NGÔ PHÚC LƯƠNG

CHUYÊN NGÀNH HỆ THỐNG THÔNG TIN

2025

BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

NGÔ PHÚC LƯƠNG

CẢI THIỆN MÔ HÌNH NGÔN NGỮ LỚN BẰNG RAG FRAMEWORK VÀ ỨNG DỤNG TRONG HỆ THỐNG HỎI ĐÁP TỰ ĐỘNG

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ NGÀNH HỆ THỐNG THÔNG TIN

Hà Nội – 2025

BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

NGÔ PHÚC LƯƠNG

CẢI THIỆN MÔ HÌNH NGÔN NGỮ LỚN BẰNG RAG FRAMEWORK VÀ ỨNG DỤNG TRONG HỆ THỐNG HỎI ĐÁP TỰ ĐỘNG

Ngành: Hệ thống thông tin

Mã số: 8480104

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ NGÀNH HỆ THỐNG THÔNG TIN

NGƯỜI HƯỚNG DẪN: **TS. NGUYỄN MẠNH CƯỜNG**

Hà Nội – 2025

LỜI CAM ĐOAN

Tôi xin cam đoan đề án tốt nghiệp thạc sĩ ngành Hệ thống thông tin với đề tài "Cải thiện mô hình ngôn ngữ lớn bằng RAG framework và ứng dụng trong hệ thống hỏi đáp tự động" là do tôi nghiên cứu, tổng hợp và thực hiện dưới sự hướng dẫn của TS. Nguyễn Mạnh Cường.

Toàn bộ nội dung của đề án, những điều được trình bày là của chính cá nhân tôi hoặc là được tham khảo, tổng hợp từ nhiều nguồn tài liệu khác nhau. Tất cả các tài liệu tham khảo, tổng hợp đều được trích xuất nguồn gốc rỏ ràng. Các số liệu, kết quả được nêu trong luận văn là trung thực và được nghiên cứu thực nghiệm.

Học viên thực hiện

Ngô Phúc Lương

LỜI CẢM ƠN

Lời đầu tiên em gửi lời cảm ơn đến thầy Nguyễn Mạnh Cường đã luôn tận tình hướng dẫn, truyền đạt những kiến thức chuyên môn cũng như kinh nghiệm liên quan và giúp em thực hiện được tiếp thu với những kiến thức mới, chuyên sâu trong quá trình nghiên cứu và thực hiện đề án tốt nghiệp.

Đồng thời, em cũng xin chân thành cảm ơn Thầy Cô của Khoa Công nghệ thông tin và Khoa đào tạo sau Đại học tại Trường Đại học Công nghiệp Hà Nội vì sự hướng dẫn và giáo dục tận tâm của Thầy Cô. Nhờ đó mà em đã được trang bị kiến thức chuyên môn và tinh thần học tập, giúp em hoàn thành đồ án tốt nghiệp này.

Mặc dù đã cố gắng trong quá trình học tập, cũng như trong quá trình thực hiện đề án tốt nghiệp nhưng không thể tránh khỏi những thiếu sót, em rất mong được sự góp ý quý báu của tất cả các thầy cô giáo cũng như tất cả các anh chị để kết quả của em được hoàn thiện hơn.

Em xin trân trọng cảm ơn!

Học viên

Ngô Phúc Lương

MỤC LỤC

LỜI NÓI ĐẦU

Trên hành trình không ngừng phát triển của công nghệ, con người ngày càng nhận được một sự hỗ trợ to lớn, đặc biệt là trong việc hỏi đáp và tìm kiếm thông tin thông qua việc sử dụng chatbot kết hợp với mô hình ngôn ngữ lớn (Large Language Model - LLM). Điều này không chỉ đánh dấu một bước tiến quan trọng trong việc tạo ra trải nghiệm người dùng thông minh hơn mà còn mở ra cánh cửa cho một thế giới thông tin dễ tiếp cận hơn và tương tác thông minh hơn.

Sự hỗ trợ của công nghệ thông qua chatbot sử dụng LLM không chỉ giúp tiết kiệm thời gian mà còn tạo ra một môi trường tương tác thông minh và hiệu quả. Người dùng có thể truy cập thông tin một cách dễ dàng và nhanh chóng chỉ bằng cách đặt câu hỏi, mà không cần phải tìm kiếm qua nhiều trang web hoặc nguồn thông tin khác nhau.

Mặc dù mô hình ngôn ngữ lớn đã mang lại những tiến bộ đáng kinh ngạc trong việc xử lý ngôn ngữ tự nhiên, nhưng vẫn tồn tại những hạn chế đáng chú ý. Để khắc phục những hạn chế này và nâng cao khả năng hiểu biết và tương tác với người dùng, Retriever-Augmented Generation - RAG đã ra đời với sứ mệnh cải thiện và mở rộng khả năng của mô hình ngôn ngữ.

Một trong những hạn chế chính của LLM là khả năng hiểu ngữ cảnh và trích xuất thông tin từ nguồn dữ liệu ngoại giao. RAG giải quyết vấn đề này bằng cách kết hợp hai phần chính: Một bộ trích xuất thông tin (Retriever) và một bộ sinh thông tin (Generator). Bộ trích xuất thông tin giúp RAG lấy thông tin từ nguồn dữ liệu lớn và đa dạng, trong khi bộ sinh thông tin tạo ra câu trả lời dựa trên thông tin này.

RAG cung cấp khả năng tìm kiếm thông tin chính xác và đáng tin cậy hơn, từ đó cải thiện khả năng đưa ra câu trả lời chính xác và phù hợp hơn với ngữ cảnh.

Điều này giúp tăng cường khả năng tương tác tự nhiên và hiệu quả của chatbot, đồng thời giảm thiểu sự hiểu lầm và thông tin không chính xác.

Ngoài ra, RAG cũng mở ra cánh cửa cho việc tận dụng tri thức từ đồ thị tri thức và nguồn dữ liệu phức tạp, từ đó cung cấp câu trả lời phong phú và đa chiều hơn. Khả năng này giúp chatbot không chỉ đơn giản là cung cấp thông tin mà còn tạo ra một trải nghiệm tương tác sâu sắc và thú vị cho người dùng.

Đề tài "Cải thiện mô hình ngôn ngữ lớn bằng RAG framework và ứng dụng trong hệ thống hỏi đáp tự động" được thực hiện với mục tiêu chính là nâng cao khả năng hiểu biết, tương tác và đáp ứng nhu cầu thông tin của hệ thống hỏi đáp tự động thông qua mô hình ngôn ngữ lớn kết hợp với RAG framework.

Nội dung đề án sẽ bao gồm 3 chương chính, cụ thể như sau:

- Chương 1: Cơ sở lý thuyết – Chương này sẽ tập trung vào việc đề cập đến cơ sở lý thuyết về trí tuệ nhân tạo, xử lý ngôn ngữ tự nhiên và mô hình ngôn ngữ lớn để xây dựng nền tảng kiến thức cho các phần tiếp theo của đề tài.
- Chương 2: Cải thiện hệ thống hỏi đáp tự động dựa trên RAG Framework - Chương này sẽ trình bày chi tiết về cách áp dụng RAG framework để cải thiện hệ thống hỏi đáp tự động. Chúng ta sẽ xem xét cách RAG framework giúp nâng cao khả năng hiểu biết và tương tác của hệ thống, tạo ra trải nghiệm người dùng tốt hơn.
- Chương 3: Xây dựng hệ thống hỏi đáp tự động bằng mô hình ngôn ngữ lớn và rag framework - Trong chương này sẽ khám phá quá trình xây dựng hệ thống hỏi đáp tự động bằng cách kết hợp mô hình ngôn ngữ lớn và RAG framework và tìm hiểu về việc tích hợp hai công nghệ này để tạo ra một hệ thống hiệu quả và đáp ứng nhu cầu của người dùng một cách toàn diện.

