**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A blue and white sign with red text

Description automatically generated**

**TIỂU LUẬN TỐT NGHIỆP**

**XÂY DỰNG HỆ THỐNG CHATBOT THÔNG MINH HỖ TRỢ TÌM KIẾM TÀI LIỆU HỌC TẬP**

**Thành phố Hồ Chí Minh - 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A blue and white sign with red text

Description automatically generated**

**XÂY DỰNG HỆ THỐNG CHATBOT THÔNG MINH HỖ TRỢ TÌM KIẾM TÀI LIỆU HỌC TẬP**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | Ths. Ma Ngân Giang |
| **Sinh viên thực hiện:** | 47.01.104.132 – Nguyễn Phan Trà My |
| **Lớp:** | 47.01.CNTT.c |
| **Niên khóa:** | 2024 – 2025 |

**Thành phố Hồ Chí Minh - 2024**

# LỜI CẢM ƠN

Trong quá trình thực hiện và hoàn thành tiểu luận này, em xin gửi lời cảm ơn chân thành và sâu sắc đến cô ThS. Ma Ngân Giang, người đã tận tâm hướng dẫn, chỉ bảo em trong suốt quá trình nghiên cứu và viết bài. Sự tận tụy và nhiệt tình của thầy/cô không chỉ giúp em hoàn thành tốt công việc mà còn mở rộng kiến thức và khả năng tư duy của em.

Em cũng xin bày tỏ lòng biết ơn đến Ban Giám hiệu và toàn thể các thầy cô tại Khoa Công nghệ thông tin, những người đã tạo điều kiện và cung cấp nguồn tài liệu học tập quý báu, giúp em hoàn thiện bài tiểu luận này. Sự hỗ trợ của các thầy cô là động lực lớn để em nỗ lực không ngừng trong quá trình học tập và nghiên cứu.

Cuối cùng, em xin gửi lời cảm ơn chân thành đến gia đình, bạn bè, những người luôn bên cạnh, động viên và chia sẻ trong suốt quá trình thực hiện bài tiểu luận. Sự ủng hộ và khích lệ của mọi người là nguồn động viên to lớn để em vượt qua những khó khăn và thử thách.

Mặc dù đã hết sức cố gắng, tiểu luận chắc chắn không tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý từ thầy/cô và bạn đọc để hoàn thiện bài nghiên cứu của mình.

Em xin chân thành cảm ơn.

# MỤC LỤC

[LỜI CẢM ƠN 3](#_Toc186201041)

[MỤC LỤC 4](#_Toc186201042)

[BẢNG CÁC KÝ HIỆU, CHỮ VIẾT TẮT 6](#_Toc186201043)

[DANH MỤC CÁC BẢNG BIỂU 7](#_Toc186201044)

[DANH MỤC CÁC HÌNH VẼ 8](#_Toc186201045)

[TÓM TẮT 9](#_Toc186201046)

[CHƯƠNG 1: MỞ ĐẦU 10](#_Toc186201047)

[1.1. Lý do chọn đề tài 10](#_Toc186201048)

[1.2. Mục tiêu nghiên cứu 10](#_Toc186201049)

[1.3. Đối tượng và phạm vi nghiên cứu 10](#_Toc186201050)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 11](#_Toc186201051)

[2.1. Tổng quan về chatbot: 11](#_Toc186201052)

[2.1.1. Chatbot là gì? 11](#_Toc186201053)

[2.1.2. Chatbot hoạt động như thế nào? 11](#_Toc186201054)

[2.1.3. Ứng dụng của chatbot trong giáo dục: 12](#_Toc186201055)

[2.2. Công nghệ AI trong chatbot: 12](#_Toc186201056)

[2.3. Vector Embeddings 13](#_Toc186201057)

[2.4. Tạo tăng cường truy suất RAG: 13](#_Toc186201058)

[2.5. Tổng quan về chương trình giáo dục phổ thông 2018 14](#_Toc186201059)

[2.6. Công nghệ sử dụng: 15](#_Toc186201060)

[2.6.1. Ngôn ngữ lập trình 15](#_Toc186201061)

[CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG 16](#_Toc186201062)

[3.1. Tài liệu học tập: 16](#_Toc186201063)

[3.2. Tiền xử lý dữ liệu: 16](#_Toc186201064)

[3.3. Tạo vector embeddings: 16](#_Toc186201065)

[3.4. Lưu trữ và tổ chức dữ liệu: 16](#_Toc186201066)

[3.5.1. Kiến trúc tổng thể: 17](#_Toc186201067)

[3.5.2. Các thành phần chính: 17](#_Toc186201068)

[3.5.3. Luồng hoạt động chính của hệ thống: 18](#_Toc186201069)

[CHƯƠNG 4: THIẾT KẾ HỆ THỐNG 19](#_Toc186201070)

[4.1. Kiến trúc hệ thống: 19](#_Toc186201071)

[4.1.1. Sơ đồ usecase: 19](#_Toc186201072)

[4.2. Thiết kế cơ sở dữ liệu: 19](#_Toc186201073)

[4.3. Thiết kế API: Thiết kế các endpoint API và tương tác với các thành phần khác. 20](#_Toc186201074)

[4.4. Tích hợp GPT-4o và LlamaIndex: Mô tả quá trình tích hợp các công nghệ AI. 20](#_Toc186201075)

[CHƯƠNG 5: KIỂM THỬ, ĐÁNH GIÁ HỆ THỐNG VÀ KẾT LUẬN 21](#_Toc186201076)

[5.1. Kiểm thử chức năng: Phương pháp kiểm thử đơn vị và tích hợp. 21](#_Toc186201077)

[5.2. Kiểm thử hiệu năng: Thử nghiệm về hiệu suất và tốc độ của hệ thống. 21](#_Toc186201078)

[5.3. Đánh giá kết quả: Mức độ chính xác và hiệu quả của hệ thống chatbot. 21](#_Toc186201079)

[5.4. Kết luận: Tổng kết những kết quả chính đạt được. 21](#_Toc186201080)

[5.5. Hướng phát triển trong tương lai: Các đề xuất cải tiến và phát triển sau này. 21](#_Toc186201081)

[KẾT LUẬN 24](#_Toc186201082)

[DANH MỤC TÀI LIỆU THAM KHẢO 25](#_Toc186201083)

[PHỤ LỤC 26](#_Toc186201084)

# BẢNG CÁC KÝ HIỆU, CHỮ VIẾT TẮT

**Viết tắt Tiếng Anh Tiếng Việt**

NLP Neuro-Linguistic Programming Lập trình ngôn ngữ tư duy

ML Neuro-Linguistic Programming Lập trình ngôn ngữ tư duy

NLU Neuro-Linguistic Programming Lập trình ngôn ngữ tư duy

RAG Neuro-Linguistic Programming Lập trình ngôn ngữ tư duy

FAQ Neuro-Linguistic Programming Lập trình ngôn ngữ tư duy

OCR Optical Character Recognition Nhận dạng kí tự quang học

# DANH MỤC CÁC BẢNG BIỂU

# DANH MỤC CÁC HÌNH VẼ

# TÓM TẮT

Dự án "Xây dựng hệ thống chatbot thông minh hỗ trợ tìm kiếm tài liệu học tập" tập trung phát triển một công cụ chatbot sử dụng mô hình Retrieval Augmented Generation (RAG) kết hợp với GPT-4o để hỗ trợ tìm kiếm tài liệu học tập từ chương trình giáo dục phổ thông 2018. Hệ thống được thiết kế nhằm cải thiện việc truy xuất thông tin học tập, giúp người dùng có thể tìm kiếm và truy cập tài liệu nhanh chóng và hiệu quả. Các tính năng chính bao gồm tìm kiếm tài liệu theo từ khóa, phân loại theo chủ đề, và đề xuất tài liệu dựa trên lịch sử tìm kiếm.

