

# ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH

**TRƯỜNG** **ĐẠI** **HỌC** **CÔNG** **NGHỆ** **THÔNG** **TIN** **KHOA** **HỆ** **THỐNG** **THÔNG** **TIN**

**BÁO** **CÁO** **ĐỒ** **ÁN**

**TRUY VẤN THÔNG TIN ĐA PHƯƠNG TIỆN**

**ĐỀ** **TÀI**

**TRIỂN KHAI HỆ THỐNG CHATBOT BẰNG RAG VÀ LANGCHAIN**

Giảng viên hướng dẫn: Ths.Đỗ Văn Tiến

Sinh viên thực hiện:

Phạm Đăng Hoàng 22520472

Nguyễn Quốc Khánh 22520646

**TP.** **Hồ** **Chí** **Minh,** **tháng** 1 **năm** 2025

1

BÁO CÁO ĐỒ ÁN TRUY VẤN THÔNG TIN ĐA PHƯƠNG TIỆN

[ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH 1](#_Toc187916043)

[MỞ ĐẦU 3](#_Toc187916044)

[CHƯƠNG 1: GIỚI THIỆU 3](#_Toc187916045)

[**1.1.** **Lý** **do** **chọn** **đề** **tài .** 3](#_Toc187916046)

[**1.2.** **Tính** **cấp** **thiết** **của** **vấn** **đề** **nghiên** **cứu.** 3](#_Toc187916047)

[**1.3.** **Mục** **tiêu** **nghiên** **cứu.** 3](#_Toc187916048)

[**1.4.** **Phương pháp nghiên cứu.** 3](#_Toc187916049)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 4](#_Toc187916050)

[**2.1. Giới thiệu chatbot.** 4](#_Toc187916051)

[**2.2. Công nghệ RAG. (Retriaval Augemented Generation)** 4](#_Toc187916052)

[**2.3. Framework Langchain.** 5](#_Toc187916053)

[**2.3.1. Bài toán Query về văn bản.** 6](#_Toc187916054)

[**2.3.2. Hiệu quả sử dụng Langchain:** 7](#_Toc187916055)

[**2.4. Embedding.** 7](#_Toc187916056)

[**2.7.** **FAISS** (**Facebook AI Similarity Search).** 8](#_Toc187916057)

[**2.8.** **Mô hình ngôn ngữ lớn (LLM).** 8](#_Toc187916058)

[CHƯƠNG 3: PHÁT TRIỂN HỆ THỐNG CHATBOT 9](#_Toc187916059)

[**3.** **Thiết kế hệ thống.** 9](#_Toc187916060)

[**3.1. Triển khai chatbot.** 10](#_Toc187916061)

[CHƯƠNG 4: THỬ NGHIỆM VÀ ĐÁNH GIÁ 10](#_Toc187916062)

[**4.1. Thử nghiệm** 10](#_Toc187916063)

[**4.1.1. Kết quả.** 11](#_Toc187916064)

[**4.1.2. Kết luận.** 11](#_Toc187916065)

[TÀI LIỆU THAM KHẢO 11](#_Toc187916066)

|  |  |  |
| --- | --- | --- |
| Công việc | Hoàng | Khánh |
| Nội dung lí thuyết |  |  |
| Code demo |  |  |
| Làm slide |  |  |
| Làm báo cáo |  |  |

# MỞ ĐẦU

Trong những năm gần đây, trí tuệ nhân tạo (AI) đã có những bước phát triển mạnh mẽ, mang lại nhiều thay đổi trong cách chúng ta tương tác với công nghệ. Một trong những ứng dụng nổi bật của AI là phát triển các chatbot, công cụ giúp tự động hóa giao tiếp giữa người dùng và hệ thống, từ đó giảm thiểu thời gian và chi phí cho các tổ chức.

Ứng dụng thực tiễn của chatbot, cùng những thách thức và triển vọng của công nghệ trong tương lai.

# CHƯƠNG 1: GIỚI THIỆU

## **Lý** **do** **chọn** **đề** **tài .**

**Xu hướng phát triển công nghệ AI:** Trong bối cảnh công nghệ trí tuệ nhân tạo (AI) ngày càng phát triển mạnh mẽ, đặc biệt trong lĩnh vực xử lý ngôn ngữ tự nhiên và giao tiếp giữa máy tính và con người, việc phát triển và cải tiến chatbot trở thành một yêu cầu cấp thiết. Đề tài này nhằm nghiên cứu và áp dụng công nghệ mới nhất để nâng cao khả năng tương tác của chatbot.