DANH MỤC CÁC KÝ HIỆU VÀ TỪ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
AI	Artificial Intelligence	Trí tuệ nhân tạo
NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
LLM	Large Language Model	Mô hình ngôn ngữ lớn
GPT	Generative Pre-trained Transformer	Bộ biến đổi sinh trước được huấn luyện
RNN	Recurrent Neural Network	Mạng nơ-ron hồi tiếp
RAG	Retrieval Augmented Generation	Hệ tăng cường truy xuất
	Chatbot	Hệ thống hỏi đáp tự động
	File	Tài liệu
IR	Information Retrieval	Truy xuất thông tin
NLG	Natural Language Generation	Tạo ngôn ngữ tự nhiên
QAS	Question Answering Systems	Hệ thống hỏi đáp
ML	Machine Learning	Học máy
DL	Deep Learning	Học sâu
ANI	Artificial Narrow Intelligence	Trí tuệ nhân tạo hẹp
AGI	Artificial General Intelligence	Trí tuệ nhân tạo tổng hợp
SGI	Superintelligent Artificial Intelligence	Siêu trí tuệ nhân tạo

DANH MỤC CÁC HÌNH ẢNH

DANH MỤC CÁC BẢNG BIỂU

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1. TỔNG QUAN VỀ TRÍ TUỆ NHÂN TẠO

1.1.1. Giới thiệu về Trí tuệ nhân tạo

Trí tuệ nhân tạo (Artificial Intelligence - AI) đã trở thành một lĩnh vực không thể phủ nhận sức ảnh hưởng của nó đối với cuộc sống con người. AI là một lĩnh vực khoa học máy tính nhằm tạo ra các hệ thống máy tính có khả năng học và thực hiện các tác vụ mà trước đây chỉ có thể được thực hiện bởi con người. AI bao gồm việc phát triển các thuật toán, phần mềm và hệ thống máy tính có thể học hỏi, suy luận, giải quyết vấn đề và đưa ra quyết định một cách tự động. Hiểu một cách đơn giản, AI chính là nỗ lực của con người nhằm tái tạo trí thông minh của mình trong máy móc, cho phép chúng “suy nghĩ” và “hành động” như con người.

AI có thể được truy nguyên từ những năm 1950, khi các nhà khoa học máy tính bắt đầu khám phá khả năng tạo ra máy móc "thông minh". Tuy nhiên, chỉ trong những thập kỷ gần đây, với sự phát triển nhanh chóng của công nghệ máy tính và dữ liệu lớn, AI mới thực sự bùng nổ và trở thành một trong những lĩnh vực công nghệ quan trọng nhất. Trong bối cảnh mà công nghệ ngày càng tiến bộ, AI đã và đang thay đổi hoàn toàn cách chúng ta làm việc, giải trí, y tế, giao thông và nhiều lĩnh vực khác. Từ việc dự đoán thời tiết, gợi ý sản phẩm mua hàng, xử lý ngôn ngữ tự nhiên đến tự động hóa quy trình sản xuất, AI đã mở ra nhiều cơ hội mới và nâng cao hiệu quả công việc.

Có nhiều phương pháp và kỹ thuật khác nhau trong AI, bao gồm:

- Học máy (Machine Learning): Đây là một nhánh của AI tập trung vào việc phát triển các thuật toán cho phép máy tính học từ dữ liệu và cải thiện hiệu suất của chúng theo thời gian mà không cần được lập trình cụ thể.
- Học sâu (Deep Learning): Một phân nhánh của học máy sử dụng các mạng neural nhân tạo có nhiều lớp để xử lý và học từ dữ liệu phức tạp.

- Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP): Tập trung vào việc phát triển hệ thống có khả năng hiểu, diễn giải và tạo ra ngôn ngữ tự nhiên của con người.
- Thị giác máy tính (Computer Vision): Nghiên cứu về cách máy tính có thể hiểu và xử lý thông tin từ hình ảnh và video.
- Hệ thống chuyên gia (Expert Systems): Các hệ thống AI được thiết kế để mô phỏng quá trình đưa ra quyết định của các chuyên gia trong một lĩnh vực cụ thể.

Tuy nhiên, sức mạnh của AI cũng đi kèm với những thách thức đáng kể. Việc sử dụng AI đặt ra câu hỏi về đạo đức và trách nhiệm, đặc biệt là trong lĩnh vực quyền riêng tư và an ninh thông tin. Khả năng của AI trong việc thu thập, xử lý và phân tích dữ liệu có thể đặt ra nguy cơ cho sự riêng tư và an toàn của cá nhận:

- Đạo đức và quyền riêng tư: Khi AI ngày càng được tích hợp vào cuộc sống hàng ngày, nó làm dấy lên những lo ngại về quyền riêng tư và sử dụng dữ liệu cá nhân. Ví dụ, việc sử dụng AI trong hệ thống giám sát có thể dẫn đến vi phạm quyền riêng tư của công dân.
- Thiên kiến và công bằng: Các hệ thống AI có thể kế thừa và khuếch đại các thiên kiến hiện có trong dữ liệu huấn luyện, dẫn đến kết quả không công bằng hoặc phân biệt đối xử.
- Tính minh bạch và giải thích được: Nhiều hệ thống AI, đặc biệt là các mô hình học sâu, hoạt động như "hộp đen", khó giải thích cách chúng đưa ra quyết định. Điều này có thể gây ra vấn đề trong các lĩnh vực như y tế hoặc tài chính, nơi cần sự minh bạch trong quá trình ra quyết định.
- An ninh và bảo mật: Các hệ thống AI có thể trở thành mục tiêu của các cuộc tấn công mạng, và việc bảo vệ chúng khỏi các mối đe dọa này là một thách thức lớn.

- Tác động đến việc làm: Sự phát triển của AI có thể dẫn đến việc tự động hóa nhiều công việc, gây ra lo ngại về thất nghiệp và cần phải đào tạo lại lực lượng lao động.

1.1.2. Lịch sử phát triển Trí tuệ nhân tạo

Công trình đầu tiên hiện được công nhận rộng rãi là AI được thực hiện bởi Warren McCulloch và Walter Pitts (1943). Họ đã dựa trên ba nguồn: kiến thức về sinh lý học cơ bản và chức năng của các tế bào thần kinh trong não; một phân tích chính thức về logic mệnh đề do Russell và Whitehead thực hiện; và lý thuyết tính toán của Turing. Họ đề xuất một mô hình tế bào thần kinh nhân tạo trong đó mỗi tế bào thần kinh được mô tả là "bật" hoặc "tắt", với công tắc chuyển sang "bật" xảy ra để đáp ứng với sự kích thích của một số lượng đủ các tế bào thần kinh lân cận. Trạng thái của một tế bào thần kinh được coi là "tương đương về mặt thực tế với một mệnh đề đề xuất kích thích thích hợp của nó". Ví dụ, họ đã chỉ ra rằng bất kỳ hàm tính toán nào cũng có thể được tính toán bởi một số mạng lưới các tế bào thần kinh được kết nối và tất cả các kết nối logic (và, hoặc, không, v.v.) có thể được triển khai bằng các cấu trúc mạng đơn giản.

Princeton là nơi có một nhân vật có ảnh hưởng khác trong lĩnh vực AI, John McCarthy. Sau khi nhận bằng Tiến sĩ tại đây vào năm 1951 và làm giảng viên trong hai năm, McCarthy chuyển đến Stanford rồi đến Cao đẳng Dartmouth, nơi chính thức khai sinh ra lĩnh vực này. McCarthy đã thuyết phục Minsky, Claude Shannon và Nathaniel Rochester giúp ông tập hợp các nhà nghiên cứu Hoa Kỳ quan tâm đến lý thuyết máy tự động, mạng nơ-ron và nghiên cứu về trí thông minh. Họ đã tổ chức một hội thảo kéo dài hai tháng tại Dartmouth vào mùa hè năm 1956. Đề xuất nêu rõ: Chúng tôi đề xuất rằng một nghiên cứu kéo dài 2 tháng, với 10 người về trí tuệ nhân tạo sẽ được thực hiện vào mùa hè năm 1956 tại Cao đẳng Dartmouth ở Hanover, New Hampshire. Nghiên cứu này sẽ được tiến hành dựa trên phỏng đoán rằng về nguyên tắc, mọi khía cạnh của việc học hoặc bất kỳ đặc điểm nào khác của trí thông minh đều có thể được mô tả chính xác đến mức

có thể tạo ra một cỗ máy để mô phỏng nó. Chúng tôi sẽ cố gắng tìm cách để máy móc sử dụng ngôn ngữ, hình thành các khái niệm và trừu tượng, giải quyết các loại vấn đề hiện chỉ dành cho con người và cải thiện bản thân. Chúng tôi nghĩ rằng có thể đạt được tiến bộ đáng kể trong một hoặc nhiều vấn đề này nếu một nhóm các nhà khoa học được lựa chọn cẩn thận cùng nhau làm việc trong một mùa hè.

