm để xây dựng các ứng dụng tìm kiếm phức tạp, hệ thống phân tích log và ứng dụng xử lý dữ liệu lớn.

- Nhà quản trị hệ thống và mạng

Elasticsearch được sử dụng để giám sát và phân tích log từ hệ thống và mạng. Công nghệ giúp quản trị viên hệ thống dễ dàng tìm ra thông tin quan trọng và sự cố.

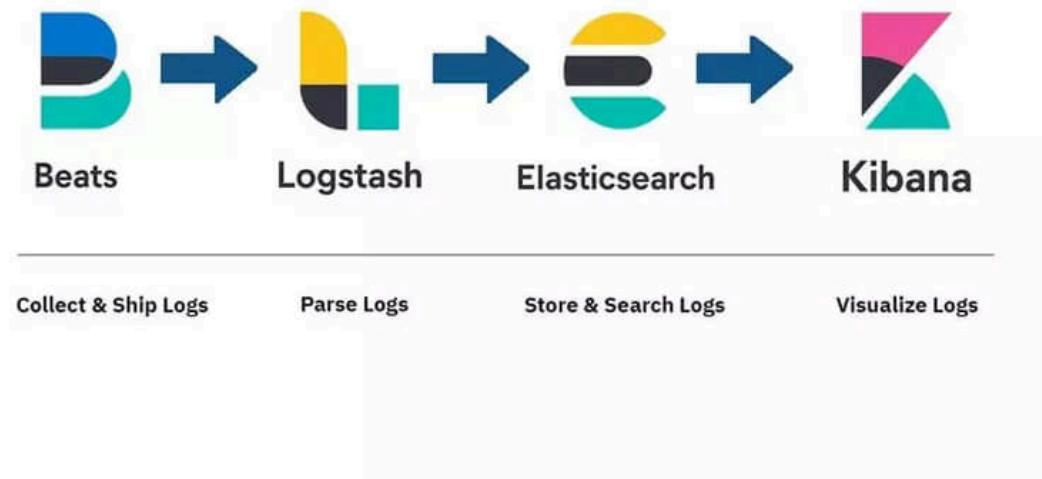
- Khoa học dữ liệu và phân tích

Các chuyên gia khoa học dữ liệu và phân tích sử dụng Elasticsearch để tìm kiếm thông tin, thực hiện phân tích và trực quan hóa dữ liệu.

- Cơ sở hạ tầng và hoạch định nguồn nhân lực
Elasticsearch cũng được sử dụng trong việc tìm kiếm thông tin về cơ sở hạ tầng, quản lý nguồn nhân lực và dự án.
- Ngành quảng cáo và tiếp thị
Trong lĩnh vực quảng cáo và tiếp thị, người dùng Elasticsearch để tìm kiếm, lọc, phân tích dữ liệu khách hàng, lưu lượng truy cập và khách hàng tiềm năng.

3.2.2 Nguyên tắc hoạt động của Elasticsearch

Elastic stack bao gồm logstash, elasticsearch và kibana. Logstash chịu trách nhiệm thu thập tất cả dữ liệu thô và xử lý dữ liệu trước khi lập chỉ mục và lưu trữ nó trong elasticsearch. Sau khi được lập chỉ mục, chúng ta có thể chạy các truy vấn phức tạp đối với dữ liệu của họ và sử dụng các tập hợp để truy xuất các bản tóm tắt phức tạp về dữ liệu của họ. Từ Kibana, người dùng có thể tạo trực quan hóa dữ liệu mạnh mẽ, chia sẻ bảng thông tin và quản lý Elastic Stack

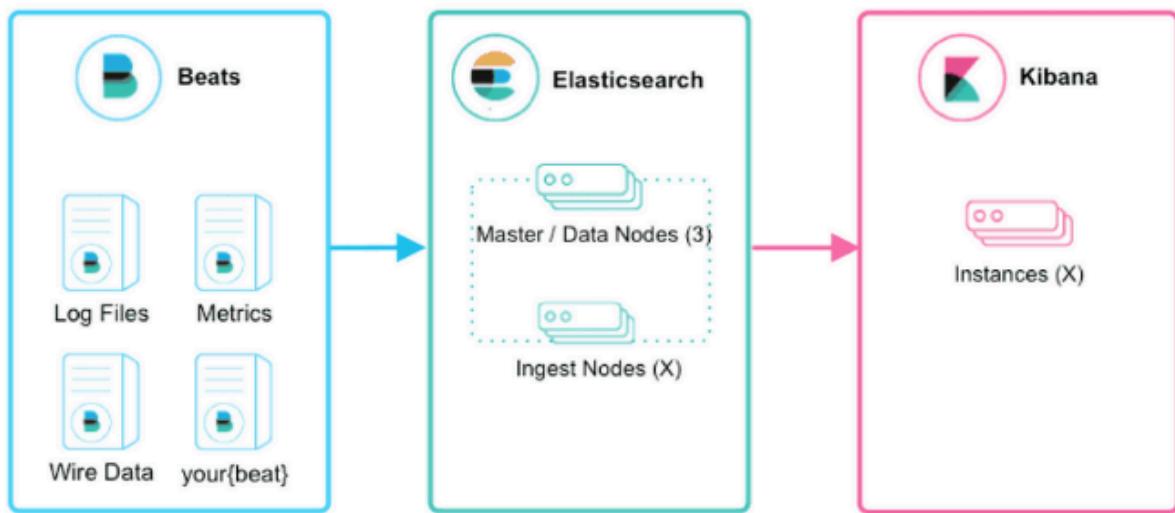


3.2.3 Một số khái niệm cần biết về Elasticsearch

- Document
 - Một trong những khái niệm quan trọng cần biết khi làm việc với Elasticsearch là "Document". Đây là một bản ghi dữ liệu cụ thể, được biểu

diễn dưới dạng cấu trúc JSON. Mỗi document trong Elasticsearch có một unique ID và thuộc về một index cụ thể.

- Bên trong document có thể chứa bất kỳ loại dữ liệu cấu trúc nào, từ văn bản và số liệu đến ngày tháng và đối tượng phức tạp. Documents thường được tổ chức theo kiểu "tài liệu" trong ngữ cảnh của ứng dụng. Ví dụ như document biểu diễn một sản phẩm, một bài blog, một người dùng hoặc một sự kiện. Mỗi tài liệu sẽ được lưu trữ trong một index cụ thể trong Elasticsearch.
- Index
 - Thuật ngữ Index được dùng trong công nghệ Elasticsearch là gì? Đây là một cơ sở dữ liệu tương tự như cơ sở dữ liệu trong hệ thống quan hệ. Tuy nhiên, khác với quan hệ cơ sở dữ liệu, mỗi index trong Elasticsearch thường chứa các loại dữ liệu liên quan đến một chủ đề cụ thể.
 - Mỗi index có thể chứa nhiều "document" (bản ghi dữ liệu) và được sử dụng để tổ chức và lưu trữ dữ liệu một cách hiệu quả để cho việc tìm kiếm và truy xuất. Index được xem như một tập hợp các tài liệu với các thuộc tính và trường dữ liệu tương tự.
- Shard
 - Shard là một khái niệm quan trọng mà bạn cần biết khi tìm hiểu Elasticsearch là gì? Trong quá trình lưu trữ lượng lớn dữ liệu, Elasticsearch chia index thành các phần nhỏ gọi là "shards" để tận dụng khả năng mở rộng ngang và tăng hiệu suất truy xuất dữ liệu.
 - Mỗi shard là một phần của index và chứa một phần dữ liệu. Elasticsearch sử dụng shard để phân phối dữ liệu trên nhiều node, tận dụng khả năng xử lý song song và tăng khả năng mở rộng của hệ thống.
- Node
 - Trong Elasticsearch, "node" là một thành phần quan trọng trong cơ sở hạ tầng phân tán. Node đại diện cho một thành phần riêng lẻ của một cụm Elasticsearch. Chúng có thể làm nhiều công việc như lưu trữ dữ liệu, tham gia vào việc xử lý tìm kiếm và phân tích dữ liệu hoặc đóng vai trò như một điểm giao tiếp cho ứng dụng.
 - Mỗi node trong một cụm Elasticsearch có thể có một vai trò khác nhau. Ví dụ như một node có thể là "data node" - nơi lưu trữ dữ liệu, "master node" - điều phối hoạt động của cụm hoặc "ingest node" - xử lý và tiền xử lý dữ liệu trước khi lưu trữ. Các loại node khác nhau có vai trò quan trọng trong quá trình hoạt động và hiệu suất của cụm Elasticsearch.