**Nhu cầu truy vấn thông tin đa phương tiện**: Trong thời đại thông tin hiện nay, người dùng ngày càng yêu cầu những câu trả lời nhanh chóng và chính xác từ nhiều loại hình dữ liệu khác nhau, bao gồm văn bản, hình ảnh, và video. Việc xây dựng một chatbot có khả năng truy vấn thông tin đa phương tiện sẽ đáp ứng nhu cầu cấp thiết này, đồng thời mang lại trải nghiệm người dùng tốt hơn.

## **Tính** **cấp** **thiết** **của** **vấn** **đề** **nghiên** **cứu.**

Thế giới hiện nay đang chứng kiến sự bùng nổ của công nghệ chatbot, với nhu cầu ngày càng cao về giao tiếp tự động và thông minh. Việc tích hợp công nghệ RAG (Retrieval-Augmented Generation) với Langchain sẽ tạo ra chatbot có khả năng truy xuất thông tin một cách chính xác và tự nhiên, đáp ứng nhu cầu tìm kiếm thông tin đa phương tiện một cách hiệu quả.

## **Mục** **tiêu** **nghiên** **cứu.**

* Nghiên cứu và triển khai một chatbot thông minh sử dụng RAG và Langchain.
* Đánh giá hiệu quả hoạt động của chatbot qua các kịch bản truy vấn thông tin đa phương tiện.
* Đề xuất cải tiến cho các ứng dụng thực tiễn của chatbot trong tương lai.

## **Phương pháp nghiên cứu.**

* Nghiên cứu lý thuyết qua tài liệu chuyên ngành.
* Phát triển phần mềm chatbot bằng ngôn ngữ lập trình Python.
* Thực hiện thử nghiệm người dùng và thu thập dữ liệu.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

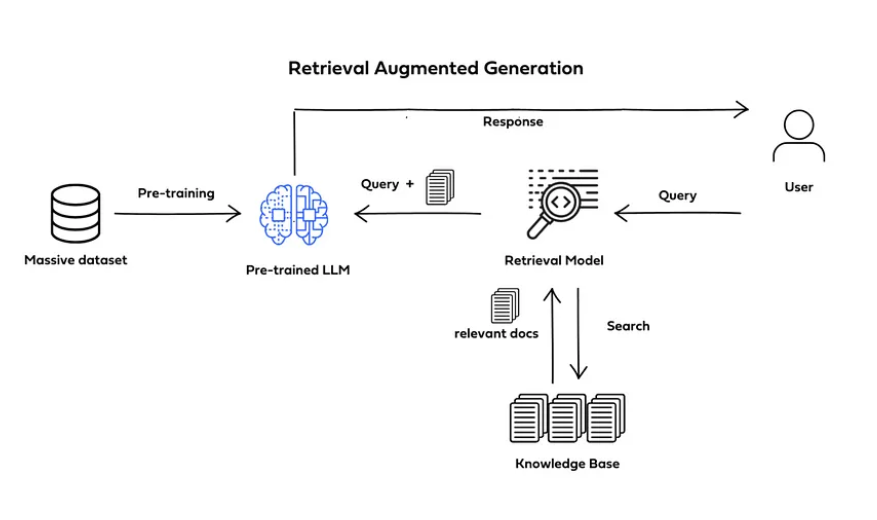
## **2.1. Giới thiệu chatbot.**

Chatbot là chương trình máy tính được thiết kế để mô phỏng cuộc trò chuyện với con người thông qua giao diện nhắn tin. Chatbot có thể hoạt động dựa trên quy tắc đã được lập trình sẵn hoặc áp dụng học máy để cải thiện khả năng hiểu ngôn ngữ tự nhiên. Hiện nay, chatbot được áp dụng rộng rãi trong dịch vụ khách hàng, quản lý thông tin, hỗ trợ học tập và giải trí.