Hệ thống chuyên gia thương mại thành công đầu tiên, R1, bắt đầu hoạt động tại Digital Equipment Corporation (McDermott, 1982). Chương trình này giúp định cấu hình các đơn đặt hàng cho các hệ thống máy tính mới; đến năm 1986, chương trình đã giúp công ty tiết kiệm được khoảng 40 triệu đô la mỗi năm. Đến năm 1988, nhóm AI của DEC đã triển khai 40 hệ thống chuyên gia và sẽ còn triển khai thêm nữa. DuPont đã sử dụng 100 hệ thống và phát triển 500 hệ thống, tiết kiệm được khoảng 10 triệu đô la mỗi năm. Hầu như mọi tập đoàn lớn của Hoa Kỳ đều có nhóm AI riêng và đang sử dụng hoặc nghiên cứu các hệ thống chuyên gia.[1]

Vào giữa những năm 1980, ít nhất bốn nhóm khác nhau đã phát minh lại thuật toán học lan truyền ngược được Bryson tìm ra lần đầu tiên vào năm 1969. Thuật toán này đã được áp dụng cho nhiều vấn đề học tập trong khoa học máy tính và tâm lý học, và việc phổ biến rộng rãi các kết quả trong bộ sưu tập Xử lý phân tán song song (Rumelhart và McClelland, 1986) đã gây ra sự quan tâm lớn.

Trong suốt 60 năm lịch sử của khoa học máy tính, trọng tâm là thuật toán như là chủ đề nghiên cứu chính. Nhưng một số công trình gần đây về AI cho thấy rằng đối với nhiều vấn đề, việc lo lắng về dữ liệu và ít kén chọn hơn về thuật toán áp dụng có ý nghĩa hơn.

1.1.3. Đặc điểm của Trí tuệ nhân tạo

- **Ưu điểm của Trí tuệ nhân tạo**

AI có thể thực hiện các tác vụ lặp lại một cách nhanh chóng và chính xác hơn con người, giúp tăng cường hiệu quả và năng suất trong nhiều lĩnh vực công việc. AI có khả năng xử lý một khối lượng lớn các công việc mang tính lặp lại,

chẳng hạn như nhập liệu, kiểm tra lỗi, hoặc xử lý giao dịch, với tốc độ nhanh hơn nhiều so với con người. Điều này không chỉ giảm thiểu thời gian hoàn thành công việc mà còn giúp doanh nghiệp tiết kiệm chi phí lao động. Máy móc không mắc các lỗi do mệt mỏi, thiếu tập trung hay cảm xúc như con người. Trong các lĩnh vực như sản xuất, tài chính hay y tế, AI đảm bảo tính chính xác cao trong từng thao tác, giảm thiểu rủi ro và sai sót. Nhờ đảm nhận các công việc nhàm chán và tốn thời gian, AI cho phép nhân viên tập trung vào các công việc đòi hỏi tư duy sáng tạo, ra quyết định chiến lược hoặc các nhiệm vụ mang tính giá trị gia tăng cao hơn.

AI có khả năng xử lý và phân tích lượng dữ liệu lớn một cách hiệu quả, giúp phát hiện ra các mẫu và thông tin quan trọng từ dữ liệu phức tạp. AI, với các thuật toán tiên tiến và khả năng tính toán mạnh mẽ, có thể phân tích hàng triệu điểm dữ liệu trong thời gian ngắn, điều mà con người hoặc các công cụ truyền thống không thể đạt được.

Hệ thống AI có khả năng học từ dữ liệu mới và cải thiện hiệu suất theo thời gian mà không cần sự can thiệp của con người. Các hệ thống AI liên tục cải thiện hiệu quả hoạt động thông qua cơ chế học tập dựa trên phản hồi (Feedback Learning). Khi càng xử lý nhiều dữ liệu, AI càng trở nên chính xác và hiệu quả hơn.

AI được áp dụng rộng rãi trong nhiều lĩnh vực như y tế, tài chính, sản xuất, marketing, giúp tạo ra những giải pháp sáng tạo và tiện ích cho xã hội. Ví dụ, trong lĩnh vực y tế, AI đã và đang tạo ra những bước tiến đột phá trong chăm sóc sức khỏe và điều trị bệnh. Bên cạnh đI giúp cải thiện hoạt động quản lý tài chính và tăng hiệu quả trong các giao dịch. AI còn tối ưu hóa quy trình sản xuất, giảm chi phí và nâng cao chất lượng sản phẩm.

AI có khả năng dự đoán xu hướng và cung cấp tư vấn thông minh dựa trên dữ liệu, giúp hỗ trợ quyết định và lập kế hoạch hiệu quả. Từ đó, AI cung cấp các giải pháp và đề xuất dựa trên phân tích dữ liệu, giúp đưa ra quyết định chiến lược

hơn. Khả năng dự đoán và tư vấn của AI không chỉ giúp tổ chức và cá nhân hiểu rõ hơn về tương lai mà còn cung cấp các giải pháp tối ưu để hành động. Từ việc lập kế hoạch chiến lược đến dự báo xu hướng và quản lý rủi ro, AI trở thành công cụ đắc lực giúp nâng cao hiệu quả trong nhiều lĩnh vực khác nhau.

- **Một số hạn chế của Trí tuệ nhân tạo**

Bên cạnh đó AI cũng có những hạn chế:

Trách nhiệm và đạo đức: Sự phát triển của AI đặt ra những thách thức về trách nhiệm và đạo đức trong việc sử dụng công nghệ này, đảm bảo rằng nó hướng tới lợi ích cộng đồng và không gây hậu quả tiêu cực cho xã hội. Khi AI đưa ra quyết định dẫn đến sai lầm hoặc hậu quả nghiêm trọng, câu hỏi về trách nhiệm pháp lý trở nên phức tạp. AI học từ dữ liệu, và nếu dữ liệu đầu vào chứa đựng thành kiến hoặc không đại diện đầy đủ, AI có thể đưa ra các quyết định bất công hoặc phân biệt đối xử. AI thu thập và phân tích lượng lớn dữ liệu cá nhân, dẫn đến nguy cơ xâm phạm quyền riêng tư và an ninh. AI có thể bị sử dụng cho các mục đích bất hợp pháp hoặc phi đạo đức. AI có thể dẫn đến mất việc làm do tự động hóa, làm gia tăng bất bình đẳng xã hội. Điều này đặt ra câu hỏi về đạo đức trong việc cân bằng giữa tiến bộ công nghệ và lợi ích của con người. Những thách thức về trách nhiệm và đạo đức của AI nhấn mạnh tầm quan trọng của việc phát triển các quy định, tiêu chuẩn, và khuôn khổ pháp lý rõ ràng. Đồng thời, cần có sự hợp tác giữa các tổ chức công nghệ, chính phủ, và cộng đồng để đảm bảo rằng AI được sử dụng một cách có đạo đức, hướng tới lợi ích chung, và tránh các hậu quả tiêu cực cho xã hội.

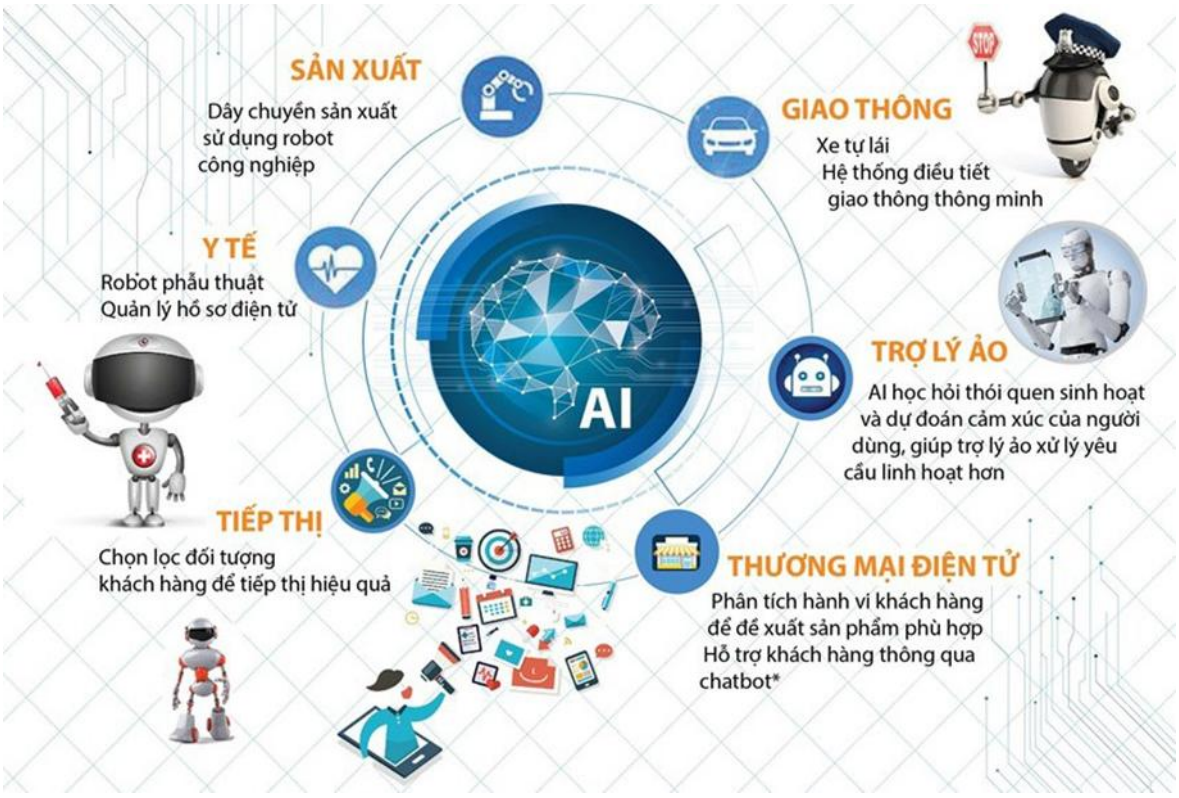
Thiếu sự hiểu biết và tinh tế như con người: Mặc dù AI có khả năng xử lý dữ liệu mạnh mẽ, nhưng nó vẫn thiếu sự hiểu biết và tinh tế như con người trong việc đánh giá ngữ cảnh phức tạp và quyết định đạo đức. Một chatbot AI có thể trả lời các câu hỏi thông thường, nhưng khi câu hỏi chứa yếu tố mỉa mai hoặc hàm ý, AI khó có thể hiểu đúng ý nghĩa. Trong văn hóa, AI khó nhận diện hoặc áp dụng các phong tục, tín ngưỡng một cách phù hợp vì thiếu sự cảm nhận ngầm định. AI