# MỞ ĐẦU

## Lý do chọn đề tài

Trong thời đại công nghệ số, việc truy cập thông tin học tập một cách nhanh chóng và hiệu quả là một nhu cầu thiết yếu đối với học sinh, sinh viên, và giáo viên. Tuy nhiên, việc tìm kiếm tài liệu học tập phù hợp giữa hàng loạt thông tin có thể gặp nhiều khó khăn, đặc biệt khi người dùng cần tìm kiếm theo một chủ đề cụ thể hoặc theo yêu cầu riêng biệt.

Sự phát triển của trí tuệ nhân tạo (AI) và các hệ thống chatbot thông minh đã mở ra những tiềm năng mới trong việc hỗ trợ tìm kiếm và truy xuất thông tin học tập một cách dễ dàng. Một hệ thống chatbot không chỉ giúp người dùng tìm kiếm nhanh chóng, mà còn có khả năng hiểu được các câu hỏi phức tạp và đưa ra các gợi ý tài liệu liên quan, từ đó nâng cao hiệu quả học tập và giảng dạy.

Vì những lý do trên, đề tài "Xây dựng hệ thống chatbot thông minh hỗ trợ tìm kiếm tài liệu học tập" được chọn với mục tiêu xây dựng một công cụ hỗ trợ hiệu quả cho học sinh, sinh viên, giáo viên trong việc tìm kiếm và truy cập các tài liệu học tập.

## Mục tiêu nghiên cứu

Phát triển một hệ thống chatbot thông minh có khả năng hỗ trợ người dùng trong việc tìm kiếm và truy cập tài liệu học tập một cách nhanh chóng và hiệu quả.

## Đối tượng và phạm vi nghiên cứu

* Đối tượng: Sinh viên, giáo viên, và những người có nhu cầu tìm kiếm tài liệu học tập.
* Phạm vi: Hệ thống sẽ tập trung vào tài liệu học tập của chương trình giáo dục phổ thông 2018

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## Tổng quan về chatbot:

### Chatbot là gì?

Chatbot là chương trình máy tính mô phỏng lại các đoạn hội thoại của con người, văn bản hoặc lời nói, cho phép con người tương tác với các thiết bị kỹ thuật số như thể họ đang giao tiếp với một người thật. Chatbot có thể đơn giản như các chương trình cơ bản trả lời một truy vấn đơn giản bằng một phản hồi một dòng hoặc phức tạp như trợ lý kỹ thuật số học hỏi và phát triển để cung cấp mức độ cá nhân hóa ngày càng tăng khi chúng thu thập và xử lý thông tin.

### Chatbot hoạt động như thế nào?

Với sự thúc đẩy của AI, các quy tắc tự động, xử lý ngôn ngữ tự nhiên (NLP) và học máy (ML) thì chatbot xử lý dữ liệu để đưa ra phản hồi cho mọi loại yêu cầu.

Có 2 loại chatbot chính:

* **Task-oriented (declarative) chatbots (Chatbot khai báo)** những chương trình đơn lẻ tập trung vào thực hiện một chức năng. Bằng cách cách sử dụng các quy tắc, NLP và một chút ít ML, Task-oriented (declarative) chatbots tạo ra các phản hồi tự động nhưng mang tính hội thoại, trả lời các câu hỏi hoặc yêu cầu từ người dùng. Các tương tác với chatbot định hướng nhiệm vụ thường rất cụ thể và có cấu trúc rõ ràng, phù hợp nhất với các chức năng hỗ trợ và dịch vụ. Một cách dễ hiểu, đây có thể được xem như phiên bản tương tác nâng cao của các câu hỏi thường gặp (FAQ) nhưng được tích hợp khả năng phản hồi linh hoạt hơn.
* **Data-driven and predictive (conversational) chatbots (Chatbot dự đoán)** thường được xem là trợ lý ảo. Đây loại chatbot tiên tiến với khả năng tương tác và cá nhân hóa cao hơn nhiều so với các chatbot chỉ thực hiện nhiệm vụ đơn giản. Các chatbot này sử dụng các công nghệ hiện đại như hiểu ngôn ngữ tự nhiên (NLU), xử lý ngôn ngữ tự nhiên (NLP) và học máy (ML) để không ngừng học hỏi và cải thiện trong quá trình hoạt động. Nhờ vậy, chúng có khả năng nhận biết ngữ cảnh, áp dụng trí tuệ dự đoán và phân tích để cá nhân hóa trải nghiệm người dùng dựa trên hồ sơ và hành vi trước đây. Chatbot này không chỉ hiểu sâu về ngữ cảnh mà còn có thể học hỏi sở thích của người dùng theo thời gian, từ đó đưa ra các gợi ý phù hợp hoặc thậm chí dự đoán nhu cầu trong tương lai. Ngoài ra, chúng còn có thể chủ động khởi xướng cuộc trò chuyện, chẳng hạn như nhắc nhở, gợi ý hoặc cung cấp thông tin hữu ích.

### Ứng dụng của chatbot trong giáo dục:

Chatbot hiện đang được ứng dụng rộng rãi trong nhiều lĩnh vực và trong số đó không thể thiếu giáo dục. Nhờ khả năng hỗ trợ học tập thông minh, tương tác tự nhiên và cung cấp thông tin một cách nhanh chóng mà chatbot được ứng dụng rộng rãi trong lĩnh vực giáo dục.

Ứng dụng chatbot vào giáo dục thì nổi bật với tính năng giải đáp câu hỏi trong học tập. thông qua tích công nghệ xử lý ngôn ngữ tự nhiên (NLP) và trí tuệ nhân tạo, chatbot có thể hiểu được ngữ cảnh của các câu hỏi từ học sinh và sinh viên, từ đó đưa ra câu trả lời chính xác và chi tiết. Không chỉ dừng lại ở việc trả lời các câu hỏi đơn giản, chatbot còn có khả năng giải thích các vấn đề phức tạp, cung cấp ví dụ minh họa và hướng dẫn từng bước để người học hiểu rõ hơn.