- Cluster
 - Cluster trong Elasticsearch tương ứng với một nhóm các nodes hoạt động cùng nhau để lưu trữ dữ liệu và cung cấp các dịch vụ tìm kiếm và phân tích. Cụm này có thể bao gồm một hoặc nhiều nodes Elasticsearch và tạo ra một môi trường phân tán cho việc lưu trữ và xử lý dữ liệu.
 - Phân tán dữ liệu trong Cluster Elasticsearch là cách cung cấp khả năng mở rộng và độ tin cậy cho hệ thống. Quản lý và cấu hình Cluster chính là những yếu tố quan trọng trong việc triển khai, duy trì một hệ thống Elasticsearch.

3.2.4 Yêu cầu bộ nhớ Elasticsearch

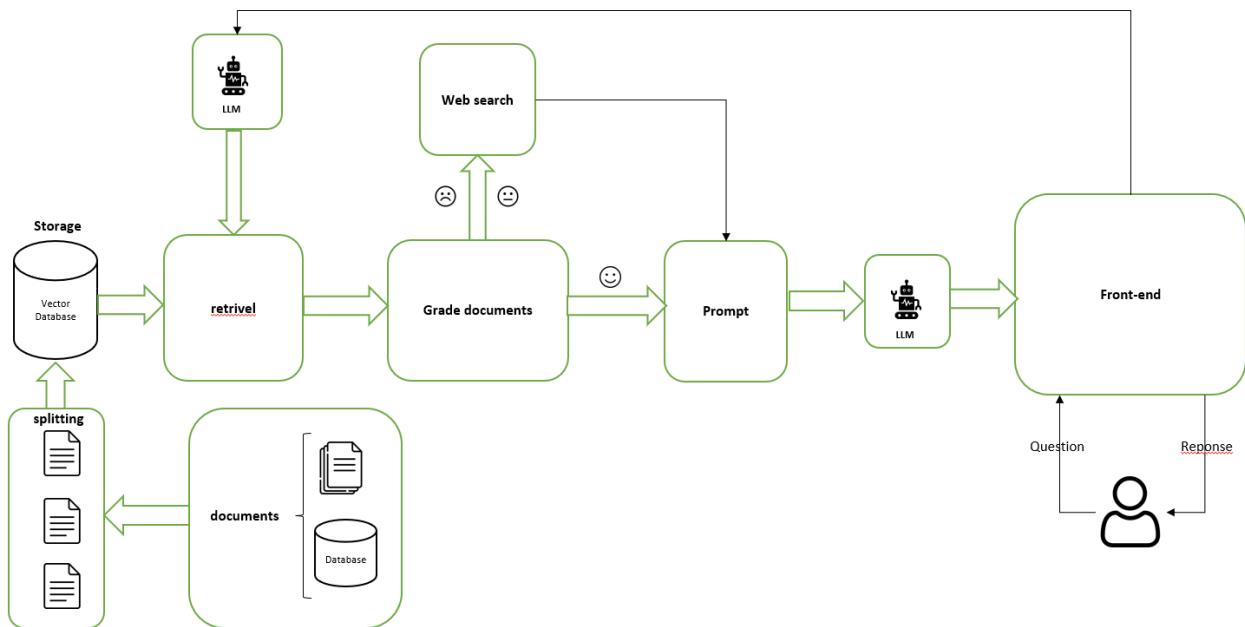
Elasticsearch được chạy trên Java. Do đó, nó yêu cầu công cụ thời gian chạy JVM. Mỗi quá trình Java yêu cầu một lượng bộ nhớ heap cụ thể. Tương tự, Elasticsearch cũng yêu cầu bộ nhớ heap để chạy tiến trình. Phân bổ kích thước heap chính xác cho quá trình elasticsearch đóng một vai trò quan trọng.

3.2.4 Tổng kết

Tóm lại, ElasticSearch chứng tỏ là một công cụ mạnh mẽ và linh hoạt cho các ứng dụng tìm kiếm và quản lý dữ liệu hiện đại. Với khả năng lập chỉ mục hiệu quả, khả năng tìm kiếm cực nhanh và khả năng mở rộng mạnh mẽ, nó trao quyền cho các nhà phát triển và tổ chức cung cấp trải nghiệm người dùng đặc biệt và sâu hơn về dữ liệu của họ.

CHƯƠNG 4: ỨNG DỤNG VÀ THIẾT KẾ CHATBOT

4.1 Luồng hoạt động ứng dụng



1. Người dùng gửi câu hỏi.
 - Người dùng nhập câu hỏi vào khung hỏi đáp trên web.
 - Giao diện gửi yêu cầu chứa câu hỏi đến API Gateway qua HTTP request.
2. API Gateway Chuyển Đến Chatbot Engine:
 - API Gateway xử lý và chuyển yêu cầu đến Chatbot Engine.
 - Chatbot Engine nhận yêu cầu và sử dụng LangChain Integration để xử lý.
3. Tạo câu hỏi cô đọng để truy vấn Elasticsearch
 - Nếu trong cửa sổ đã có những đoạn hội thoại trước đó, câu hỏi của người dùng và lịch sử chat sẽ được gửi cho LLM để tạo câu hỏi cô đọng.
 - Nếu không, giữ nguyên câu hỏi của người dùng.
4. LangChain truy vấn Elasticsearch:
 - LangChain sử dụng câu hỏi cô đọng truy vấn Elasticsearch để tìm các phần tài liệu liên quan.
5. Elasticsearch Trả Kết Quả:
 - Elasticsearch trả về những tài liệu có liên quan đến truy vấn được gửi.
6. Đánh giá độ liên quan của tài liệu.