không có khả năng cảm nhận cảm xúc, do đó không thể tương tác với con người theo cách tinh tế và nhân văn.

Bảo mật thông tin: AI đặt ra những thách thức về bảo mật thông tin và quyền riêng tư khi xử lý dữ liệu cá nhân và nhạy cảm. AI xử lý khối lượng lớn dữ liệu, bao gồm cả dữ liệu nhạy cảm như thông tin tài chính, sức khỏe và hành vi cá nhân. Nếu không được bảo mật đúng cách, dữ liệu này có thể bị đánh cắp hoặc sử dụng sai mục đích. AI thường xuyên thu thập dữ liệu cá nhân mà người dùng không biết hoặc không đồng ý, dẫn đến vi phạm quyền riêng tư.

- **Các lĩnh vực ứng dụng trí tuệ nhân tạo**

AI trong thời gian gần đây đang được ứng dụng mạnh mẽ trong các lĩnh vực liên quan đến cuộc sống của con người. Có thể kể đến một số điển hình:



Hình 1.1: Các lĩnh vực ứng dụng AI (Nguồn: Infographics)

Y Tế và Dược Học: Trí tuệ nhân tạo (AI) đang có ứng dụng mạnh mẽ trong lĩnh vực y tế và dược học. Từ việc chẩn đoán bệnh dựa trên dữ liệu hồ sơ y khoa đến việc dự báo nguy cơ mắc bệnh, AI đang hỗ trợ các bác sĩ và nhà nghiên cứu

trong việc tìm ra các giải pháp hiệu quả và chuẩn xác. Hệ thống AI cũng được áp dụng trong nghiên cứu dược phẩm, giúp phát triển các loại thuốc mới và cải thiện quy trình sản xuất.

Tài Chính và Ngân Hàng: Trong lĩnh vực tài chính và ngân hàng, AI đóng vai trò quan trọng trong việc phân tích dữ liệu thị trường, dự báo xu hướng và hỗ trợ quyết định đầu tư. Hệ thống AI cũng được sử dụng để tạo ra chatbot hỗ trợ khách hàng, giúp nâng cao trải nghiệm người dùng và tối ưu hóa quy trình giao dịch.

Tự Động Hóa và Sản Xuất: Trong ngành sản xuất, AI được tích hợp vào robot để tăng cường hiệu suất và linh hoạt trong quá trình sản xuất. Ngoài ra, AI cũng hỗ trợ trong việc dự báo và quản lý chuỗi cung ứng, giúp tối ưu hóa hoạt động sản xuất và giảm thiểu lãng phí.

Giao Thông và Vận Tải: Trong lĩnh vực giao thông và vận tải, công nghệ AI đang giúp phát triển hệ thống xe tự hành và cải thiện quản lý giao thông. Từ việc dự đoán tình hình giao thông đến tối ưu hóa lộ trình, AI đang đóng vai trò quan trọng trong việc giảm ùn tắc và cải thiện an toàn giao thông.

Marketing và Quảng Cáo: Trong lĩnh vực marketing và quảng cáo, AI được sử dụng để tạo ra chiến lược tiếp thị cá nhân hóa và tăng cường khả năng tiếp cận đúng đối tượng. Công nghệ AI giúp các doanh nghiệp hiểu rõ hơn về khách hàng và tạo ra các chiến dịch quảng cáo hiệu quả hơn.

Nhìn chung, AI là một lĩnh vực đang phát triển nhanh chóng với tiềm năng to lớn để cải thiện cuộc sống của chúng ta theo nhiều cách. Tuy nhiên, nó cũng đặt ra những thách thức đáng kể cần phải giải quyết. Bằng cách tiếp cận phát triển và triển khai AI một cách có trách nhiệm và có suy xét, con người có thể tận dụng sức mạnh của công nghệ này để mang lại những giá trị tích cực cho xã hội.

- **Sự phát triển và tương lai của Trí tuệ nhân tạo**

Theo dòng chảy của cuộc cách mạng 4.0, trí tuệ nhân tạo ngày càng được phổ biến và ứng dụng rộng rãi trong mọi lĩnh vực của cuộc sống, mặc dù được John McCarthy – nhà khoa học máy tính người Mỹ đề cập lần đầu tiên vào những năm 1950 nhưng đến ngày nay thuật ngữ trí tuệ nhân tạo mới thực sự được biết đến rộng rãi và được các "ông lớn" của làng công nghệ chạy đua phát triển.

Những dự đoán về ứng dụng công nghệ AI trong nhiều lĩnh vực khác nhau, các nhà nghiên cứu, doanh nghiệp, khởi nghiệp và chính phủ có thể định hướng mục tiêu phát triển trong tương lai.

Công nghệ trí tuệ nhân tạo (AI) được xem như một công cụ then chốt giúp thúc đẩy quá trình cải cách hành chính trở nên hiệu quả và toàn diện hơn, từ đó giải quyết nhiều khó khăn và hạn chế đang tồn tại trong công tác quản lý, điều hành của các cơ quan nhà nước ở mọi cấp độ. Một trong những ứng dụng tiêu biểu của AI là việc triển khai các hệ thống chatbot và trợ lý ảo tại các trung tâm dịch vụ hành chính công. Những công nghệ này giúp người dân dễ dàng tiếp cận thông tin, nhận được phản hồi nhanh chóng và chính xác mà không phải mất thời gian chờ đợi hay xếp hàng đông đúc như truyền thống.

Ngoài ra, với nguồn dữ liệu lớn được lưu trữ trong các hệ thống của chính phủ, AI còn có khả năng xử lý, tổ chức và phân tích dữ liệu phức tạp từ nhiều nguồn khác nhau để trích xuất thông tin quan trọng, đồng thời cung cấp các bản tóm tắt súc tích giúp các nhà quản lý dễ dàng đưa ra quyết định chính xác và kịp thời. Tuy nhiên, việc áp dụng AI trong lĩnh vực này cũng đặt ra những thách thức không nhỏ, đặc biệt là về mặt an ninh mạng và bảo mật thông tin. Do đó, các hệ thống AI cần được thiết kế với các biện pháp bảo vệ chặt chẽ nhằm ngăn chặn các rủi ro về bảo mật, đảm bảo an toàn dữ liệu cá nhân cũng như thông tin quan trọng của nhà nước.

Nhận dạng khuôn mặt bằng trí tuệ nhân tạo là một ứng dụng phổ biến để xác minh đặc tính gương mặt. Máy tính có khả năng tự động xác định và nhận biết một người từ một bức ảnh số hoặc một khung hình trong video. Quá trình này

thuộc lĩnh vực thị giác máy tính, nơi mà máy tính có khả năng nhìn và phân tích hình ảnh hiệu quả hơn so với con người. Thay vì dựa vào ví dụ điểm nút trên khuôn mặt, công nghệ AI còn có thể sử dụng các phương pháp khác để nhận diện khuôn mặt.