## **2.2. Công nghệ RAG. (Retriaval Augemented Generation)**

**RAG** là một phương pháp kết hợp giữa tìm kiếm thông tin (retrieval) và sinh câu trả lời (generation). Khi một câu hỏi được gửi đến chatbot, RAG sẽ thực hiện hai bước:

* **Truy vấn**: Hệ thống sẽ truy vấn các nguồn thông tin bên ngoài (như cơ sở dữ liệu, tài liệu, hoặc trang web) để tìm kiếm dữ liệu liên quan đến câu hỏi.
* **Sinh câu trả lời**: Sau khi thu thập thông tin, hệ thống sẽ sử dụng mô hình ngôn ngữ để tạo ra câu trả lời tự nhiên và phù hợp với ngữ cảnh.

Sự kết hợp này giúp chatbot không chỉ cung cấp thông tin chính xác mà còn tạo ra những phản hồi tự nhiên, mang tính tương tác cao.

* Massive Dataset: Đây là tập dữ liệu lớn, bao gồm thông tin đa dạng mà mô hình sẽ học từ đó. Dữ liệu này có thể bao gồm văn bản, tài liệu, trang web, và nhiều nguồn khác.
* Pre-training: Giai đoạn này liên quan đến việc huấn luyện mô hình ngôn ngữ lớn (LLM) trên tập dữ liệu lớn để nó có thể hiểu và tạo ra văn bản. Mô hình sẽ học cách dự đoán từ tiếp theo trong văn bản và các mối quan hệ giữa các từ.
* Pre-trained LLM: Mô hình ngôn ngữ đã được huấn luyện trước mang lại khả năng hiểu ngữ nghĩa và sinh văn bản. Điều này giúp nó có thể tương tác với các truy vấn từ người dùng và sản sinh câu trả lời có ngữ cảnh.
* User Query: Khi người dùng đặt câu hỏi, truy vấn này được gửi đến mô hình để xử lý.
* Retrieval Model: Mô hình này chịu trách nhiệm tìm kiếm và lấy tài liệu phù hợp từ Knowledge Base (cơ sở tri thức). Nó sử dụng các thuật toán tìm kiếm dựa trên ngữ nghĩa để tìm ra các tài liệu liên quan nhất đến truy vấn của người dùng.
* Search: Tại bước này, mô hình thực hiện một tìm kiếm trong cơ sở tri thức để xác định các tài liệu phù hợp (relevant docs) có thể giúp trả lời truy vấn.
* Relevant Docs: Các tài liệu được tìm thấy sẽ được gửi lại cho mô hình ngôn ngữ lớn.
* Query + Relevant Docs: Mô hình kết hợp giữa truy vấn của người dùng và các tài liệu liên quan để tạo ra một câu trả lời chính xác và có ngữ cảnh hơn.
* Response: Cuối cùng, mô hình sinh ra câu trả lời dựa trên thông tin mà nó đã truy xuất và phản hồi lại cho người dùng.
* User: Đây là người dùng cuối, người đặt câu hỏi và nhận câu trả lời từ hệ thống.