- Các tài liệu trả về được gộp lại, câu hỏi của người dùng và cả lịch sử chat được gửi cho LLM đánh giá.
- Có 3 loại nhãn được sử dụng: “liên quan”, “không liên quan”, “mơ hồ”
 - “liên quan”: Tài liệu có thể sử dụng để tạo câu trả lời. Không cần tra cứu tài liệu trên web.
 - “không liên quan”: Tài liệu không thể sử dụng để tạo câu trả lời. Tài liệu sẽ bị thay thế bởi kết quả tìm kiếm trên web.
 - “mơ hồ”: Tài liệu có thể gần, nhưng ko đủ để cung cấp câu trả lời. Kết hợp tài liệu hiện tại và kết quả tìm kiếm trên web.

6.1. Tra cứu dữ liệu trên web:

- Sử dụng câu hỏi cố định, tra cứu bằng Tavily Search.
- Kết quả trả về được sử dụng tùy theo nhãn phía trên.

7. Tạo câu trả lời:

- Kết hợp câu hỏi người dùng, lịch sử chat, dữ liệu để tạo prompt đầu vào.
- Sử dụng prompt trên, nhận câu trả lời của LLM

8. API Gateway trả về giao diện người dùng:

- API Gateway trả lại kết quả cho giao diện người dùng.

9. Người dùng nhận phản hồi:

- Người dùng nhận phản hồi và hiển thị trên giao diện.

4.2 Xây dựng ứng dụng

Bước 1 : Cài đặt môi trường và thư viện

Cài đặt Python và các thư viện cần thiết. Tạo một môi trường ảo (virtual environment) và cài đặt các thư viện trong requirements.txt

Bước 2: Thiết lập Elasticsearch

Cài đặt và cấu hình Elasticsearch. Có thể sử dụng Elasticsearch trên cloud hoặc cài đặt local

Tạo một tài khoản Elasticsearch cloud, sau đó tạo một cluster và lưu lại các thông tin kết nối cần thiết như “ELASTIC_CLOUD_ID”, “ELASTIC_API_KEY”, “ELASTICSEARCH_URL”, “ELASTICSEARCH_USERNAME”, “ELASTICSEARCH_PASSWORD”.

Bước 3: Thêm api key của các phần khác trong ứng dụng.

- TAVILY_API_KEY: api key của Tavily để tìm kiếm dữ liệu trên web
- OPENAI_API_KEY: api key của Openai (tùy vào mô hình LLM được sử dụng)

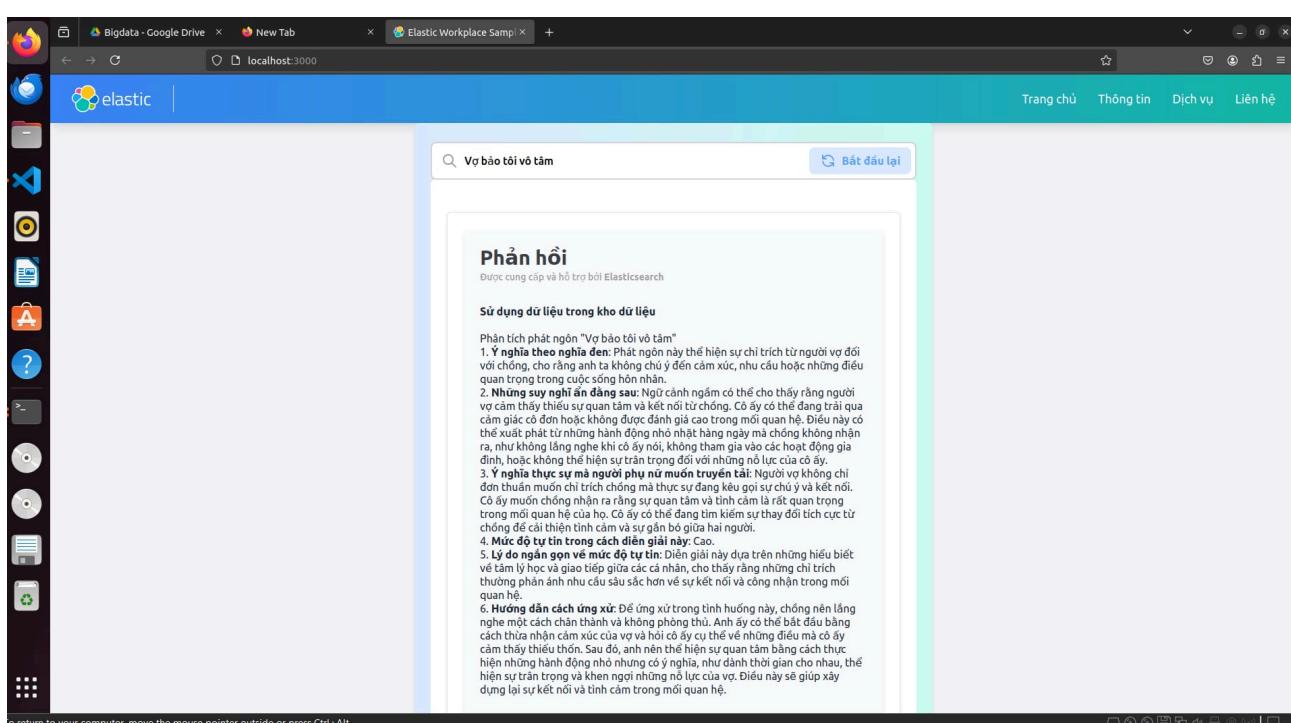
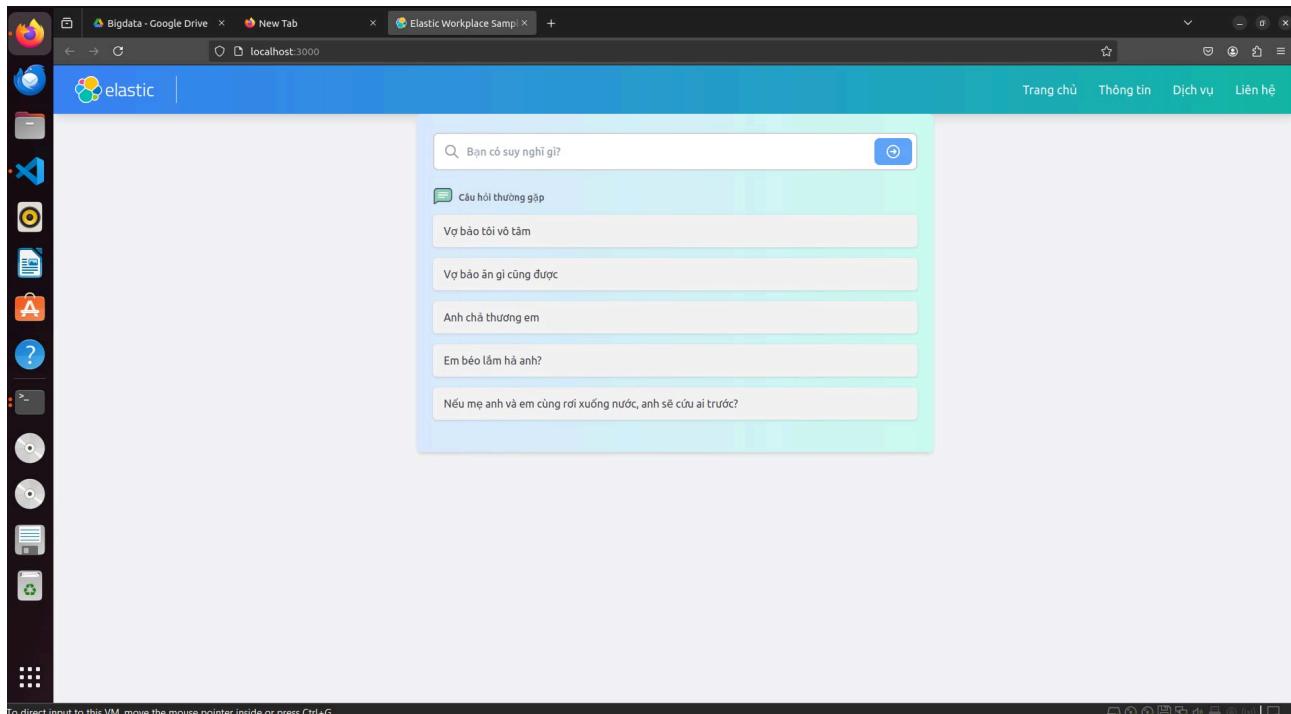
Bước 4: Xử lý và tải dữ liệu vào Elasticsearch