Công nghệ trí tuệ nhân tạo đã đạt được nhiều thành công trong nhiều lĩnh vực khác nhau, mặc dù vẫn còn nhiều tiềm năng phát triển. Năm 2016, giá trị thị trường toàn cầu của AI là 4 tỷ USD, nhưng dự báo sẽ tăng lên 169 tỷ USD vào năm 2025 và 15.700 tỷ USD vào năm 2035. Với sự thay đổi liên tục trong công nghệ và ứng dụng trong cuộc sống hàng ngày, công nghệ trí tuệ nhân tạo đang trở thành trọng tâm của nhiều nhà nghiên cứu trong tương lai.

- **So sánh và phân tích giữa AI với Trí tuệ tự nhiên**

Trí tuệ nhân tạo (AI) là kết quả của quá trình nghiên cứu và phát triển các yếu tố thông minh, có khả năng phân tích môi trường và thực hiện hành động để đạt kết quả tối ưu. AI đòi hỏi sự kết hợp từ nhiều lĩnh vực như khoa học máy tính, tâm lý học, kinh tế và lý thuyết điều khiển. Các ứng dụng của AI ngày càng mở rộng, bao gồm việc điều khiển hệ thống, lập kế hoạch, khai thác dữ liệu, nhận dạng giọng nói, logictics và nhận diện khuôn mặt.

Năng lực trí tuệ của con người là kết quả của quá trình tích luỹ kinh nghiệm, sự linh hoạt trong việc thích nghi, khả năng xử lý ý tưởng trừu tượng và khả năng biến đổi môi trường thông qua việc học hỏi. Nó tạo ra một phạm vi thông tin đa dạng, từ quan sát trong du lịch đến dữ liệu trong các lĩnh vực khác nhau. Trí tuệ con người cũng cung cấp thông tin về các mối quan hệ và mạng lưới xã hội, đóng góp vào việc hình thành kiến thức và sự hiểu biết của chúng ta.

Tuy trí tuệ AI và trí tuệ con người đều là hai dạng trí tuệ thông minh nhưng chúng có nhiều điểm khác biệt như sau:

Bảng 1.1: Bảng so sánh trí tuệ nhân tạo với trí tuệ con người

Yếu Tố	**Trí tuệ nhân tạo**	**Trí tuệ con người**
Nguồn gốc hình thành	Được hình thành thông qua lập trình và học máy	Hình thành từ kinh nghiệm, cảm nhận và sự sáng tạo
Tốc độ xử lý công việc	Xử lý nhanh chóng và chính xác các tác vụ cụ thể	Kết hợp sự cảm nhận, sáng tạo và xử lý suy nghĩ trừu tượng
Khả năng ra quyết định	Dựa vào quy luật lập trình và dữ liệu đầu vào	Kết hợp logic, trực giác và tầm nhìn chiến lược
Mức độ chính xác	Chủ yếu đạt được thông qua tính toán máy tính	Kết hợp thông tin lý thuyết và sự hiểu biết sâu sắc
Sự sáng tạo	Tập trung vào tác vụ cụ thể và mô hình dữ liệu	Kết hợp sự sáng tạo, đạo đức và khả năng tạo ra ý tưởng mới
Khả năng tương tác xã hội	Thường dựa vào mã lực máy tính để tương tác	Có khả năng tương tác xã hội sâu sắc và tự nhiên
Khả năng thích ứng với môi trường	Không có tính ứng linh hoạt với môi trường đặc biệt, chỉ hoạt động	Có khả năng thích ứng linh hoạt và sáng tạo trong môi trường đa dạng

	dựa trên những thông tin lập trình sẵn	
Sự đa nhiệm	Thường cần điều chỉnh và lập trình lại để thích nghi với các tác vụ mới.	Có thể thực hiện nhiều nhiệm vụ cùng lúc.

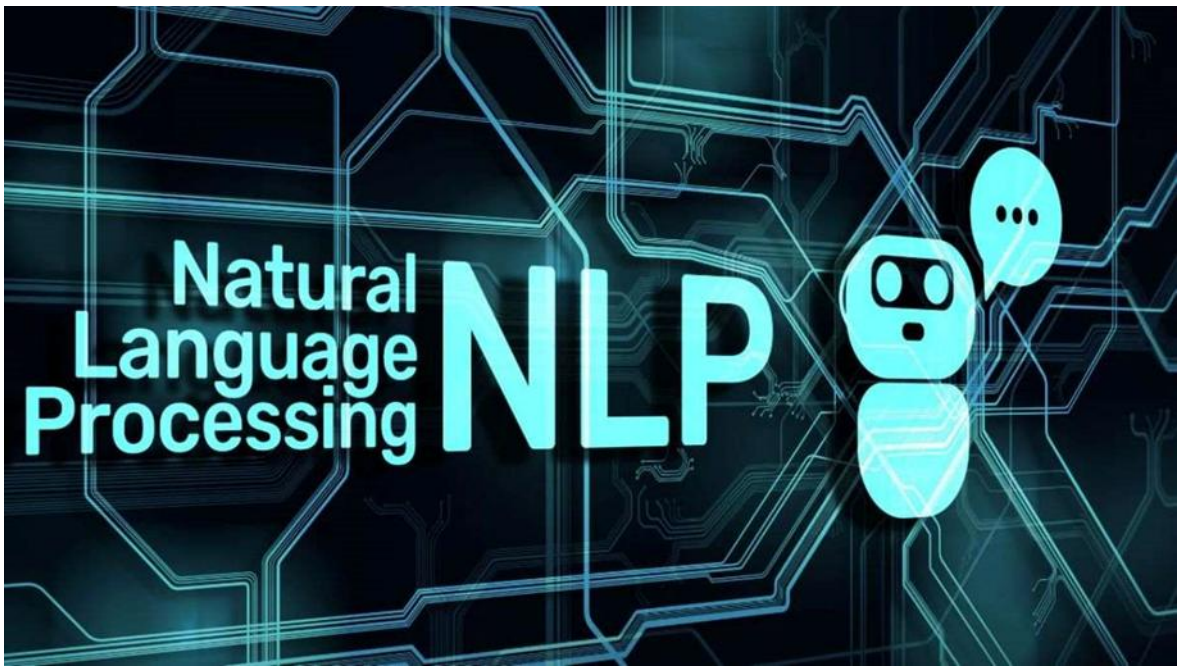
1.2. TỔNG QUAN VỀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN

1.2.1. Giới thiệu Ngôn ngữ tự nhiên

Trong thời đại công nghệ 4.0, Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) đang trở thành một trong những lĩnh vực nổi bật của Trí Tuệ Nhân Tạo, mở ra cánh cửa cho việc tương tác và hiểu biết giữa con người và máy tính thông qua ngôn ngữ tự nhiên.

Xử lý Ngôn ngữ Tự nhiên (NLP) là một lĩnh vực nghiên cứu nằm ở giao điểm của khoa học máy tính, trí tuệ nhân tạo và ngôn ngữ học. NLP tập trung vào việc phát triển các kỹ thuật và phương pháp để máy tính có thể hiểu, diễn giải và tạo ra ngôn ngữ tự nhiên của con người. Trong thời đại số hóa ngày nay, NLP đóng vai trò quan trọng trong việc cải thiện tương tác giữa con người và máy tính, đồng thời mở ra nhiều ứng dụng mới trong các lĩnh vực như dịch máy, trợ lý ảo, và phân tích dữ liệu văn bản.

NLP là lĩnh vực nghiên cứu và ứng dụng máy tính trong việc hiểu, phân tích và tạo ra ngôn ngữ tự nhiên một cách tự động. Với sự phát triển của học máy (ML) và học sâu (DL), NLP đã đạt được những tiến bộ đáng kinh ngạc trong việc xử lý ngôn ngữ tự nhiên, từ việc nhận diện giọng nói đến dịch máy và tổng hợp văn bản.



Hình 1.2: Xử lý Ngôn ngữ Tự nhiên (Nguồn: leilasafari)

1.2.2. Lịch sử phát triển của kỹ thuật Xử lý Ngôn ngữ tự nhiên

Lịch sử phát triển của NLP có thể được chia thành ba giai đoạn chính. Giai đoạn đầu, từ những năm 1950 đến 1980, tập trung vào các hệ thống dựa trên quy tắc và ngữ pháp. Các nhà nghiên cứu cố gắng mô hình hóa ngôn ngữ thông qua các quy tắc cứng nhắc, nhưng phương pháp này gặp nhiều hạn chế do sự phức tạp và đa dạng của ngôn ngữ tự nhiên. Giai đoạn thứ hai, từ những năm 1980 đến 2000, chứng kiến sự chuyển hướng sang các phương pháp học máy thống kê và mô hình xác suất. Cách tiếp cận này cho phép hệ thống NLP học từ dữ liệu thực tế, cải thiện đáng kể hiệu suất và khả năng thích ứng. Giai đoạn thứ ba, từ năm 2000 đến nay, đánh dấu sự bùng nổ của deep learning và neural networks trong NLP, với đột phá quan trọng nhất là sự ra đời của mô hình Transformer vào năm 2017.