## Công nghệ AI trong chatbot:

GPT-4o là mô hình biến đổi được đào tạo trước đa phương thức và đa ngôn ngữ được phát triển bởi OpenAI, một tổ chức nổi tiếng trong lĩnh vực nghiên cứu trí tuệ nhân tạo. Chữ “o” trong GPT-4o là viết tắt của omni và nhấn mạnh rằng GPT-4o là một mô hình AI đa phương thức với khả năng âm thanh và hình ảnh.

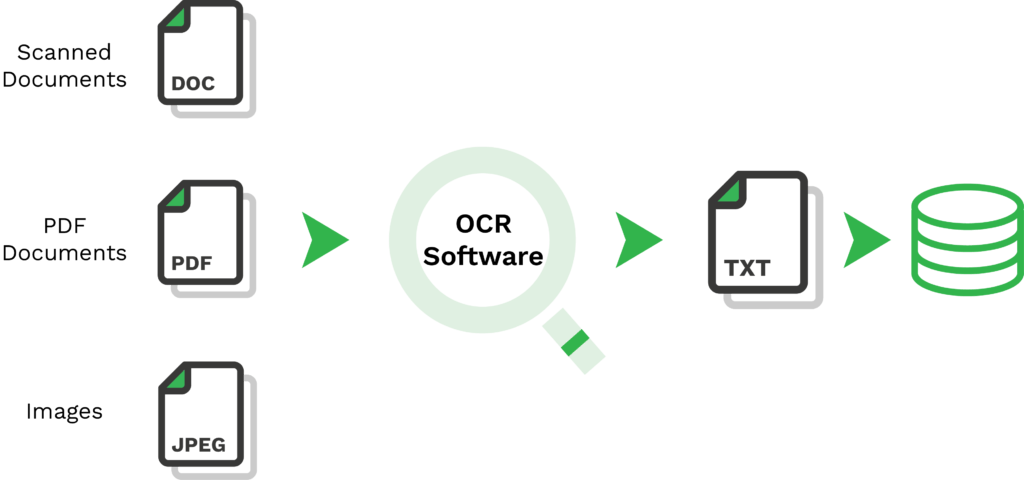
GPT-4o có chức năng trả lời câu hỏi, phân tích và tóm tắt tài liệu, phân tích dữ liệu, phân tích tình cảm, …

Nhờ kết hợp các công nghệ tiên tiến như xử lý ngôn ngữ tự nhiên (NLP), học sâu (Deep Learning), và học tăng cường (Reinforcement Learning), GPT-4o có khả năng hiểu sâu ngữ cảnh của các cuộc hội thoại mà model GPT-4o có thể đưa ra những phản hồi có tính chính xác cao, phù hợp và có giá trị.

## Nhận dạng ký tự quang học OCR - *Optical Character Recognition*

OCR là quá trình chuyển đổi kí tự văn bản trong hình ảnh thành định dạng văn bản có thể đọc được bằng máy.

OCR hoạt động bằng cách cho máy quét đọc tài liệu và chuyển đổi chúng thành dữ liệu nhị phân, phần mềm OCR sẽ phân tích hình ảnh được quét để phân vùng sáng thành nền của văn bản còn vùng tối là các kí tự văn bản. Sau đó phần mềm OCR sẽ làm sạch hình ảnh và loại bỏ lỗi để chuẩn bị cho việc đọc văn bản. Tiếp đến là Sử dụng thuật toán So khớp mẫu (**Pattern matching**) hoặc Trích xuất đặc điểm (**Feature extraction**) để nhận dạng văn bản. Cuối cùng hệ thống sẽ chuyển đổi dữ liệu văn bản thành một tập tin được lưu trên máy tính.



Hình 2.1- Cách hoạt động của chương trình OCR

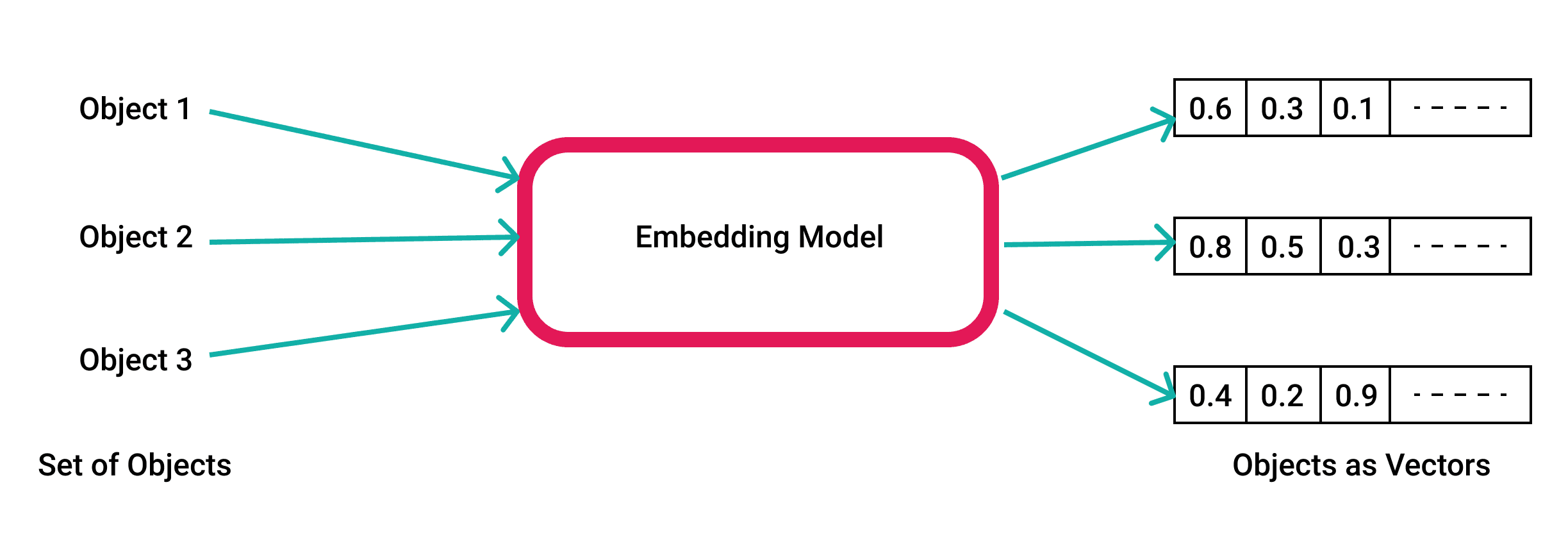
## Vector Databases và Vector Embeddings:

Vector Databases là hệ thống được thiết kế chuyên sử dụng để lưu trữ và truy xuất dữ liệu vector.

Vector embeddings là một kỹ thuật trong học máy và xử lý ngôn ngữ tự nhiên (NLP) nhằm chuyển đổi các đối tượng (như từ, câu, hay tài liệu) thành các vector số (dãy số). Mục tiêu của việc sử dụng vector embeddings là biểu diễn các đối tượng này trong một không gian vector có nghĩa, giúp các thuật toán máy học dễ dàng hơn trong việc xử lý và so sánh.