**Hiệu quả sử dụng RAG** (**Retriaval Augemented Generation):**

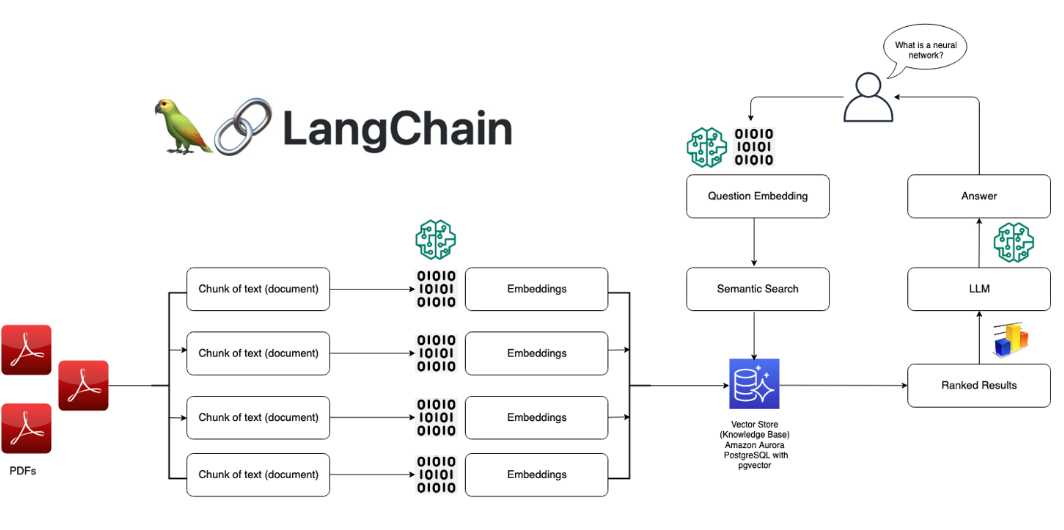
* **Kết hợp truy xuất thông tin và sinh văn bản**: RAG tận dụng sức mạnh của tìm kiếm dữ liệu để bổ sung ngữ cảnh và kiến thức cho mô hình ngôn ngữ, giúp nó không chỉ dựa vào những gì đã học mà còn có thể truy cập thông tin bên ngoài.
* **Cải thiện độ chính xác**:Thông tin được lấy từ các tài liệu liên quan giúp câu trả lời không chỉ chính xác mà còn đáng tin cậy hơn.
* **Khả năng cập nhật thông tin**:Vì hệ thống dựa vào cơ sở dữ liệu và nguồn tri thức bên ngoài, nó có thể phản hồi dựa trên thông tin cập nhật gần nhất mà không cần đào tạo lại mô hình ngôn ngữ.
* **Tăng cường tính ngữ cảnh**:RAG không chỉ trả lời câu hỏi mà còn kết hợp dữ liệu ngữ cảnh liên quan để giải thích rõ ràng và chi tiết hơn.
* **Phản hồi tự nhiên và tương tác**: LLM sử dụng khả năng sinh văn bản để tạo ra câu trả lời giống con người, giúp tăng trải nghiệm người dùng.
* **Hiệu quả với dữ liệu lớn**: Các mô hình truy xuất (retrieval) như Faiss hoặc ChromaDB giúp tìm kiếm nhanh chóng ngay cả với cơ sở dữ liệu cực lớn.

## **2.3. Framework Langchain.**

**Langchain** là một framework mạnh mẽ giúp xây dựng ứng dụng trí tuệ nhân tạo, đặc biệt là trong việc tạo ra các chatbot và các ứng dụng liên quan đến ngôn ngữ. Langchain hỗ trợ tích hợp dễ dàng giữa các mô hình sinh văn bản và mô hình truy vấn, tạo ra một hệ thống linh hoạt và mở rộng. Langchain gồm các thành phần chính:

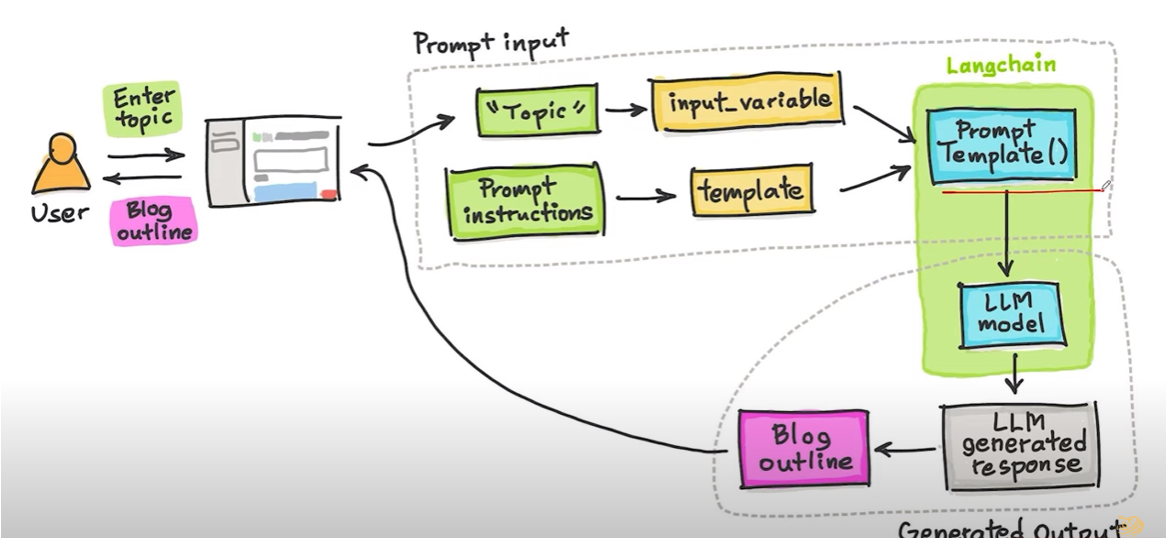
* **Chain:** Là cấu trúc chính cho phép kết nối nhiều bước xử lý khác nhau trong một quy trình, cho phép gửi đầu vào, thực hiện các thao tác, và nhận đầu ra.
* **Prompting**: Cung cấp khả năng tạo và quản lý các prompt (lời nhắc) cho mô hình, giúp điều chỉnh đầu vào cho đúng ngữ cảnh và mục tiêu.
* **Retrieval**: Hỗ trợ tích hợp khung truy vấn cơ sở dữ liệu để lấy thông tin từ nguồn dữ liệu bên ngoài, cải thiện tính chính xác của các câu trả lời.
* **Memory**: Khả năng ghi nhớ thông tin giữa các lần tương tác để tạo ra các cuộc hội thoại tự nhiên hơn.