Giai đoạn đầu, từ những năm 1950 đến 1980, thời điểm này, xử lý ngôn ngữ tự nhiên chủ yếu là một lĩnh vực nghiên cứu sơ khai, được thúc đẩy bởi sự ra đời của các máy tính hiện đại và mong muốn tự động hóa các tác vụ ngôn ngữ. Các nhà nghiên cứu quan tâm đến việc hiểu liệu máy móc có thể "hiểu" và "xử lý" ngôn ngữ con người hay không. Sử dụng các quy tắc cú pháp và từ vựng viết

tay để phân tích và xử lý ngôn ngữ. Công nghệ tính toán còn rất hạn chế, dẫn đến hiệu suất thấp và không thể xử lý ngữ cảnh phức tạp.

Giai đoạn thứ hai, từ những năm 1980 đến 2000, giai đoạn này đánh dấu một sự chuyển mình quan trọng trong lĩnh vực Xử lý Ngôn ngữ Tự nhiên (NLP), khi các phương pháp thống kê bắt đầu thay thế các phương pháp dựa trên quy tắc, và sự phát triển của các mô hình học máy bắt đầu tạo ra những bước tiến đáng kể. Các hệ thống NLP giai đoạn này có khả năng xử lý các vấn đề phức tạp hơn của ngôn ngữ tự nhiên, từ việc phân tích cú pháp đến nhận diện thực thể và phân loại văn bản.

Giai đoạn thứ ba, từ năm 2000 đến nay là thời kỳ phát triển mạnh mẽ nhất của kỹ thuật Xử lý Ngôn ngữ Tự nhiên (NLP), khi các tiến bộ về học máy, học sâu (deep learning), và sự phát triển của các mô hình ngôn ngữ tiên tiến đã mang lại những bước đột phá đáng kể trong khả năng xử lý và hiểu ngôn ngữ tự nhiên của máy tính. Sự chuyển mình từ các mô hình thống kê sang các mô hình học sâu đã cách mạng hóa toàn bộ lĩnh vực NLP, mở ra nhiều ứng dụng mới và cải thiện chất lượng xử lý ngôn ngữ một cách đáng kể.

1.2.3. Các thành phần và mô hình cơ bản

Các thành phần cơ bản của NLP bao gồm nhiều bước xử lý từ cấp độ thấp đến cao. Tiền xử lý văn bản là bước đầu tiên, bao gồm tokenization (phân tách văn bản thành các đơn vị nhỏ hơn), normalization (chuẩn hóa văn bản), stemming và lemmatization (đưa từ về dạng gốc). Tiếp theo là phân tích hình thái học và cú pháp, giúp xác định cấu trúc ngữ pháp của câu. Phân tích ngữ nghĩa và ngữ dụng đi sâu hơn vào ý nghĩa của từ và câu trong ngữ cảnh cụ thể. Các bước này tạo nền tảng cho việc hiểu và xử lý ngôn ngữ tự nhiên một cách toàn diện.

Trong những năm gần đây, sự phát triển của các kỹ thuật và mô hình mới đã đưa NLP lên một tầm cao mới. Word Embeddings như Word2Vec và GloVe cho phép biểu diễn từ dưới dạng vector số trong không gian đa chiều, giúp máy tính nắm bắt được mối quan hệ ngữ nghĩa giữa các từ. Mô hình Transformer, với

cơ chế Attention, đã cách mạng hóa cách xử lý dữ liệu ngôn ngữ, cho phép xử lý song song và hiệu quả hơn. Các mô hình tiên tiến như BERT và GPT, dựa trên kiến trúc Transformer, đã đạt được kết quả ấn tượng trong nhiều nhiệm vụ NLP, từ dịch máy đến sinh văn bản tự nhiên.

1.2.4. Các kỹ thuật và công nghệ trong xử lý ngôn ngữ tự nhiên

NLP kết hợp các phương pháp từ máy học, xử lý ngôn ngữ tự nhiên và lĩnh vực trí tuệ nhân tạo để xử lý và hiểu ngôn ngữ một cách tự nhiên giống như con người. Thông thường, việc triển khai NLP bắt đầu bằng cách thu thập và chuẩn bị dữ liệu văn bản hoặc giọng nói phi cấu trúc từ các nguồn khác nhau.

Các bước hoạt động của NLP bao gồm:

Tiền xử lý dữ liệu

- Tokenization: Bước đầu tiên trong NLP là tách văn bản thành các phần tử nhỏ hơn gọi là "tokens" như từ, câu, v.v.
- Stopword Removal: Loại bỏ các từ phổ biến không mang ý nghĩa như "và", "là", giúp tập trung vào các từ khóa quan trọng hơn.
- Stemming và Lemmatization: Chuẩn hóa các từ về dạng gốc (stem) hoặc từ điển (lemma) để giảm đa dạng từ vựng.

Đào tạo mô hình

Trong quá trình đào tạo, các chuyên gia sử dụng dữ liệu đã được tiền xử lý và kỹ thuật máy học để huấn luyện các mô hình NLP. Điều này giúp cải thiện khả năng của thuật toán trong việc hiểu và xử lý thông tin ngôn ngữ tự nhiên. Để đạt hiệu suất cao, việc cung cấp cho phần mềm các tập dữ liệu lớn là cần thiết để tối ưu hóa và nâng cao độ chính xác của mô hình.

Triển khai và suy luận

Kế tiếp, những chuyên gia về học máy sẽ tiến hành triển khai mô hình hoặc tích hợp nó vào một môi trường sản xuất sẵn có. Mô hình NLP sẽ tiếp nhận dữ

liệu đầu vào và tạo ra dự đoán cho các trường hợp sử dụng cụ thể mà nó được thiết kế. Bằng cách này, bạn có thể chạy ứng dụng NLP trực tiếp trên dữ liệu và nhận kết quả dự đoán theo nhu cầu của mình, cung cấp sự linh hoạt và đáp ứng nhanh chóng cho các yêu cầu xử lý ngôn ngữ tự nhiên.

1.2.5. Ứng dụng và thách thức xử lý ngôn ngữ tự nhiên

Ứng dụng của NLP ngày càng đa dạng và rộng rãi trong cuộc sống hàng ngày. Dịch máy tự động giúp phá vỡ rào cản ngôn ngữ, trong khi trợ lý ảo như Siri hay Google Assistant tạo ra cách thức tương tác mới giữa người và máy. Phân tích cảm xúc được sử dụng rộng rãi trong nghiên cứu thị trường và quản lý thương hiệu. Chatbot và hệ thống đối thoại tự động hóa dịch vụ khách hàng, trong khi hệ thống trả lời câu hỏi cải thiện trải nghiệm tìm kiếm thông tin. NLP cũng đóng vai trò quan trọng trong việc tóm tắt văn bản, phân loại tài liệu và sinh nội dung tự động, mở ra nhiều khả năng mới trong xử lý và tạo ra thông tin.

Tuy nhiên, lĩnh vực NLP vẫn phải đối mặt với nhiều thách thức. Xử lý đa ngôn ngữ và ngôn ngữ ít tài nguyên vẫn là một vấn đề phức tạp. Cải thiện khả năng hiểu ngữ cảnh và suy luận của máy tính là một mục tiêu quan trọng để đạt được sự hiểu biết sâu sắc hơn về ngôn ngữ tự nhiên. Tăng tính giải thích được và minh bạch của mô hình NLP cũng là một yêu cầu cấp thiết, đặc biệt khi các ứng dụng NLP ngày càng ảnh hưởng đến quyết định quan trọng trong nhiều lĩnh vực.

Trong tương lai, NLP hứa hẹn sẽ tiếp tục phát triển mạnh mẽ. Việc tích hợp kiến thức chuyên ngành vào mô hình NLP sẽ mở ra khả năng ứng dụng trong các lĩnh vực chuyên biệt như y tế, luật pháp hay tài chính. Xử lý ngôn ngữ đa phương thức, kết hợp với hình ảnh và âm thanh, sẽ tạo ra các hệ thống AI toàn diện hơn. Tuy nhiên, cùng với sự phát triển này, các vấn đề đạo đức và xã hội liên quan đến NLP cũng cần được quan tâm và giải quyết kịp thời.