Vector embeddings, từ bao hàm, câu và danh mục tài liệu đều là nền tảng cơ bản trong NLP cho các nhiệm vụ như phân tích tình cảm và tóm tắt tài liệu. Có các kĩ thuật như Word2Vec, GloVe, BERT và các dịch vụ như OpenAI cung cấp nhiều cách tiếp cận khác nhau để tạo ra các vector embeddings, nâng cao khả năng ứng dụng của chúng trong nhiều nhiệm vụ trong xử lý ngôn ngữ tự nhiên.

Vector Embeddings thường được ứng dụng trong dịch thuật, chatbot, phân tích dữ liệu cho nghiên cứu thị trườngvà hệ thống đề xuất trong thương mại điện tử. Chúng đóng vai trò trong giải quyết các thách thức như khả năng mở rộng hoặc nắm bắt sở thích phức tạp của người dùng và có thể rất hiệu quả trong việc tạo điều kiện học tập năng động để thích ứng với các xu hướng thay đổi.



Hình 2.2- Quy Trình Biểu Diễn Đối Tượng Dưới Dạng Vector Bằng Mô Hình Embedding

## Tạo tăng cường truy suất RAG (Retrieval-Augmented Generation):

### RAG là gì?

RAG là quá trình tối ưu hóa đầu ra của một mô hình ngôn ngữ lớn, vì vậy nó tham khảo một cơ sở kiến thức có thẩm quyền bên ngoài các nguồn dữ liệu đào tạo của nó trước khi tạo phản hồi. Mô hình ngôn ngữ lớn (LLM) được đào tạo trên khối lượng dữ liệu khổng lồ và có sử dụng hàng tỷ tham số để tạo ra đầu ra ban đầu cho các nhiệm vụ như trả lời câu hỏi, dịch ngôn ngữ và hoàn thành câu. RAG mở rộng các khả năng vốn đã mạnh mẽ của LLM đến các miền cụ thể hoặc cơ sở kiến thức nội bộ của tổ chức, tất cả mà không cần đào tạo lại mô hình. Đây là một cách tiếp cận hiệu quả về chi phí để cải thiện đầu ra LLM, để nó vẫn phù hợp, chính xác và hữu ích trong nhiều bối cảnh khác nhau.

### Tại sao RAG lại quan trọng?

LLM là công nghệ của trí truệ nhân tạo, có nhiệm vụ hỗ trợ chatbot và các ứng dụng của xử lý ngôn ngữ tự nhiên với mục tiêu là tạo ra các con bot có thể trả lời bất kì câu hỏi nào của người dùng trong nhiều bối cảnh khác nhau bằng cách tham chiếu chéo các nguồn kiến thức mà nó có thể truy cập được. nhưng không may, bản chất công nghệ LLM tạo ra này cũng không thể dự đoán được câu trả lời của LLM. Bên cạnh đó, dữ liệu đào tạo LLM là ở dạng tĩnh và lượng kiến thức mà nó chứa sẽ có bị bị cũ đi.

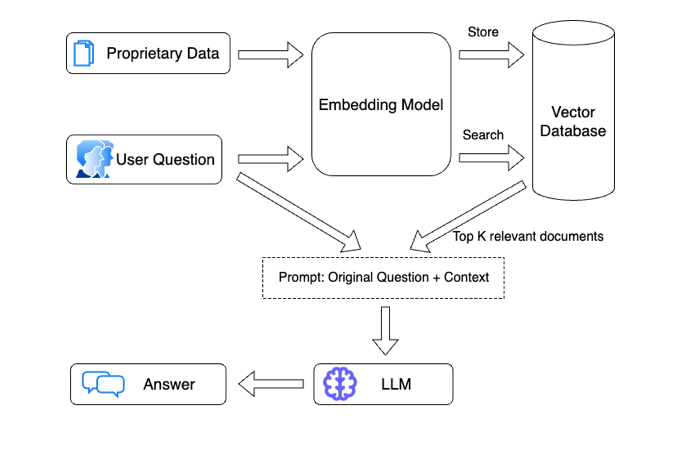
Từ đây ta có các nhược điểm của LLM là:

* Nếu nó không có câu trả lời cho câu hỏi được đề ra thì nó sẽ đưa thông tin sai lệch
* Đưa ra những thông tin lỗi thời và có ý chung chung khi người dùng mong đợi một câu tra lời chính xác và thông tin đó còn mới.
* Tạo các câu trả lời từ những nguồn không rõ ràng.
* Tạo ra những phản hồi không chính xác do nắm sai thuật ngữ hay trong nhiều nguồn thông tin khác nhau nhưng lại sử dụng một thuạt ngữ để nói về những điều khác nhau.

Từ đây, mô hình RAG được xây ra để thực hiện những điều hạn chế của LLM. Nó chuyển hướng LLM để truy xuất thông tin liên quan từ các nguồn kiến thức có thẩm quyền, đã xác định trước. Các tổ chức có quyền kiểm soát tốt hơn đối với đầu ra văn bản đã tạo và người dùng nắm được thông tin chi tiết về cách LLM tạo phản hồi.

Mô hình RAG trải qua các giai đoạn sau:

* Tạo dữ liệu bên ngoài: Dữ liệu bên ngoài là dữ liệu mới bên ngoài của tập dữ liệu đào tạo ban đầu của LLM. Dữ liệu có thể tồn tại ở nhiều định dạng khác nhau như tệp, bản ghi cơ sở dữ liệu hoặc văn bản dạng dài. Một kỹ thuật AI khác, được gọi là mô hình ngôn ngữ nhúng, chuyển đổi dữ liệu thành các dạng trình bày số và lưu trữ nó trong cơ sở dữ liệu véc-tơ. Quá trình này tạo ra một thư viện kiến thức mà các mô hình AI tạo sinh có thể hiểu được.
* Truy xuất thông tin liên quan: Truy vấn người dùng được chuyển đổi thành dạng trình bày véc-tơ và được mang đi so sánh với cơ sở dữ liệu véc-tơ. Sự liên quan được tính toán và thiết lập bằng các phép tính và dạng trình bày véc-tơ toán học.
* Tăng cường lời nhắc LLM: mô hình RAG tăng cường đầu vào của người dùng (hoặc lời nhắc) bằng cách thêm dữ liệu truy xuất liên quan trong ngữ cảnh. Bước này sử dụng các kỹ thuật tạo câu lệnh để giao tiếp hiệu quả với LLM. Với lời nhắc tăng cường, các mô hình ngôn ngữ lớn có thể tạo ra câu trả lời chính xác cho các truy vấn của người dùng



Hình 2.3- Cách mô hình RAG hoạt động

## Tổng quan về chương trình giáo dục phổ thông 2018

Đây là chương trình giáo dục mới nhằm đáp ứng yêu cầu đổi mới văn bản và toàn diện giáo dục với trọng tâm là phát triển phẩm chất năng lực.