### **2.3.1. Bài toán Query về văn bản.**



* PDFs: Một tập hợp các tài liệu PDF, đây là nguồn thông tin chính mà chatbot sẽ khai thác.
* Chunk of Text: Tài liệu PDF được chia thành các đoạn văn bản nhỏ hơn (chunks) để dễ dàng xử lý và truy xuất.
* Embeddings: Mỗi đoạn văn bản sẽ được chuyển đổi thành vector embeddings, giúp thể hiện nghĩa của thông tin trong không gian vector. Điều này hỗ trợ trong việc so sánh và tìm kiếm thông tin.
* Question Embedding: Khi người dùng đặt câu hỏi, câu hỏi cũng được chuyển đổi thành vector embedding tương tự.
* Semantic Search: Việc tìm kiếm dựa trên ngữ nghĩa (semantic search) giúp đối chiếu vector embedding của câu hỏi với các vectors của đoạn văn bản để xác định các tài liệu liên quan nhất.
* Vector Store (Knowledge Base): Một cơ sở dữ liệu vector (như Amazon Aurora hoặc PostgreSQL với pgvector) lưu trữ các embeddings, cho phép thực hiện tìm kiếm nhanh chóng và hiệu quả.
* LLM (Large Language Model): Sau khi tìm thấy các đoạn văn bản liên quan, mô hình ngôn ngữ lớn sẽ được sử dụng để tạo ra câu trả lời từ các thông tin đã truy xuất.
* Ranked Results: Cuối cùng, các kết quả được đánh giá và xếp hạng trước khi trả về cho người dùng dưới dạng câu trả lời, đảm bảo rằng thông tin chính xác và có giá trị.

### **2.3.2. Hiệu quả sử dụng Langchain:**



* Load file và xử lí vector DB đơn giản hơn.
* Giao tiếp với LLM model có Prompt Template chỉnh sửa dễ dàng hơn. ( Prompt Template là một đoạn text bao gồm các chỉ dẫn, các yêu cầu, các ví dụ để đưa vào mô hình ngôn ngữ và nó có thể nhận đầu vào là các tham số do người dùng truyền vào).

## **2.4. Embedding.**

**Embedding** là một cách biểu diễn dữ liệu ( văn bản) dưới dạng các **vector số** trong không gian nhiều chiều (multi-dimensional space)

Mục đích:

* Biểu diễn dữ liệu có ý nghĩa về mặt ngữ nghĩa (semantic).
* Giúp so sánh, tìm kiếm và phân tích dữ liệu dễ dàng hơn.

+ Sử dụng : ***dangvantuan/vietnamese-document-embedding*** *(hugging face) (up to 8096 tokens)*

* 1. **Vector DB ( Vector Database)**

LLM không thể nào nhìn được toàn bộ văn bản (giới hạn input size).

=> Cần chia nhỏ văn bản thành những đoạn văn bản, mỗi đoạn sẽ được đưa vào model embedding để sinh ra vector đặc trưng và các vector đặc trưng sẽ được đưa vào DB gọi là vector DB.