NLP không chỉ là công nghệ mà còn là sức mạnh đưa con người và máy tính gần nhau hơn thông qua ngôn ngữ tự nhiên. Để phát triển bền vững, cần tiếp

tục nghiên cứu và ứng dụng NLP một cách sáng tạo và đảm bảo tính minh bạch, an toàn và bảo mật thông tin cho mọi người.

1.3. TỔNG QUAN VỀ MÔ HÌNH NGÔN NGỮ LỚN

1.3.1. Giới thiệu mô hình ngôn ngữ lớn

Mô hình Ngôn ngữ Lớn (Large Language Models - LLM) đã trở thành một trong những đột phá quan trọng nhất trong lĩnh vực Xử lý Ngôn ngữ Tự nhiên (NLP) và Trí tuệ Nhân tạo (AI) trong những năm gần đây. LLM là các mô hình học sâu được đào tạo trên một lượng dữ liệu văn bản khổng lồ, có khả năng hiểu và tạo ra ngôn ngữ tự nhiên ở mức độ phức tạp và tinh tế chưa từng có. Sự xuất hiện của LLM đã mở ra một kỷ nguyên mới trong việc phát triển các ứng dụng AI có khả năng tương tác với con người một cách tự nhiên và hiệu quả.

Hình 1.3: Mô hình ngôn ngữ lớn (LLM) (Nguồn: AI business)

Các mô hình ngôn ngữ có thể có mức độ phức tạp khác nhau, từ các mô hình n-gram đơn giản đến các mô hình mạng nơ-ron mô phỏng hệ thần kinh của con người với cấu trúc vô cùng phức tạp. Tuy nhiên, thuật ngữ "Large language model" thường được sử dụng để chỉ đến các mô hình sử dụng kỹ thuật học sâu và có số lượng tham số lớn, có thể từ hàng tỷ đến hàng nghìn tỷ. Những mô hình này có khả năng phát hiện các quy luật phức tạp trong ngôn ngữ và tạo ra các văn bản có chất lượng gần như như viết bởi con người.

1.3.2. Các loại mô hình ngôn ngữ lớn

Đến thời điểm hiện tại thì hầu hết kiến trúc mà các mô hình ngôn ngữ lớn sử dụng đều là Transformer (ngoại trừ mô hình RWKV sử dụng RNN). Thông thường, có thể chia large language model thành 3 loại dựa trên mục đích:

Encoder only: Được sử dụng để xử lý dữ liệu đầu vào mà không yêu cầu phần Decoder. Trong mô hình này, phần Encoder đảm nhiệm việc mã hóa thông tin từ đầu vào và trích xuất các đặc trưng quan trọng mà không cần sinh ra đầu ra tương ứng. Mô hình này thường được áp dụng trong các tác vụ như phân loại văn bản, tóm tắt văn bản hoặc bất kỳ nhiệm vụ nào mà không đòi hỏi việc dự đoán hoặc tạo ra đầu ra mới. Sự linh hoạt và hiệu quả của Encoder only khi mã hóa thông tin từ dữ liệu đầu vào và trích xuất đặc trưng đã giúp nó trở thành một công cụ quan trọng trong xử lý ngôn ngữ tự nhiên và các lĩnh vực khác của trí tuệ nhân tạo.0

Decoder only: Đặc điểm chính của mô hình này là khả năng tạo ra văn bản theo trình tự (auto-regressive), giúp sinh ra những đoạn văn bản có logic và liên kết dựa trên ngữ cảnh từ đầu vào hoặc câu hỏi đã cho. Mô hình Decoder only thường được sử dụng trong các tác vụ yêu cầu sáng tạo như hoàn thiện văn bản, tóm tắt nội dung, trả lời câu hỏi, và tạo ra văn bản độc đáo. Ưu điểm của mô hình là khả năng xử lý các tác vụ này một cách hiệu quả và logic. Tuy nhiên, một số hạn chế của mô hình này có thể bao gồm việc đòi hỏi lượng dữ liệu lớn để huấn luyện hiệu quả, cũng như có thể gặp khó khăn trong việc xử lý các tác vụ phức tạp đòi hỏi sự tương tác giữa ngữ cảnh và thông tin từ đầu vào. Để cải thiện hiệu suất, việc kết hợp mô hình Decoder only với các phần Encoder có thể là một hướng đi được ưa chuộng trong nghiên cứu và ứng dụng trong tương lai.

Encoder - Decoder: Mô hình sở hữu cả encoder và decoder như transformer, có khả năng đọc hiểu và sinh văn bản. Tuy nhiên, người dùng thường không lựa chọn mô hình dạng encoder-decoder để tăng kích thước lên thành một mô hình ngôn ngữ lớn bởi một số lý do như:

- Phức tạp và tốn kém: Khi kết hợp cả hai phần Encoder và Decoder, mô hình trở nên phức tạp hơn, đồng nghĩa với việc tăng cường tài nguyên tính toán cần thiết. Việc huấn luyện và triển khai một mô hình lớn như vậy sẽ đòi hỏi chi phí tính toán và bộ nhớ lớn, gây ra sự gia tăng đáng kể về chi phí vận hành.
- Hiệu suất và độ chính xác: Mặc dù mô hình Encoder-Decoder có khả năng đọc hiểu và sinh văn bản, việc mở rộng kích thước của nó thành mô hình ngôn ngữ lớn không nhất thiết đảm bảo cải thiện đáng kể về hiệu suất và độ chính xác. Điều này có thể dẫn đến việc mặc dù có sự gia tăng về kích thước mô hình nhưng không đem lại hiệu quả với việc cải thiện khả năng sáng tạo và đa dạng của văn bản sinh ra trong mô hình.
- Quản lý ngữ cảnh: Mô hình Encoder-Decoder có thể gặp khó khăn trong việc quản lý ngữ cảnh phức tạp và tương tác giữa thông tin từ phần Encoder và phần Decoder. Điều này có thể làm giảm khả năng sinh văn bản mạch lạc và logic trong một số trường hợp.

1.3.3. Kiến trúc của mô hình ngôn ngữ lớn

Về mặt kỹ thuật, LLM thường được xây dựng dựa trên kiến trúc Transformer, một loại mạng nơ-ron được giới thiệu vào năm 2017 bởi Vaswani và cộng sự. Kiến trúc này sử dụng cơ chế self-attention, cho phép mô hình xử lý hiệu quả các phụ thuộc dài hạn trong dữ liệu ngôn ngữ. LLM được đào tạo theo phương pháp học không giám sát, trong đó mô hình học cách dự đoán từ tiếp theo trong một chuỗi văn bản dựa trên ngữ cảnh trước đó. Quá trình đào tạo này đòi hỏi một lượng lớn dữ liệu và tài nguyên tính toán, nhưng kết quả là một mô hình có khả năng nắm bắt các mẫu và cấu trúc phức tạp của ngôn ngữ tự nhiên. Kiến trúc của LLM chủ yếu bao gồm nhiều lớp mạng nơ-ron, như recurrent layers, feedforward layers, embedding layers, attention layers. Các lớp này hoạt động cùng nhau để xử lý văn bản đầu vào và tạo dự đoán đầu ra:

Lớp nhúng chuyển đổi mỗi từ trong văn bản thành một biểu diễn số có nhiều chiều. Các biểu diễn này bao gồm thông tin về ý nghĩa và cấu trúc ngữ pháp của từng từ hoặc token trong câu và giúp mô hình hiểu bản chất của văn bản.

Các lớp truyền thẳng bao gồm nhiều lớp kết nối đầy đủ và áp dụng các phép biến đổi phi tuyến cho các vector nhúng đầu vào. Những lớp này hỗ trợ mô hình học các thông tin trừu tượng hơn từ văn bản đầu vào.

Các lớp tuần tự của LLM được tạo ra để diễn giải thông tin từ văn bản đầu vào theo trình tự. Những lớp này duy trì trạng thái ẩn được cập nhật ở mỗi bước thời gian, giúp mô hình hiểu được sự phụ thuộc giữa các từ trong câu.

Các lớp chú ý (Attention layers) là một thành phần quan trọng trong mô hình LLM, cho phép mô hình tập trung vào các phần khác nhau của văn bản đầu vào một cách có chọn lọc. Cơ chế này giúp mô hình tập trung vào những phần quan trọng nhất của văn bản đầu vào và tạo ra các dự đoán chính xác hơn.