Chương trình giáo dục phổ thông cụ thể hoá mục tiêu giáo dục phổ thông, giúp học sinh làm chủ kiến thức phổ thông, biết vận dụng hiệu quả kiến thức, kĩ năng đã học vào đời sống và tự học suốt đời, có định hướng lựa chọn nghề nghiệp phù hợp, biết xây dựng và phát triển hài hoà các mối quan hệ xã hội, có cá tính, nhân cách và đời sống tâm hồn phong phú, nhờ đó có được cuộc sống có ý nghĩa và đóng góp tích cực vào sự phát triển của đất nước và nhân loại.

## Công nghệ sử dụng:

### LlamaIndex:

Là một khung điều phối giúp tích hợp dữ liệu với các mô hình AI để trả lời câu hỏi cảu người dùng

Ưu điểm:

* Hỗ trợ nhiều nguồn dữ liệu (cơ sở dữ liệu, văn bản, API, dữ liệu phi cấu trúc
* Tìm kiếm thông tin hiệu quả dựa trên độ tương đồng vector.
* Tự động hóa tiền xử lý và tối ưu lưu trữ dữ liệu.



Hình 2.4- LlamaIndex logo

### Python:

Python là ngôn ngữ lập trình được sử dụng rộng rãi trong các ứng dụng web, máy học (ML) và khoa học dữ liệu. Đây là ngôn ngữ lập trình dễ học, có thể chạy trên nhiều nền tảng khác nhau nên đây là ngôn ngữ lập trình chính cho backend. Với thư viện phong phú giúp hỗ trợ các công cụ xử lý ngôn ngữ tự nhiên (NLP), học máy (ML) và trí tuệ nhân tạo như spaCy, OpenAI, LlamaIndex, FastAPI, SQLAlchemy, Pandas, NumPy, …

Ưu điểm

* Dễ học và dễ sử dụng
* Thư viện phong phú
* Ứng dụng được trong nhiều lĩnh vực



Hình 2.5- Python logo

### React:

React là thư viện JavaScript mã nguồn mở, được dùng để xây dựng giao diện cho người dùng.

Ưu điểm:

* Nhanh và dễ học
* Tính năng linh hoạt
* Phát triển ứng dụng với cung cụ mạnh mẽ

A blue and black symbol

Description automatically generated

Hình 2.6- React logo

### PostgreSQL:

PostgreSQL là hệ thống quản trị cơ sở dữ liệu quan hệ và đối tượng (object-relational database management system) miễn phíu và nguồn mở tiến tiến nhất hiện nay.

Ưu điểm:

* Có các tính năng và tiện ích mở rộng phông phú
* Giao tiếp với máy chủ cơ sở dữ liệu bằng cách sử dụng các đối tượng trong mã của chúng.



Hình 2.7- PostgreSQL logo

### Các công nghệ khác:

* Thư viện Tesseract OCR (pytesseract): Đây là công cụ OCR mạnh mẽ hỗ trợ nhiều loại ngôn ngữ bao gồm cả tiếng Việt.
* OpenAI Embedding API (text-embedding-ada-002): Mô hình nhúng giúp chuyển đổi văn bản thành vector.
* FastAPI là 1 web framework hiện đại, hiệu năng cao để xây dựng APIs voiwsw Python.

# PHÂN TÍCH HỆ THỐNG

## Phân tích yêu cầu:

### Yêu cầu chức năng:

“Hệ thống chatbot thông minh hỗ trợ tìm kiếm tài liệu học tập” gồm có các chức năng:

* Tương tác thông minh: Chatbot có khả năng hiểu và phản hồi các câu hỏi của người dùng liên quan đến tài liệu học tập.
* Tìm kiếm tài: Hỗ trợ người dùng tìm kiếm tài liệu dựa trên từ khóa, danh mục hoặc chủ đề.
* Gợi ý tài liệu: Gợi ý tài liệu liên quan dựa trên lịch sử tìm kiếm và sở thích của người dùng.
* Cung cấp thông tin chi tiết: Cung cấp thông tin chi tiết về tài liệu, bao gồm tên, tác giả, mô tả.
* Hỗ trợ nhiều định dạng tài liệu: Có khả năng tìm kiếm tài liệu ở nhiều định dạng khác nhau như PDF, DOCX, hình ảnh

### Chức năng phụ:

* Hỗ trợ tìm kiếm thông tin từ tài liệu được thêm bởi người dùng: Chatbot không chỉ trả lời câu hỏi của người dùng dựa theo những tài liệu đã có sẵn trên hệ thống mà còn có thể tìm kiếm dựa trên tài liệu mà người dùng tự thêm vào.
* Tạo phiên trò chuyện mới: cho phép người dùng bắt đầu 1 phiên trò chuyện mới với chatbot và có đặt tên cho mỗi phiên trò chuyện.
* Lưu trữ phiên trò chuyện: Lưu lại nội dung của phiên trò truyện trước khi tạo phiên mới và hiển thị danh sách các phiên ở sidebar.
* Chuyển đổi giữa các phiên trò chuyện:P người dùng có thể chuyển đổi qua lại giữa các phiên trò chuyện với chatbot.

## Đặc tả hệ thống:

### Đặc tả chức năng chính:

* Tương tác thông minh:
* Chatbot hiểu và phản hồi các câu hỏi của người dùng liên quan đến tài liệu học tập.
* Cung cấp câu trả lời dựa trên dữ liệu trong hệ thống, không lấy dữ liệu bên ngoài tài liệu được cho.
* Tìm kiếm tài liệu:
* Tìm kiếm tài liệu theo từ khóa, chủ đề, hoặc danh mục.
* Gợi ý tài liệu dựa trên lịch sử tìm kiếm và sở thích cá nhân.
* Cá nhân hóa:
* Lưu trữ lịch sử tìm kiếm và câu hỏi của người dùng để cải thiện trải nghiệm.

### Đặc tả chức năng phụ:

* Hỗ trợ tìm kiếm thông tin từ tài liệu được thêm bởi người dùng:
* Chức năng này cho phép người dùng thêm các tài liệu cá nhân vào hệ thống. Chatbot sẽ tích hợp các tài liệu này vào cơ sở dữ liệu, xử lý chúng, và trả lời các câu hỏi dựa trên nội dung từ cả tài liệu gốc lẫn tài liệu người dùng thêm vào.
* Tạo phiên trò chuyện mới:
* Cho phép người dùng tạo một phiên trò chuyện mới bất kỳ lúc nào.
* Tự động lưu trữ nội dung của phiên trò chuyện trước đó.
* Đặt tên cho phiên trò chuyện dựa trên câu hỏi đầu tiên hoặc do người dùng nhập.
* Lưu trữ phiên trò chuyện:
* Hệ thống tự động lưu lại nội dung của phiên trò chuyện hiện tại trước khi người dùng bắt đầu phiên mới.
* Các phiên trò chuyện được hiển thị dưới dạng danh sách trong sidebar để tiện quản lý.
* Chuyển đổi giữa các phiên trò chuyện:
* Người dùng có thể chuyển đổi giữa các phiên trò chuyện đã lưu để tiếp tục tương tác hoặc xem lại nội dung.