A screenshot of a computer

Description automatically generated

* 1. **Chunking**

Chunking là kỹ thuật chia nhỏ văn bản thành các đoạn (chunk) có nghĩa để dễ dàng tìm kiếm và xử lý. Việc chia văn bản giúp:

* Tối ưu hóa quá trình truy vấn thông tin bằng cách hạn chế phạm vi dữ liệu cần tìm kiếm.
* Tăng tốc độ truy xuất và giảm tải cho mô hình ngôn ngữ khi sinh câu trả lời.
* Giảm thiểu việc vượt quá max token.

Sử dụng chunk size = 500.

overlap = 10.

## **FAISS** (**Facebook AI Similarity Search).**

FAISS là thư viện được thiết kế để giải quyết vấn đề tìm kiếm và truy vấn dữ liệu tương tự. Nó hỗ trợ:

* Lưu trữ và tìm kiếm hiệu quả trong không gian vectơ.
* Duy trì tốc độ cao trong việc truy vấn và tìm kiếm gần đúng, phù hợp với các ứng dụng sử dụng chunk dữ liệu.

## **Mô hình ngôn ngữ lớn (LLM).**

Mô hình ngôn ngữ lớn (LLM) là một loại mô hình học máy được huấn luyện trên tập dữ liệu văn bản khổng lồ để thực hiện các nhiệm vụ xử lý ngôn ngữ tự nhiên (NLP). Chúng có khả năng tạo, hiểu, và tương tác với văn bản một cách gần giống con người.

Sử dụng : call api ***llama-3.3-70b-versatile*** ( langchain groq )

**Cách thức hoạt động:**

+ Sau khi lấy được 3 dữ liệu tương đồng nhất với question.

+ Sử dụng cosin semilarity để truy vấn được một kết quả.

+ Sau đó đưa kết quả và question để tạo một ngữ cảnh đưa vào LLM generat

ra answer.

**Vai trò của LLM trong Chatbot:**

+ Hiểu ngôn ngữ tự nhiên (NLP): Giải mã ý nghĩa câu hỏi từ người dùng.

+ Sinh câu trả lời tự nhiên: Đưa ra phản hồi giống con người.

+ Tích hợp đa ngữ cảnh: Hiểu được nội dung câu hỏi và ngữ cảnh hội thoại.

**Ứng dụng của LLM trong Chatbot:**

+ Độ chính xác cao: Trích xuất thông tin và đưa ra câu trả lời chính xác hơn.

+ Khả năng mở rộng: Hỗ trợ đa ngôn ngữ và nhiều lĩnh vực.

+ Tương tác tự nhiên: Hội thoại linh hoạt và mượt mà hơn.

**Hạn chế của LLM:**

+ Chi phí tính toán cao: Cần tài nguyên lớn để huấn luyện và vận hành.

+ Rủi ro thông tin sai lệch: Có thể tạo nội dung không chính xác nếu không

được kiểm soát tốt.

+ Phụ thuộc vào dữ liệu đầu vào: Kết quả phụ thuộc chất lượng và tính đầy đủ

của dữ liệu huấn luyện.