Sử dụng langchain để xử lý dữ liệu từ các file docs, pdf, csv ... Chuyển hóa dữ liệu thành dạng embedding

Tải dữ liệu lên cloud của Elasticsearch

Bước 5: Triển khai chatbot và kết nối với giao diện người dùng

- API Gateway: Triển khai API để nhận yêu cầu từ giao diện người dùng và chuyển đến chatbot. Bạn có thể sử dụng Flask hoặc FastAPI.
- Giao diện người dùng (UI): Tạo giao diện người dùng để người dùng có thể nhập câu hỏi và nhận phản hồi từ chatbot



4.3 Đánh giá

Mô hình LLM mà chatbot đang sử dụng là GPT-4o-mini. Chúng tôi sẽ so sánh kết quả so với phiên bản GPT-4o-mini.

Để công bằng, prompt gốc của chatbot cũng được sử dụng cho phiên bản GPT-4o-mini.

Dữ liệu:

Dữ liệu được sử dụng:

- Sách Đàn ông đến từ sao Hỏa, đàn bà đến từ sao Kim. Tác giả: Tiến sĩ John Gray
Dịch giả: Phạm Thị Thanh Huyền, Ngô Thị Chuyên.
- Tài liệu: Phụ nữ muốn gì ở đàn ông? (Nguồn Scribd)
- Dữ liệu công khai trên internet.

So sánh câu trả lời :

Câu hỏi	ChatBot	GPT-4o-mini
Vợ bảo tôi vô tâm	<p>Sử dụng dữ liệu trong kho dữ liệu Phân tích phát ngôn của người phụ nữ</p> <p>1. Ý nghĩa theo nghĩa đen: Người phụ nữ cảm thấy chồng mình không quan tâm đến cô, đặc biệt là khi cô muốn trò chuyện. Cô cảm thấy bị bỏ rơi và không được lắng nghe.</p> <p>2. Những suy nghĩ ẩn đằng sau: Phát ngôn này không chỉ đơn thuần là một lời phàn nàn về việc chồng không nói chuyện. Nó còn phản ánh sự thiếu thốn trong mối quan hệ, cảm giác cô đơn và sự cần thiết được thấu hiểu. Cô có thể đang trải qua những cảm xúc phức tạp, như sự thất vọng, buồn bã và cảm giác không được yêu thương.</p>	<p>1. Ý nghĩa theo nghĩa đen:</p> <ul style="list-style-type: none">Câu nói "vô tâm" trong ngữ cảnh này có thể hiểu theo nghĩa đen là "không quan tâm" hoặc "thiếu chú ý đến cảm xúc của người khác". Vợ bạn có thể đang nói rằng bạn không chú ý đến những nhu cầu cảm xúc hoặc hành động của cô ấy, dẫn đến cảm giác bị bỏ rơi hoặc không được quan tâm. <p>2. Những suy nghĩ ẩn đằng sau:</p> <ul style="list-style-type: none">Ngữ cảnh ngầm: Khi người phụ nữ nói bạn vô tâm, có thể cô ấy không chỉ đang phê

3. Ý nghĩa thực sự mà người phụ nữ muốn truyền tải: Cô ấy không chỉ muốn chồng lắng nghe mà còn muốn cảm thấy rằng mình quan trọng và được yêu thương. Cô ấy cần sự kết nối cảm xúc và sự hỗ trợ từ chồng, đặc biệt trong những lúc khó khăn. Cô ấy có thể đang tìm kiếm sự khẳng định rằng mối quan hệ của họ vẫn còn giá trị và đáng được chăm sóc.

4. Mức độ tự tin trong cách diễn giải: Cao.

5. Lý do về mức độ tự tin: Diễn giải này dựa trên những yếu tố cảm xúc và tâm lý phổ biến trong các mối quan hệ, cũng như bối cảnh mà người phụ nữ đang trải qua. Những cảm xúc như cô đơn và cần được lắng nghe là rất phổ biến và dễ hiểu trong tình huống này.

6. Hướng dẫn cách ứng xử: Để cải thiện tình hình, người chồng nên chủ động lắng nghe và tạo không gian cho vợ chia sẻ cảm xúc của mình. Anh ấy có thể bắt đầu bằng cách hỏi cô ấy về những điều đang làm cô ấy cảm thấy không thoải mái và thể hiện sự quan tâm chân thành. Việc này không chỉ giúp vợ cảm

phản hành động cụ thể nào đó, mà đang bày tỏ sự thất vọng về việc bạn không thể hiện sự quan tâm đủ mạnh mẽ đối với mối quan hệ hay những cảm xúc của cô ấy. Cô ấy có thể cảm thấy cô đơn, bị lãng quên hoặc không được lắng nghe.

● **Ý nghĩa ẩn giấu:** Việc cô ấy sử dụng từ "vô tâm" có thể là cách gián tiếp yêu cầu bạn chú ý đến cô ấy hơn, quan tâm nhiều hơn đến những chi tiết nhỏ trong cuộc sống hàng ngày, cũng như thể hiện sự thấu hiểu về cảm xúc của cô ấy.

3. Ý nghĩa thực sự mà bạn có thể muốn truyền tải:

● Tôi nghĩ rằng vợ bạn không chỉ đang nói về hành vi thiếu quan tâm cụ thể nào đó mà có thể đang muốn bạn thể hiện nhiều hơn sự nhạy cảm và sự quan tâm đối với cảm xúc của cô ấy. Cô ấy có thể đang cảm thấy thiếu sự kết nối hoặc không nhận được đủ