# THIẾT KẾ HỆ THỐNG

## Sơ đồ phân ra chức năng:

A diagram of a company

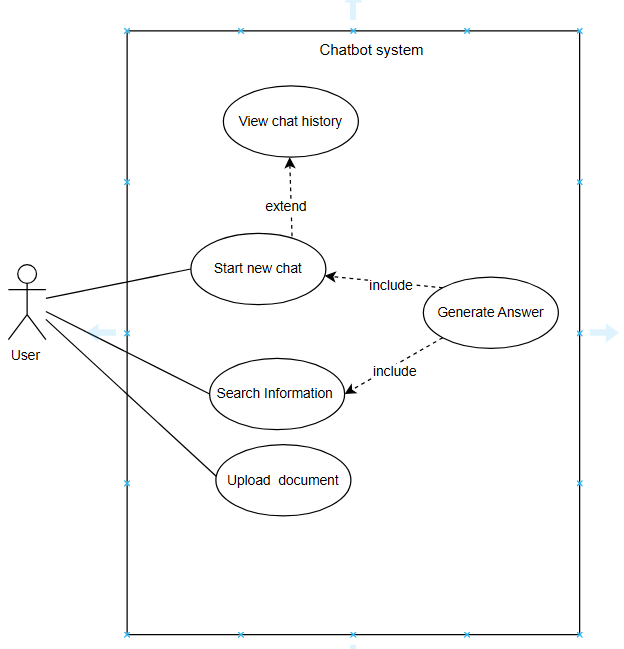
Description automatically generated

Hình 4.1- Sơ đồ phân rã chức năng

Mô tả chỉ tiết:

* Tương tác với người dùng:
* Nhận đầu vào từ người dùng: Nhận câu hỏi từ người dùng
* Hiển thị phản hồi từ hệ thống: Hệ thống đưa ra câu trả lời tương ứng với câu hỏi mà người dùng gửi
* Quản lý phiên trò chuyện: Bắt đầu, lưu, chuyển đổi giữa các phiên trò chuyện.
* Xử lý dữ liệu:
* Tiền xử lý dữ liệu tải lên: Trích xuất nội dung từ tài liệu. Phân loại tài liệu nếu tài liệu không phải định dạng file không lấy được dữ liệu văn bản thì phải chạy qua công cụ OCR để lấy dữ liệu văn bản.
* Tạo vector Embedding: Chuyển đổi nội dung văn bản thành vector số phục vụ tìm kiếm.
* Tích Hợp Dữ Liệu Người Dùng: Kết hợp dữ liệu người dùng tải lên với dữ liệu hệ thống.
* Truy Xuất Thông Tin:
* Xử Lý Truy Vấn: Phân tích và hiểu câu hỏi của người dùng bằng NLP.
* Phân Tích Truy Vấn: Tách câu hỏi thành các thành phần.
* Nhận Diện Ý Định: Xác định mục đích của câu hỏi.
* Trích Xuất Từ Khóa: Lấy các từ khóa quan trọng để tìm kiếm.
* Xếp Hạng Kết Quả: Sắp xếp tài liệu dựa trên độ liên quan.
* Tìm Kiếm Nguồn Dữ Liệu: Truy vấn trên dữ liệu có sẵn và tài liệu của người dùng.
* Tạo Phản Hồi: Kết hợp dữ liệu truy vấn với GPT-4o để trả lời.
* Quản Lý Hệ Thống
* Quản Lý Cơ Sở Dữ Liệu: Tổ chức, lưu trữ tài liệu, vector embeddings, và metadata.
* Theo Dõi Hiệu Suất Hệ Thống: Ghi nhận hiệu suất, log lỗi, và tương tác để cải thiện.

## Sơ đồ usecase:



Hình 4.2 - Sơ đồ use case tổng quát

### Usecase Search Information:

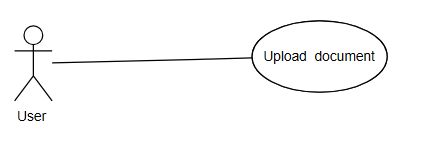
A black line with text on it

Description automatically generated

Hình 4.3 - Usecase Search Information

* Tác nhân: User
* Mô tả: Usecase cho phép user tìm kiếm thông tin từ tài liệu có sẵn trong hệ thống chatbot hoặc từ file user gửi
* Điều kiện:
* User nhập nội dung tìm kiếm (từ khóa hoặc câu hỏi).
* Tài liệu liên quan đã có sẵn trong cơ sở dữ liệu hệ thống hoặc đã được tải lên bởi User.
* Sự kiện và kết quả:
  + Nếu tìm kiếm thành công thì hệ thống truy vấn dữ liệu dựa trên nội dung nhập vào và có kết quả trả về
  + Nếu tìm kiếm thất bại, hệ thống thông báo "Không tìm thấy dữ liệu phù hợp."

### Usecase Upload Document:



Hình 4.4 - Usecase Upload Document

* Tác nhân: User
* Mô tả: Use case cho phép User tải lên các tài liệu (PDF, DOCX, hình ảnh) để hệ thống phân tích và tích hợp nội dung vào cơ sở dữ liệu.
* Điều kiện: User chọn tệp tin cần tải lên từ thiết bị cá nhân.
* Sự kiện và kết quả:
  + Nếu tìm kiếm thành công thì hệ thống hiển thị: “Tải lên thanh công” rồi tiến hành xử lý tài liệu (trích xuất văn bản, tạo embeddings) và được thêm vào cơ sở dữ liệu để hỗ trợ các truy vấn trong tương lai
  + Nếu tìm kiếm thất bại, hệ thống hiển thị thông báo lỗi (ví dụ: "Định dạng không hợp lệ" hoặc "Kích thước tệp quá lớn")

### Use Case: Start New Chat

A black line with a white background

Description automatically generated

Hình 4.5 - Use case Start New Chat

* Tác nhân: User
* Mô tả: Use case cho phép User bắt đầu một phiên trò chuyện mới với chatbot.
* Điều kiện: User nhấn vào nút "Tạo phiên trò chuyện mới" trên giao diện.
* Sự kiện và kết quả:
* Nếu khởi tạo thành công, hệ thống lưu lại nội dung của phiên trò chuyện cũ (nếu có), hiển thị giao diện trống để bắt đầu trò chuyện mới. User có thể nhập câu hỏi mới vào phiên trò chuyện.
* Nếu khởi tạo thất bại: Hệ thống hiển thị thông báo lỗi