# CHƯƠNG 3: PHÁT TRIỂN HỆ THỐNG CHATBOT

## **Thiết kế hệ thống.**

**A diagram of a software application

Description automatically generated**

* Data Preparing (Chuẩn bị dữ liệu):
* Dữ liệu ban đầu được xử lý thông qua các bước như Embedding (tạo vector biểu diễn), Text Splitter (chia nhỏ văn bản).
* Các công cụ như Faiss, ChromaDB, hoặc các framework tương tự có thể được sử dụng trong bước này.
* Kết quả của giai đoạn này là một tập hợp dữ liệu đã được chuyển đổi thành các vector để sử dụng trong cơ sở dữ liệu vector.
* Create Vector DB (Tạo cơ sở dữ liệu vector):
* Sau khi dữ liệu đã được xử lý, nó sẽ được lưu trữ trong một Cơ sở dữ liệu vector.
* Đây là nơi lưu trữ các vector biểu diễn dữ liệu, cho phép truy vấn nhanh chóng dựa trên độ tương đồng vector.
* User Query (Người dùng đặt câu hỏi):
* Người dùng đưa ra truy vấn hoặc câu hỏi.
* Truy vấn này được đưa qua một Prompt Template (mẫu gợi ý) để chuẩn bị phù hợp trước khi chuyển đến LLM.
* LLM to Understand Query (LLM hiểu truy vấn):
* Mô hình ngôn ngữ lớn (LLM) được sử dụng để hiểu ý định của truy vấn và chuyển nó thành một dạng phù hợp để tìm kiếm trong cơ sở dữ liệu vector.
* Query DB (Truy vấn cơ sở dữ liệu):
* Dựa trên thông tin từ LLM, cơ sở dữ liệu vector được truy vấn để tìm ra các tài liệu liên quan.
* Các tài liệu này có thể là các phần văn bản, dữ liệu hoặc thông tin gần giống với ý định truy vấn của người dùng.
* LLM to Generate Answer (LLM tạo câu trả lời):
* Các tài liệu truy xuất từ cơ sở dữ liệu vector được LLM sử dụng để tạo câu trả lời hoàn chỉnh.
* Quá trình này kết hợp thông tin từ dữ liệu liên quan và khả năng tạo ngôn ngữ của LLM.
* User Response (Phản hồi người dùng):
* Người dùng đặt câu hỏi → truy vấn được hiểu bởi LLM → tìm kiếm dữ liệu trong cơ sở dữ liệu vector → LLM sử dụng dữ liệu tìm được để tạo câu trả lời → phản hồi được gửi lại cho người dùng.

**Tóm tắt luồng chính:**

* Dữ liệu được xử lý và lưu trữ trong cơ sở dữ liệu vector.
* Người dùng đặt câu hỏi → truy vấn được hiểu bởi LLM → tìm kiếm dữ liệu trong cơ sở dữ liệu vector → LLM sử dụng dữ liệu tìm được để tạo câu trả lời → phản hồi được gửi lại cho người dùng.

## **3.1. Triển khai chatbot.**

Link model: https://www.kaggle.com/code/khanhnq1301/chatbot-rag-langchain

# CHƯƠNG 4: THỬ NGHIỆM VÀ ĐÁNH GIÁ

## **4.1. Thử nghiệm**

Mục tiêu thử nghiệm: Đánh giá độ chính xác, tốc độ phản hồi, và khả năng tương tác của chatbot.



### **4.1.1. Kết quả.**

Hệ thống chatbot với khả năng cung cấp câu trả lời phù hợp với câu hỏi từ người dùng. Sử dụng các công nghệ tiên tiến như RAG, LangChain và mô hình ngôn ngữ lớn (LLM). Tuy nhiên vẫn cần cải thiện thuật toán để nâng cao hiệu suất và độ chính xác, và mở rộng khả năng tích hợp dữ liệu từ nhiều nguồn khác nhau.

### **4.1.2. Kết luận.**

Việc phát triển một hệ thống chatbot mạnh mẽ đã mang lại nhiều lợi ích cho trải nghiệm người dùng và khả năng đáp ứng thông tin. Bằng cách kết hợp các công nghệ tiên tiến như Retrieval-Augmented Generation (RAG), chunking, FAISS, và các mô hình ngôn ngữ lớn (LLM), đã bước đầu xây dựng một chatbot có khả năng cung cấp các câu trả lời chính xác và tự nhiên dựa trên thông tin có sẵn.

Quá trình phát triển và thử nghiệm cũng đã chỉ ra rằng việc tối ưu hóa trải nghiệm người dùng qua giao diện web, cũng như khả năng mở rộng hệ thống, là rất quan trọng. Các thử nghiệm cho thấy chatbot đã đạt được độ chính xác và thời gian phản hồi tốt, tạo ra một trải nghiệm tương tác mượt mà và thân thiện. Tuy nhiên, vẫn còn nhiều cơ hội để cải tiến và tối ưu hóa các thuật toán để nâng cao khả năng hoạt động của chatbot.

# TÀI LIỆU THAM KHẢO

[1] [irbookonlinereading.pdf](https://courses.uit.edu.vn/pluginfile.php/678071/mod_folder/content/0/irbookonlinereading.pdf?forcedownload=1).

[2] [csdlulti-151212085744.pdf](https://courses.uit.edu.vn/pluginfile.php/678071/mod_folder/content/0/csdlulti-151212085744.pdf?forcedownload=1).

[3] Retrieval-Augmented Generation (RAG) – Facebook AI.

[4] Building a RAG Model with Langchain.

[5] Chat OpenAI.