	<p>thấy được lắng nghe mà còn củng cố mối quan hệ của họ.</p>	<p>sự hỗ trợ về mặt tinh thần từ bạn.</p>
	<p>4. Mức độ tự tin trong cách diễn giải này:</p> <ul style="list-style-type: none"> • Tự tin trung bình. Tôi tự tin rằng đây là một phản ánh khá đúng về những gì có thể đang diễn ra, nhưng như mọi tình huống giao tiếp, ngữ cảnh cụ thể có thể thay đổi ý nghĩa, vì vậy cần phải thận trọng khi áp dụng. 	<p>5. Lý do ngắn gọn về mức độ tự tin:</p> <ul style="list-style-type: none"> • Phản ứng của vợ bạn là một tín hiệu khá rõ ràng về cảm giác thiếu sự quan tâm. Tuy nhiên, cũng có thể có yếu tố cảm xúc khác đằng sau câu nói này mà bạn cần phải hiểu thêm qua các hành động và ngữ cảnh cụ thể.
	<p>6. Hướng dẫn cách ứng xử:</p> <ul style="list-style-type: none"> • Lắng nghe và thể hiện sự thấu hiểu là bước quan trọng. Hãy ngồi lại và hỏi cô ấy cảm thấy như thế nào, những điều cô ấy cần hoặc mong muốn. Đừng chỉ biện minh, mà hãy thể 	

		hiện sự quan tâm và sẵn sàng cải thiện.
Vợ bảo ăn gì cũng được (tiếp tục hỏi từ câu trên)	<p>Tra cứu dữ liệu trên internet</p> <p>Phân tích phát ngôn của người phụ nữ</p> <p>1. Ý nghĩa theo nghĩa đen: Khi người phụ nữ nói "Ăn gì cũng được", cô ấy đang thể hiện sự linh hoạt trong việc lựa chọn món ăn, có vẻ như không có yêu cầu cụ thể nào về thực phẩm.</p> <p>2. Những suy nghĩ ẩn đằng sau: Tuy nhiên, câu nói này thường mang một ý nghĩa sâu xa hơn. Nó có thể phản ánh sự mệt mỏi hoặc chán nản khi phải đưa ra quyết định, hoặc có thể là một cách để tránh xung đột. Cô ấy có thể cảm thấy áp lực khi phải chọn món ăn, hoặc có thể đang tìm kiếm sự quan tâm từ người yêu, mong muốn anh ấy chủ động hơn trong việc lựa chọn.</p> <p>3. Ý nghĩa thực sự mà người phụ nữ muốn truyền tải: Cô ấy không chỉ muốn đơn giản là ăn gì đó, mà thực sự mong muốn người yêu thể hiện sự</p>	<p>1. Ý nghĩa theo nghĩa đen:</p> <ul style="list-style-type: none"> Câu nói "ăn gì cũng được" theo nghĩa đen có thể hiểu là vợ bạn không có sự yêu cầu cụ thể về món ăn, cô ấy sẵn sàng ăn bất cứ món gì bạn chọn. <p>2. Những suy nghĩ ẩn đằng sau:</p> <ul style="list-style-type: none"> Ngữ cảnh ngầm: Mặc dù câu nói này có vẻ như là một lời đồng ý, nhưng thường khi phụ nữ nói "ăn gì cũng được," họ có thể đang biểu lộ sự mệt mỏi hoặc không còn hứng thú trong việc phải quyết định. Điều này có thể là dấu hiệu của việc cô ấy không muốn phải đưa ra lựa chọn hoặc đã cảm thấy không được quan tâm trong việc đưa ra quyết định. Ý nghĩa ẩn giấu: Câu nói có thể không đơn giản chỉ là "tùy ý bạn," mà có thể thể hiện cảm giác thất

	<p>quan tâm và chủ động trong mối quan hệ. Câu nói này có thể là một cách để cô ấy thử thách người yêu, xem liệu anh có nhận ra rằng cô đang cần sự chú ý và sự quan tâm từ anh hay không.</p> <p>4. Mức độ tự tin trong cách diễn giải: Trung bình.</p> <p>5. Lý do về mức độ tự tin: Mặc dù có nhiều yếu tố tâm lý và cảm xúc liên quan đến câu nói này, nhưng mỗi cá nhân có thể có những lý do khác nhau cho việc sử dụng câu nói này. Do đó, sự diễn giải có thể không hoàn toàn chính xác cho mọi trường hợp.</p> <p>6. Hướng dẫn cách ứng xử: Để cải thiện tình huống, người yêu nên chủ động đưa ra một vài lựa chọn cụ thể cho bữa ăn, hoặc hỏi cô ấy về những món ăn mà cô thích. Điều này không chỉ giúp giảm bớt áp lực cho cô mà còn thể hiện sự quan tâm và chăm sóc từ phía anh. Hơn nữa, việc này có thể tạo ra những khoảnh khắc thú vị và gắn kết hơn trong mối quan hệ của họ.</p>	<p>vọng hoặc muốn bạn thể hiện sự quan tâm và quyết đoán hơn trong việc chọn lựa món ăn. Đây cũng có thể là cách để cô ấy kiểm tra xem bạn có quan tâm đến sở thích của cô ấy hay không, hay chỉ chọn món ăn mà bạn muốn.</p> <p>3. Ý nghĩa thực sự mà bạn có thể muốn truyền tải:</p> <ul style="list-style-type: none"> Tôi nghĩ rằng vợ bạn không thực sự chỉ muốn bạn chọn món ăn một cách đơn giản. Cô ấy có thể đang thể hiện một cảm giác mệt mỏi, thiếu sự kết nối, hoặc muốn bạn chủ động và chăm sóc hơn trong việc quyết định, thay vì để cô ấy luôn phải làm điều đó. Câu nói này có thể là một dấu hiệu ngầm về việc cô ấy muốn bạn quan tâm đến cảm xúc và sự mong đợi của cô ấy hơn trong các tình huống tương tự. <p>4. Mức độ tự tin trong cách diễn giải này:</p> <ul style="list-style-type: none"> Tự tin trung bình. Câu nói này có thể được hiểu theo
--	--	--

nhiều cách tùy vào ngữ cảnh cụ thể và các yếu tố cảm xúc khác, nhưng tôi khá tự tin rằng đây là một tín hiệu gián tiếp của việc vợ bạn muốn bạn thể hiện sự quan tâm hơn nữa.

5. Lý do ngắn gọn về mức độ tự tin:

- Câu nói "ăn gì cũng được" không phải lúc nào cũng có nghĩa là thực sự không quan tâm, mà thường là dấu hiệu của sự thụ động hoặc mong muốn người khác quyết định. Tuy nhiên, cũng cần thận trọng vì có thể cô ấy chỉ đơn giản là không có sự lựa chọn đặc biệt vào lúc đó.

6. Hướng dẫn cách ứng xử:

- Bạn có thể thử chủ động đưa ra vài lựa chọn và hỏi ý kiến vợ về việc cô ấy thích món nào trong những lựa chọn đó. Điều này sẽ thể hiện sự quan tâm và giúp bạn tránh cảm giác rằng cô ấy không được lắng nghe. Nếu có thể, bạn cũng có