### Use Case: View Chat History

A diagram of a chat

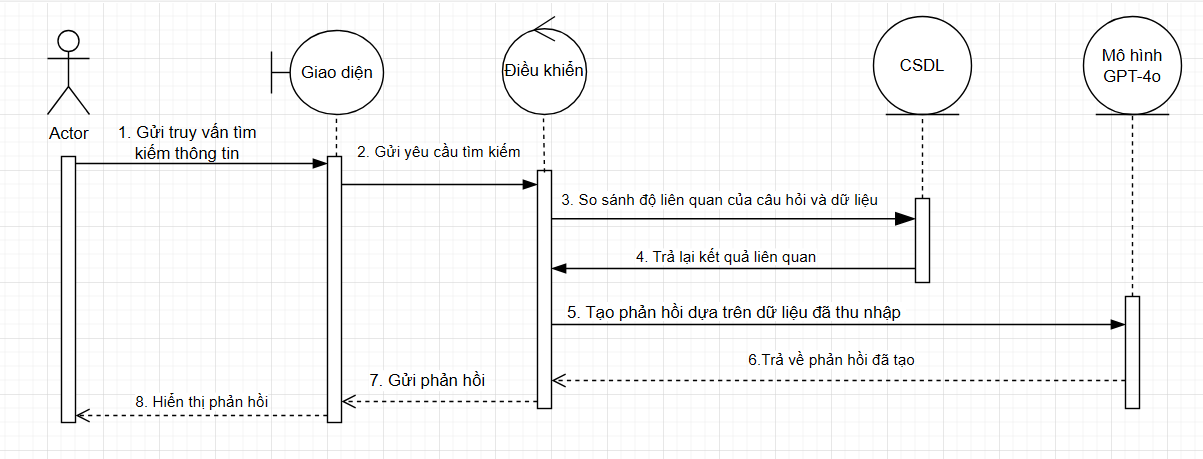
Description automatically generated

Hình 4.6 - Use case View chat history

* Tác nhân: User
* Mô tả: Use case cho phép User xem lại các phiên trò chuyện cũ đã lưu trong hệ thống.
* Điều kiện: User chọn một phiên trò chuyện từ danh sách trên giao diện sidebar.
* Sự kiện và kết quả:
* Nếu xem thành công: Hệ thống tải và hiển thị nội dung của phiên trò chuyện được chọn. User có thể tiếp tục trò chuyện hoặc xem lại toàn bộ lịch sử.
* Nếu xem thất bại: Không mở được phiên trò chuyện cũ.

## Sơ đồ tuần tự:

### Sơ đồ tuần tự Search Information



Hình 4.7 - Sơ đồ tuần tự Search Information

### Sơ đồ tuần tự Upload file

A diagram of a diagram

Description automatically generated

Hình 4.8 - Sơ đồ tuần tự Upload File

### Sơ đồ tuần tự Start New Chat

A diagram of a diagram

Description automatically generated

Hình 4.9 - Sơ đồ tuần tự Start New Chat

### Sơ đồ tuần tự View History chat

A diagram of a diagram

Description automatically generated

Hình 4.10 - Sơ đồ tuần tự View Chat Hítory

## Thiết kế cơ sở dữ liệu.

A screenshot of a computer screen

Description automatically generated

Hình 4.13. Sơ đồ cơ sở dữ liệu

### Mô tả các bảng

* Bảng documents:

Bảng 4.1 - Bảng Documents

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| document\_id | INT (PK) | Mã định danh duy nhất cho tài liệu. |
| title | NVARCHAR(100) | Tiêu đề tài liệu. |
| description | NVARCHAR(MAX) | Mô tả nội dung tài liệu. |
| file\_type | NVARCHAR(10) | Loại định dạng (PDF, DOCX, v.v.). |
| file\_path | NVARCHAR(MAX) | Đường dẫn lưu trữ tài liệu. |
| uploaded\_at | DATETIME | Thời gian tài liệu được tải lên. |

* Bảng embeddings:

Bảng 4.2 - Bảng embeddings

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| embedding\_id | INT (PK) | Mã định danh duy nhất cho vector. |
| document\_id | INT (FK) | Tham chiếu đến documents.document\_id. |
| embedding\_vector | NVARCHAR(MAX) | Dữ liệu vector embedding. |
| text\_segment | NVARCHAR(MAX) | Đoạn văn bản liên quan. |

* Bảng search\_history:

Bảng 4.3 - Bảng search\_history

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| history\_id | INT (PK) | Mã định danh duy nhất. |
| query | NVARCHAR(MAX) | Nội dung truy vấn. |
| searched\_at | DATETIME | Thời gian tìm kiếm. |

* Bảng conversations:

Bảng 4.4 - Bảng conversations

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| conversation\_id | INT (PK) | Mã định danh duy nhất. |
| started\_at | DATETIME | Thời gian bắt đầu hội thoại. |
| ended\_at | DATETIME | Thời gian kết thúc hội thoại. |

* Bảng messages:

Bảng 4.5 - Bảng messages

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| message\_id | INT (PK) | Mã định danh duy nhất cho tin nhắn. |
| conversation\_id | INT (FK) | Tham chiếu đến conversations.conversation\_id. |
| sender | NVARCHAR(10) | Người gửi (user/chatbot). |
| message\_content | NVARCHAR(MAX) | Nội dung tin nhắn. |
| sent\_at | DATETIME | Thời gian gửi tin nhắn. |

* Bảng recommendations:

Bảng 4.6 - Bảng recommendations

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| recommendation\_id | INT (PK) | Mã định danh duy nhất. |
| document\_id | INT (FK) | Tham chiếu đến documents.document\_id. |
| recommended\_at | DATETIME | Thời gian tài liệu được đề xuất. |

* Bảng upload\_files:

Bảng 4.7 - Bảng uploaded\_files

|  |  |  |
| --- | --- | --- |
| **Tên** | **Kiểu dữ liệu** | **Mô tả** |
| file\_id | INT (PK) | Mã định danh duy nhất cho tệp. |
| file\_name | NVARCHAR(100) | Tên tệp. |
| file\_type | NVARCHAR(10) | Loại định dạng của tệp. |
| file\_path | NVARCHAR(MAX) | Đường dẫn lưu trữ tệp. |
| upload\_date | DATETIME | Thời gian tải lên tệp. |

### Mô tả quan hệ:

* **documents – embeddings**
* Quan hệ: Một tài liệu (documents) có thể có nhiều vector embedding (embeddings).
* Chi tiết: Bảng embeddings chứa cột document\_id là khóa ngoại liên kết đến bảng documents. Mỗi vector embedding đại diện cho một phần nội dung của tài liệu, được sử dụng để hỗ trợ tìm kiếm thông minh.
* **conversations – messages**
* Quan hệ: Một cuộc hội thoại (conversations) có thể chứa nhiều tin nhắn (messages).
* Chi tiết: Trong bảng messages, trường conversation\_id là khóa ngoại liên kết đến bảng conversations. Mỗi tin nhắn thuộc về một cuộc hội thoại cụ thể, lưu trữ nội dung trao đổi giữa người dùng và chatbot.
* **documents – recommendations**
* Quan hệ: Một tài liệu (documents) có thể được gợi ý nhiều lần trong bảng recommendations.
* Chi tiết: Trong bảng recommendations, trường document\_id là khóa ngoại liên kết đến bảng documents. Mỗi bản ghi trong recommendations ghi nhận thời gian và tài liệu được đề xuất cho người dùng.
* **uploaded\_files**
* Quan hệ: Độc lập, không liên kết trực tiếp với các bảng khác.
* Chi tiết: Bảng uploaded\_files chứa thông tin về các tệp mà người dùng tải lên hệ thống, bao gồm tên tệp, loại định dạng, và đường dẫn lưu trữ. Tệp này có thể được xử lý để trích xuất nội dung và tìm kiếm.

## Tích hợp GPT-4o và LlamaIndex:

### Quy trình tích hợp:

* **Chuẩn bị dữ liệu**
* Tài liệu được đưa lên hệ thống và lưu và bảng documents.
* Dùng mô hình nhúng ngôn ngữ của OpenAI Embedding để xử lý các tài liệu được đưa vào và tạo thành vector embedding
* Vector embeddings được lưu giữ trong bảng embeddings.
* **Tạo chỉ mục với LlamaIndex:**
* Dữ liệu được lưu trong các bảng documents và embeddinsg được nhập vào Llamaindex.
* LlamaIndex tạo chỉ mục thông minh, cho phép truy vấn dữ liệu theo cách ngữ nghĩa thay vì chỉ dựa vào từ khóa.
* Chỉ mục được lưu trữ và tích hợp vào hệ thống backend để sử dụng trong các truy vấn.
* **Tích hợp GPT-4o:**
* Hệ thống backend (FastAPI) xử lý các truy vấn từ người dùng và truyền chúng đến LlamaIndex.
* Kết quả tìm kiếm từ LlamaIndex (các đoạn văn bản hoặc tài liệu liên quan) được chuyển đến GPT-4o để tạo ra câu trả lời tự nhiên và mạch lạc.
* GPT-4o sử dụng ngữ cảnh từ truy vấn và dữ liệu tìm kiếm để đảm bảo câu trả lời phù hợp và chính xác.

## Thiết kế giao diện chatbot

A screenshot of a chat

Description automatically generated

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận:

Trong quá trình thực hiện đồ án “xây dựng hệ thống chatbot thông minh hỗ trợ tìm kiếm tài liệu học tập”, em đạt được những kết quả sau:

* Phân tích và thiết kế hệ thống:
* Đã tiến hành phân tích yêu cầu người dùng và xác định các chức năng cần thiết cho hệ thống, bao gồm tìm kiếm tài liệu, gợi ý tài liệu, và hỗ trợ tài liệu người dùng tải lên.
* Thiết kế hệ thống hoàn chỉnh với sơ đồ phân rã chức năng, sơ đồ use case, sơ đồ tuần tự, và thiết kế cơ sở dữ liệu đảm bảo tính logic và tối ưu hóa luồng hoạt động.
* Phát triển hệ thống:
* Ứng dụng các công nghệ hiện đại như Python, FastAPI, GPT-4o, và LlamaIndex để xây dựng hệ thống backend hiệu quả và dễ mở rộng.
* Tích hợp các tính năng thông minh như xử lý ngôn ngữ tự nhiên (NLP), gợi ý tài liệu dựa trên lịch sử tìm kiếm, và tìm kiếm dữ liệu đa định dạng (PDF, DOCX, hình ảnh).
* Kiểm thử và triển khai:
* Thực hiện kiểm thử chức năng và kiểm thử hiệu năng, đảm bảo hệ thống hoạt động ổn định và cung cấp kết quả chính xác.
* Triển khai hệ thống trên nền tảng đám mây, đảm bảo truy cập ổn định từ nhiều thiết bị và khả năng mở rộng dễ dàng trong tương lai.

## Hướng phát triển:

* Tối ưu hóa hiệu suất:
* Cải thiện tốc độ xử lý tìm kiếm tài liệu và thời gian phản hồi của hệ thống để nâng cao trải nghiệm người dùng.
* Tối ưu hóa cơ sở dữ liệu vector và mã nguồn nhằm tăng khả năng xử lý đồng thời và giảm thời gian truy vấn.
* Cải thiện giao diện người dùng:
* Thiết kế lại giao diện chatbot trực quan hơn, thân thiện với mọi đối tượng người dùng.
* Tích hợp chatbot hỗ trợ trực tuyến và hướng dẫn sử dụng chi tiết để giúp người dùng khai thác tối đa tính năng của hệ thống.
* Phát triển ứng dụng di động:
* Xây dựng ứng dụng di động cho hệ thống chatbot, giúp người dùng truy cập nhanh chóng và tiện lợi mọi lúc, mọi nơi.
* Đồng bộ hóa dữ liệu giữa phiên bản web và ứng dụng di động để đảm bảo trải nghiệm liền mạch.
* Mở rộng thị trường và tích hợp đa ngôn ngữ:
* Phát triển hệ thống chatbot hỗ trợ nhiều ngôn ngữ, mở rộng phục vụ người dùng quốc tế.
* Nghiên cứu và tích hợp các tài liệu từ các hệ thống giáo dục khác nhau, phục vụ nhu cầu học tập đa dạng trên toàn cầu.
* Mở rộng tính năng:
* Cho phép hệ thống tự động phân loại tài liệu mới theo các chủ đề, từ khóa, và mức độ liên quan.
* Tích hợp tính năng lọc tài liệu cũ hoặc không phù hợp để duy trì cơ sở dữ liệu gọn gàng và hữu ích.
* Tích hợp tính năng kiểm tra ngữ pháp, cấu trúc câu, và phong cách viết để giúp người dùng cải thiện kỹ năng viết.
* Cung cấp gợi ý chỉnh sửa bài luận hoặc báo cáo dựa trên nội dung tài liệu đã nhập.

# DANH MỤC TÀI LIỆU THAM KHẢO

1. https://www.coursera.org/articles/what-is-a-chatbot
2. <https://www.ibm.com/think/topics/vector-embedding#:~:text=Vector%20embeddings%20are%20numerical%20representations,(ML)%20models%20can%20process>.
3. <https://www.ibm.com/think/topics/gpt-4o>
4. <https://www.datacamp.com/blog/vector-embedding?utm_source=google&utm_medium=paid_search&utm_campaignid=19589720824&utm_adgroupid=152984013054&utm_device=c&utm_keyword=&utm_matchtype=&utm_network=g&utm_adpostion=&utm_creative=724847714572&utm_targetid=dsa-2222697810678&utm_loc_interest_ms=&utm_loc_physical_ms=9198864&utm_content=DSA~blog~Artificial-Intelligence&utm_campaign=230119_1-sea~dsa~tofu_2-b2c_3-row-p2_4-prc_5-na_6-na_7-le_8-pdsh-go_9-nb-e_10-na_11-na-dec24&gad_source=1&gclid=EAIaIQobChMIlrGE7p7DigMVl8wWBR09pgW5EAAYASAAEgInk_D_BwE>
5. <https://aws.amazon.com/what-is/retrieval-augmented-generation/>
6. https://thuvienphapluat.vn/chinh-sach-phap-luat-moi/vn/ho-tro-phap-luat/tu-van-phap-luat/59386/chuong-trinh-giao-duc-pho-thong-2018-chuong-trinh-tong-the#

# PHỤ LỤC