		<p>thể chú ý tới các sở thích ăn uống của cô ấy từ những lần trước để làm cho cô ấy cảm thấy đặc biệt.</p>
Anh chả thương em	<p>Sử dụng dữ liệu trong kho dữ liệu</p> <p>Phân tích phát ngôn</p> <p>1. Ý nghĩa theo nghĩa đen:</p> <p>Câu nói "Anh chả thương em" có thể được hiểu đơn giản là người phụ nữ đang bày tỏ cảm giác thất vọng hoặc tổn thương vì cảm nhận rằng người đàn ông không còn tình cảm với cô.</p> <p>2. Những suy nghĩ ẩn đằng sau:</p> <p>Ngữ cảnh ngầm của câu nói này có thể phản ánh sự thiếu thốn tình cảm, sự không chắc chắn về mối quan hệ, hoặc cảm giác bị bỏ rơi. Cô ấy có thể đang cảm thấy cô đơn và cần sự khẳng định từ người đàn ông về tình cảm của anh ta dành cho cô. Câu nói này cũng có thể là một cách để cô ấy kêu gọi sự chú ý và quan tâm từ đối phương.</p> <p>3. Ý nghĩa thực sự mà người phụ nữ muốn truyền tải:</p> <p>Người phụ nữ có thể không chỉ đơn thuần muốn nói rằng anh không thương cô, mà</p>	<p>Ý nghĩa theo nghĩa đen:</p> <p>Câu nói "Anh chả thương em" theo nghĩa đen là một lời tố cáo hoặc than vãn của người phụ nữ rằng người đàn ông không có tình cảm, không yêu thương hoặc không quan tâm đến cô ấy.</p> <p>Những suy nghĩ ẩn đằng sau:</p> <p>Cảm giác tổn thương: Câu nói này có thể xuất phát từ cảm giác bị bỏ rơi hoặc thiếu sự quan tâm từ phía người đàn ông. Người phụ nữ có thể cảm thấy mối quan hệ không được chăm sóc đúng mức.</p> <p>Mong muốn được chú ý: Có thể cô ấy đang tìm kiếm sự xác nhận hoặc sự chú ý từ người đàn ông, hy vọng anh ấy sẽ nhận ra sự thiếu sót trong hành động hoặc thái độ của mình.</p> <p>Nỗi buồn và thất vọng: Đây có thể là một cách để thể hiện sự thất vọng trong mối quan hệ, khi cô ấy cảm thấy không được yêu thương hoặc đánh giá đúng mức.</p> <p>Ý nghĩa thực sự mà người</p>

	<p>thực sự cô đang tìm kiếm sự xác nhận về tình cảm của anh. Cô ấy có thể cảm thấy không được yêu thương hoặc không được đánh giá cao trong mối quan hệ này, và câu nói này là một cách để thể hiện nỗi đau và sự cần thiết của sự quan tâm từ phía anh.</p> <p>4. Mức độ tự tin trong cách diễn giải:</p> <p>Tôi tự tin ở mức độ cao trong cách diễn giải này.</p> <p>5. Lý do về mức độ tự tin:</p> <p>Sự tự tin này đến từ việc hiểu rõ các sắc thái trong giao tiếp giữa các cá nhân, đặc biệt là trong mối quan hệ tình cảm. Câu nói này thường xuất hiện trong bối cảnh khi một bên cảm thấy thiếu thốn tình cảm và cần sự khẳng định từ bên kia.</p> <p>6. Hướng dẫn cách ứng xử:</p> <p>Trong tình huống này, người đàn ông nên lắng nghe và thể hiện sự quan tâm đến cảm xúc của người phụ nữ. Anh có thể hỏi thêm về cảm giác của cô, khẳng định tình cảm của mình và thể hiện sự sẵn sàng để cải thiện mối quan hệ. Một câu nói như "Anh yêu em và anh muốn hiểu rõ hơn về những gì em đang cảm thấy" có thể giúp làm dịu tình hình và tạo ra một không gian an toàn để cô ấy chia sẻ cảm xúc của mình.</p>	<p>phụ nữ muốn truyền tải: Người phụ nữ có thể đang thể hiện sự đau khổ vì cảm thấy thiếu sự quan tâm hoặc tình cảm từ người đàn ông. Đây không phải là một câu nói đơn giản mà là một cách để tìm kiếm sự chú ý hoặc sự khẳng định tình cảm từ đối phương. Cô ấy có thể không chỉ muốn nghe một lời xin lỗi, mà còn mong muốn thấy sự thay đổi trong hành động hoặc thái độ của người đàn ông.</p> <p>Mức độ tự tin trong diễn giải:</p> <p>Trung bình. Dù câu nói này có thể được hiểu theo nhiều cách, những yếu tố cảm xúc mạnh mẽ như sự tổn thương và sự tìm kiếm sự chú ý là những yếu tố phổ biến trong trường hợp này.</p> <p>Lý do:</p> <p>Câu nói này rất phổ biến trong những mối quan hệ gần gũi, đặc biệt khi một người cảm thấy không được yêu thương hoặc quan tâm. Cách diễn giải này dựa trên ngữ cảnh thường gặp trong giao tiếp tình cảm, nhưng không thể khẳng định hoàn toàn chính xác mà không có thông tin cụ thể hơn về mối quan hệ.</p> <p>Hướng dẫn cách ứng xử:</p>
--	---	--

		<p>Lắng nghe và thấu hiểu: Đầu tiên, hãy lắng nghe những cảm xúc và suy nghĩ của cô ấy, thể hiện sự quan tâm và sẵn sàng giải quyết vấn đề.</p> <p>Tránh phủ nhận cảm xúc: Thay vì ngay lập tức phản bác hoặc biện minh, hãy tỏ ra thông cảm với cảm giác tổn thương của cô ấy và tim cách nói chuyện mở về những gì đang làm tổn thương cô ấy.</p> <p>Xây dựng lại sự kết nối: Nếu có thể, hãy thảo luận về những cách thức mà cả hai có thể cải thiện mối quan hệ, đảm bảo rằng cô ấy cảm thấy được yêu thương và quan tâm.</p>
Bạn gái hỏi tôi một câu hỏi rất khó chịu: "nếu mẹ anh và em cùng rơi xuống nước, anh sẽ cứu ai trước"	<p>Tra cứu dữ liệu trên internet Phân tích phát ngôn</p> <p>Ý nghĩa theo nghĩa đen: Câu hỏi "Nếu mẹ anh và em cùng rơi xuống nước, anh sẽ cứu ai trước?" là một tình huống giả định mà người phụ nữ đặt ra để kiểm tra sự ưu tiên và tình cảm của bạn đối với cô ấy so với mẹ của bạn.</p> <p>Những suy nghĩ ẩn đằng sau: Câu hỏi này không chỉ đơn thần là một bài kiểm tra</p>	<p>Ý nghĩa theo nghĩa đen: Phát ngôn này là một câu hỏi giả định, yêu cầu bạn lựa chọn giữa việc cứu mẹ bạn hoặc bạn gái của mình trong tình huống nguy hiểm.</p> <p>Những suy nghĩ ẩn đằng sau (Ngữ cảnh ngầm hoặc các ý nghĩa ẩn giấu): Câu hỏi này có thể không chỉ đơn giản là một thử thách về việc cứu ai trước trong một tình huống cực đoan. Nó có thể là một cách để bạn gái thể hiện sự lo lắng về vị trí của cô ấy</p>

	<p>về khả năng ra quyết định trong tình huống khẩn cấp, mà còn phản ánh những lo lắng và sự không chắc chắn trong mối quan hệ. Người phụ nữ có thể đang tìm kiếm sự khẳng định về vị trí của mình trong trái tim bạn, cũng như mong muốn biết rằng bạn sẽ đặt cô ấy lên hàng đầu trong những tình huống khó khăn. Nó cũng có thể là một cách để cô ấy thể hiện sự ghen tuông hoặc cảm giác không an toàn trong mối quan hệ.</p> <p>Ý nghĩa thực sự mà người phụ nữ muốn truyền tải:</p> <p>Người phụ nữ có thể muốn biết rằng bạn coi trọng mối quan hệ của hai người và rằng cô ấy có một vị trí quan trọng trong cuộc sống của bạn. Câu hỏi này có thể là một cách để cô ấy kiểm tra sự cam kết của bạn và cảm giác an toàn trong tình yêu.</p> <p>Mức độ tự tin trong cách diễn giải:</p> <p>Tôi tự tin ở mức trung bình trong cách diễn giải này.</p> <p>Lý do về mức độ tự tin:</p> <p>Mặc dù có thể có nhiều lý do khác nhau cho câu hỏi này, nhưng những lo lắng</p>	<p>trong mối quan hệ của bạn, đặc biệt là khi đối diện với các mối quan hệ thân thiết khác (như gia đình). Cô ấy có thể cảm thấy không chắc chắn về sự ưu tiên của bạn hoặc lo sợ rằng cô ấy không phải là người quan trọng nhất trong cuộc sống của bạn.</p> <p>Ý nghĩa thực sự mà bạn gái muốn truyền tải:</p> <p>Phát ngôn này có thể là một cách để bạn gái tìm kiếm sự đảm bảo rằng cô ấy sẽ được ưu tiên trong cuộc sống của bạn, đặc biệt là trong những tình huống khó khăn. Đây là một câu hỏi không phải để nhận câu trả lời "đúng" hay "sai", mà là để thử thách mức độ cam kết và sự quan tâm của bạn đối với mối quan hệ. Cô ấy có thể muốn thấy rằng bạn sẽ bảo vệ và chọn cô ấy nếu phải đưa ra quyết định khó khăn.</p> <p>Mức độ tự tin trong diễn giải này:</p> <p>Tôi tự tin ở mức trung bình.</p> <p>Lý do về mức độ tự tin:</p> <p>Câu hỏi này có thể có nhiều lớp ý nghĩa tùy thuộc vào mối quan hệ và cảm xúc của người hỏi, và không có một cách giải</p>
--	--	--

	<p>về sự ưu tiên và cảm giác an toàn trong mối quan hệ là những yếu tố phổ biến trong tâm lý của nhiều người phụ nữ. Tuy nhiên, không thể chắc chắn hoàn toàn về động cơ cụ thể của từng cá nhân.</p> <p>Hướng dẫn cách ứng xử:</p> <p>Khi đối mặt với câu hỏi này, bạn nên trả lời một cách chân thành và nhấn mạnh rằng bạn yêu thương cả mẹ và cô ấy. Bạn có thể nói rằng trong tình huống khẩn cấp, bạn sẽ cố gắng cứu cả hai người nếu có thể, và điều quan trọng nhất là bạn muốn cả hai đều an toàn. Điều này sẽ giúp cô ấy cảm thấy được tôn trọng và an toàn trong mối quan hệ.</p>	<p>thích duy nhất. Tuy nhiên, dựa trên sự thường gặp trong các tình huống tương tự, việc cô ấy muốn tìm kiếm sự đảm bảo về sự ưu tiên của mình trong mối quan hệ là một khả năng khá cao.</p> <p>Hướng dẫn cách ứng xử:</p> <p>Khi đối diện với câu hỏi như thế này, bạn nên trả lời một cách tinh tế và chân thành. Thay vì chỉ đơn giản lựa chọn, bạn có thể nói rằng bạn yêu thương cả mẹ và bạn gái của mình, và trong một tình huống thực tế, bạn sẽ làm hết sức mình để cứu cả hai, hoặc là nhấn mạnh rằng bạn luôn muốn duy trì sự cân bằng trong các mối quan hệ quan trọng. Câu trả lời của bạn nên thể hiện sự quan tâm và sự hiểu biết về cảm xúc của cô ấy, thay vì chỉ tập trung vào việc chọn lựa.</p>
Cô ấy hay hỏi tôi: "Em béo lắm hả anh?" Nếu tôi gật đầu đúng theo sự thật thì sẽ bị giận nguyên ngày. Còn nếu nói không thì cô ấy bảo tôi nói dối, không trả lời thành thật và cũng bị giận nguyên ngày hôm đó.	<p>Sử dụng dữ liệu trong kho dữ liệu kết hợp tra cứu dữ liệu trên internet</p> <p>Phân tích phát ngôn</p> <p>Mô tả ngắn gọn ý nghĩa theo nghĩa đen:</p> <p>Câu hỏi "Em béo lắm hả anh?" của cô ấy có thể được hiểu là một cách để</p>	<p>Ý nghĩa theo nghĩa đen:</p> <p>Phát ngôn "Em béo lắm hả anh?" có vẻ là một câu hỏi đơn giản, yêu cầu xác nhận về ngoại hình của người phụ nữ. Cô ấy đang tự hỏi về cân nặng hoặc vóc dáng của mình, và mong muốn có một câu trả lời từ bạn.</p>

	<p>cô ấy tìm kiếm sự xác nhận về hình ảnh bản thân. Cô ấy đang hỏi về cảm nhận của bạn về cân nặng của mình.</p> <p>Những suy nghĩ ẩn đằng sau:</p> <p>Câu hỏi này không chỉ đơn thuần là về cân nặng. Nó có thể phản ánh sự không tự tin của cô ấy về ngoại hình và mong muốn nhận được sự khẳng định từ bạn. Cô ấy có thể đang tìm kiếm sự an ủi hoặc khẳng định rằng bạn vẫn yêu thương cô ấy bất kể ngoại hình. Câu hỏi này cũng có thể là một cách để cô ấy kiểm tra phản ứng của bạn, xem bạn có quan tâm đến cảm xúc của cô ấy hay không.</p> <p>Ý nghĩa thực sự mà người phụ nữ muốn truyền tải:</p> <p>Cô ấy có thể đang cảm thấy không hài lòng với bản thân và muốn bạn giúp cô ấy cảm thấy tốt hơn về chính mình. Câu hỏi này có thể là một cách để cô ấy thể hiện sự nhạy cảm và mong muốn được lắng nghe, chứ không chỉ đơn thuần là một câu hỏi về cân nặng.</p> <p>Mức độ tự tin trong cách diễn giải này:</p> <p>Tôi đánh giá mức độ tự tin</p>	<p>Những suy nghĩ ẩn đằng sau:</p> <p>Dưới vẻ bế ngoài của câu hỏi, có thể có sự tự ti hoặc lo lắng về ngoại hình của cô ấy. Câu hỏi này có thể là một cách để cô ấy tìm kiếm sự khẳng định hoặc cảm giác an toàn từ bạn. Tuy nhiên, câu hỏi cũng có thể mang tính thử thách, với mục đích đánh giá sự nhạy bén và sự tinh tế trong cách bạn trả lời.</p> <p>Ý nghĩa thực sự mà người phụ nữ muốn truyền tải:</p> <p>Cô ấy không chỉ đơn giản muốn biết sự thật về ngoại hình của mình. Thực chất, cô ấy có thể đang tìm kiếm sự đồng cảm, sự hiểu biết và hỗ trợ từ bạn về vấn đề tự ti hoặc sự lo lắng về hình thể. Mặc dù câu hỏi có vẻ như một yêu cầu xác nhận về cân nặng, nhưng thực tế cô ấy có thể đang thử đo lường mức độ tinh tế và độ quan tâm của bạn khi trả lời câu hỏi này.</p> <p>Mức độ tự tin trong cách diễn giải này:</p> <p>Tự tin trung bình.</p> <p>Lý do:</p> <p>Mặc dù có thể là một câu hỏi đơn giản về ngoại hình, nhưng trong nhiều trường</p>
--	---	--

	<p>của mình là **cao**.</p> <p>Lý do ngắn gọn về mức độ tự tin:</p> <p>Điễn giải này dựa trên những hiểu biết về tâm lý học và giao tiếp giữa các cá nhân, đặc biệt là trong bối cảnh mà phụ nữ thường có xu hướng tìm kiếm sự xác nhận từ người khác về hình ảnh bản thân.</p> <p>Hướng dẫn cách ứng xử:</p> <p>Khi cô ấy hỏi câu này, bạn nên trả lời một cách nhẹ nhàng và khéo léo. Thay vì chỉ đơn thuần trả lời "Có" hay "Không", bạn có thể nói: "Em luôn xinh đẹp trong mắt anh, và anh yêu em không chỉ vì ngoại hình mà còn vì con người em." Điều này sẽ giúp cô ấy cảm thấy được yêu thương và an ủi, đồng thời tránh được sự giận dỗi không cần thiết. Hãy lắng nghe và khuyến khích cô ấy chia sẻ cảm xúc của mình để tạo ra một không gian giao tiếp tích cực hơn.</p>	<p>hợp, những câu hỏi như vậy thường ẩn chứa một sự mong đợi về sự đồng cảm và nhạy cảm từ người đối diện. Vì vậy, cách bạn xử lý câu hỏi này rất quan trọng, và phản ứng của cô ấy khi nhận được câu trả lời có thể không chỉ phản ánh sự thật mà còn là cảm xúc sâu thẳm bên trong.</p> <p>Hướng dẫn cách ứng xử:</p> <ul style="list-style-type: none"> • Cố gắng tránh trả lời theo cách quá trực tiếp (ví dụ: "Vâng, em béo" hoặc "Không, em không béo"), vì điều này có thể khiến cô ấy cảm thấy tổn thương hoặc nghi ngờ. • Thay vào đó, hãy tập trung vào việc an ủi và hỗ trợ cảm xúc của cô ấy. Ví dụ, bạn có thể trả lời: "Anh nghĩ em luôn xinh đẹp và quan trọng là em cảm thấy tự tin về bản thân mình." • Cố gắng khuyến khích cô ấy chia sẻ cảm xúc và suy nghĩ của mình thay vì chỉ tập trung vào ngoại hình.
--	---	--

Đánh giá: Với việc cung cấp dữ liệu tham khảo từ CRAG, Chatbot trả lời tự tin hơn GPT-4o-mini. Tuy không ước lượng được, nhưng có thể thấy các câu trả lời sinh ra từ Chatbot cô đọng, chính xác tập trung vào vấn đề so với câu trả lời từ GPT-4o-mini. Với một số câu hỏi liên quan đến các sự kiện, thông tin mới gần đây trên mạng xã hội, Chatbot vẫn có thể trả lời nhờ việc tra cứu nội dung trên internet.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Đề tài xây dựng chatbot sử dụng các công nghệ CRAG, Elasticsearch và LangChain đã mang lại nhiều cơ hội học hỏi và thử thách cho nhóm chúng em trong việc nghiên cứu và ứng dụng các mô hình LLM, kỹ thuật CRAG, cũng như công nghệ Elasticsearch và LangChain vào thiết kế và phát triển chatbot. Nhóm chúng em đã đạt được một số kết quả nổi bật, bao gồm:

- Nắm bắt được các kiến thức cơ bản và tổng quan về lịch sử, định nghĩa, và các mô hình LLM, đặc biệt là ứng dụng của LLM trong chatbot.
- Hiểu rõ về hiện tượng ảo giác trong các mô hình LLM và ứng dụng kỹ thuật CRAG để giảm thiểu các vấn đề này.
- Áp dụng thành công LangChain và Elasticsearch để phát triển ứng dụng chatbot có khả năng tìm kiếm và trả lời chính xác dựa trên nguồn dữ liệu lớn.

Mặc dù vậy, dự án vẫn còn một số thách thức cần khắc phục:

- Chương trình mới chỉ hỗ trợ dữ liệu pdf.
- Khả năng mở rộng

5.2 Hướng phát triển

Trong tương lai, nhóm chúng em sẽ tiếp tục cải tiến và mở rộng ứng dụng chatbot bằng các hướng phát triển sau:

- Áp dụng kiến thức về LLM, kỹ thuật CRAG, LangChain và Elasticsearch để cải tiến khả năng phân tích và xử lý dữ liệu, mở rộng hỗ trợ cho nhiều loại dữ liệu khác nhau.
- Phát triển khả năng mở rộng của chatbot để có thể ứng dụng vào nhiều lĩnh vực khác nhau như y tế, giáo dục, hỗ trợ khách hàng, và nhiều ngành nghề khác.
- Tích hợp các tính năng mới như học máy (Machine Learning) để giúp chatbot hiểu và học từ các cuộc trò chuyện, cải thiện độ chính xác và sự tương tác với người dùng.
- Đẩy mạnh việc tối ưu hóa hiệu suất tìm kiếm và phản hồi, đặc biệt khi làm việc với lượng dữ liệu lớn, để đảm bảo chatbot luôn phản hồi nhanh chóng và chính xác.

Trong quá trình hoàn thành đề tài, nhóm chúng em đã cố gắng tìm hiểu và tham khảo các tài liệu liên quan. Tuy nhiên, do thời gian có hạn nên chúng em sẽ không tránh khỏi những thiếu sót. Rất mong nhận được sự đóng góp ý kiến của thầy cô và các bạn để giúp hoàn thiện báo cáo cũng như kỹ năng của mình trong các dự án sau.

TÀI LIỆU THAM KHẢO

- [1] <https://github.com/elastic/elasticsearch-labs/tree/main/example-apps/chatbot-rag-app>
- [2] <https://arxiv.org/abs/2401.15884>
- [3] <https://viblo.asia/p/llm-101-paper-reading-tim-hieu-corrective-retrieval-augmented-generation-crags-Ym40KnoV91>
- [4] <https://docs.tavily.com/docs/python-sdk/tavily-search/getting-started>
- [5] <https://medium.com/gitconnected/langgraph-corrective-rag-local-llm-powerful-rag-chatbot-f82c9fae>

NHIỆM VỤ CỦA CÁC THÀNH VIÊN

Họ và tên	Công việc
Nguyễn Gia Lộc	+ Tìm hiểu về LLMs + Tìm hiểu về ElasticSearch, LangChain + Viết báo cáo + Làm slide thuyết trình
Trần Đức Đăng Khôi	+ Thiết kế giao diện chatbot + Tìm hiểu về ChatGPT + Thu thập dữ liệu + Viết báo cáo
Vũ Minh Khải	+ Cài đặt chatbot + Cài đặt CRAG + Đưa ra kết luận và hướng phát triển + Viết báo cáo
Đỗ Tiến Dũng	+ Tìm hiểu kỹ thuật CRAG + Cài đặt và đánh giá CRAG + Viết báo cáo + Làm slide thuyết trình