Một trong những đặc điểm nổi bật của LLM là khả năng thực hiện đa nhiệm (multi-task) và chuyển giao học tập (transfer learning). Điều này có nghĩa là một mô hình được đào tạo trên một tập dữ liệu lớn và đa dạng có thể được tinh chỉnh để thực hiện nhiều nhiệm vụ NLP khác nhau mà không cần đào tạo lại từ đầu. Ví dụ, một LLM có thể được sử dụng cho các tác vụ như dịch máy, tóm tắt văn bản, trả lời câu hỏi, và thậm chí tạo ra nội dung sáng tạo như thơ hoặc câu chuyện ngắn. Khả năng này làm cho LLM trở nên cực kỳ linh hoạt và có giá trị trong nhiều ứng dụng thực tế.

1.3.4. Cơ chế hoạt động của mô hình ngôn ngữ lớn

Quá trình một Mô hình Ngôn ngữ Lớn (LLM) trả lời câu hỏi hoặc cung cấp gợi ý khi tương tác với người dùng bắt đầu bằng việc chuyển đổi các từ trong đoạn chat hoặc câu hỏi thành các token và sau đó biểu diễn chúng dưới dạng vector số để máy có thể hiểu. Chuỗi các vector này sau đó được đưa qua mạng nơ-ron của mô hình, bao gồm các lớp (layer) mà mỗi lớp đóng góp vào việc mô hình hiểu về mối quan hệ và ngữ cảnh giữa các token.

Cuối cùng, mô hình tạo ra một phân phối xác suất trên toàn bộ từ vựng của mình cho token tiếp theo trong chuỗi. Token có xác suất cao nhất thường được chọn làm token tiếp theo. Token này sau đó được thêm vào chuỗi văn bản ban đầu, và quá trình dự đoán token tiếp theo sẽ tiếp tục lặp lại: chuỗi mới được đưa qua mạng, và một token khác được dự đoán và thêm vào chuỗi ban đầu. Quá trình này tiếp tục cho đến khi một điều kiện dừng xảy ra, như đạt đến độ dài tối đa hoặc gặp phải một token kết thúc cụ thể. Cuối cùng, chuỗi các token đã được dự đoán được chuyển đổi hoặc giải mã trở lại thành một chuỗi văn bản có thể hiểu được.

1.3.5. Lĩnh vực ứng dụng các mô hình ngôn ngữ lớn

LLM có rất nhiều ứng dụng thực tế, bao gồm:

Viết quảng cáo: Các mô hình ngôn ngữ lớn hiện nay có thể viết quảng cáo gốc. AI21 Wordspice đề xuất những thay đổi đối với câu gốc để cải thiện văn phong và giọng điệu.

Trả lời dựa trên cơ sở kiến thức: Thường được gọi là xử lý ngôn ngữ tự nhiên chuyên sâu về kiến thức (KI-NLP), kỹ thuật này đề cập đến các LLM có khả năng trả lời những câu hỏi cụ thể dựa trên thông tin được lưu trữ trong kho lưu trữ kỹ thuật số.

Phân loại văn bản: LLM có thể phân loại văn bản có ý nghĩa hoặc quan điểm tương tự nhau bằng cách sử dụng cụm. Các trường hợp sử dụng bao gồm đo lường quan điểm khách hàng, xác định mối quan hệ giữa các văn bản và tìm kiếm tài liệu.

Tạo mã: LLM thành thạo trong việc tạo mã từ lời nhắc ngôn ngữ tự nhiên. Amazon Q Developer cho phép viết mã bằng Python, JavaScript, Ruby và nhiều ngôn ngữ lập trình khác. Các ứng dụng viết mã khác bao gồm tạo truy vấn SQL, viết lệnh shell và thiết kế trang web.

Tạo văn bản: Tương tự như tạo mã, tạo văn bản có thể hoàn tất các câu không hoàn chỉnh, viết tài liệu về sản phẩm hoặc, như Alexa Create, viết một câu chuyện ngắn dành cho trẻ em.

1.3.6. Tiềm năng và thách thức của mô hình ngôn ngữ lớn

- **Tiềm năng của mô hình ngôn ngữ lớn**

LLM đang trải qua một giai đoạn phát triển đáng kinh ngạc, nơi các mô hình như GPT (Generative Pre-trained Transformer) đã mở ra một cánh cửa mới cho lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Các mô hình LLM được huấn luyện trên lượng dữ liệu khổng lồ mà không cần sự giám sát cụ thể, từ đó đạt được khả năng hiểu ngôn ngữ và tạo ra nội dung với độ chính xác và đa dạng cao.

Mặc dù đã có những bước tiến quan trọng, tiềm năng của LLM vẫn chưa được khai thác hết. Công nghệ LLM có thể được mở rộng để áp dụng trong nhiều lĩnh vực khác nhau như y tế, tài chính, giáo dục, và nhiều lĩnh vực khác. Trí tuệ nhân tạo dựa trên ngôn ngữ có thể hỗ trợ trong việc chẩn đoán bệnh, dự báo thị trường tài chính, tạo nội dung giáo dục tự động, và nhiều ứng dụng khác.

Với việc tăng cường sự đầu tư vào nghiên cứu và phát triển, LLM có thể trở thành công cụ quan trọng trong việc giải quyết những thách thức to lớn của xã hội hiện nay. Nhìn chung, sự phát triển và tiềm năng mở rộng của Học Sâu Dựa Trên Ngôn Ngữ đang mở ra những cơ hội to lớn cho cả cộng đồng nghiên cứu và các ngành công nghiệp khác nhau, với tiềm năng ứng dụng rộng lớn và tính ứng dụng cao trong thực tế.

- **Thách thức đối với mô hình ngôn ngữ lớn**

Tuy nhiên, sự phát triển nhanh chóng của LLM cũng đặt ra nhiều thách thức và lo ngại. Một trong những vấn đề chính là tính thiên vị và độ chính xác của thông tin. LLM có thể tạo ra nội dung không chính xác hoặc thiên vị, phản ánh các định kiến có trong dữ liệu đào tạo. Điều này đặt ra câu hỏi về trách nhiệm và đạo đức trong việc sử dụng các mô hình này. Ngoài ra, việc giải thích cách LLM

đưa ra quyết định cũng là một thách thức lớn, khiến việc hiểu và kiểm soát hành vi của chúng trở nên khó khăn.

Một vấn đề khác liên quan đến LLM là tính bền vững và tác động môi trường. Quá trình đào tạo và vận hành các mô hình này đòi hỏi một lượng lớn năng lượng, góp phần vào việc tăng lượng khí thải carbon. Điều này đặt ra nhu cầu cấp thiết về việc phát triển các phương pháp đào tạo và triển khai hiệu quả hơn về mặt năng lượng.

Trong tương lai, LLM có khả năng phát triển theo hướng trở nên thông minh hơn, hiểu ngữ cảnh sâu sắc hơn, và có khả năng suy luận phức tạp hơn. Việc tích hợp LLM với các dạng dữ liệu khác như hình ảnh và âm thanh có thể dẫn đến sự ra đời của các mô hình đa phương thức mạnh mẽ. Tuy nhiên, cùng với sự phát triển này, việc giải quyết các vấn đề về đạo đức, quyền riêng tư, và tính minh bạch sẽ trở nên quan trọng hơn bao giờ hết.

Tóm lại, Mô hình Ngôn ngữ Lớn đã và đang tạo ra một cuộc cách mạng trong lĩnh vực AI và NLP. Với khả năng hiểu và tạo ra ngôn ngữ tự nhiên ở mức độ cao, LLM mở ra vô số khả năng ứng dụng trong nhiều lĩnh vực của đời sống. Tuy nhiên, để khai thác tối đa tiềm năng của công nghệ này, chúng ta cần tiếp tục nghiên cứu, phát triển, và đặc biệt là giải quyết các thách thức về đạo đức và xã hội mà nó đặt ra. Sự phát triển của LLM không chỉ là một bước tiến trong công nghệ mà còn là một cơ hội để chúng ta định hình lại cách con người tương tác với máy móc và thông tin trong thời đại số.

- **Ảnh hưởng của LLM đối với việc xử lý ngôn ngữ tự nhiên và các lĩnh vực khác**

Các mô hình ngôn ngữ lớn (LLM) đã cho thấy khả năng gần với hiệu suất của con người trong nhiều nhiệm vụ phân tích khác nhau, khiến các nhà nghiên cứu sử dụng chúng cho các phân tích tốn nhiều thời gian và công sức. Tuy nhiên, khả năng xử lý các nhiệm vụ mở và chuyên môn hóa cao trong các lĩnh vực như nghiên cứu chính sách vẫn còn là câu hỏi. Nhiều nghiên cứu chỉ ra rằng danh sách

chủ đề do LLM tạo ra có sự chồng chéo đáng kể với danh sách chủ đề do con người tạo ra, với một số trục trặc nhỏ trong việc thiếu các chủ đề cụ thể trong tài liệu. Tuy nhiên, các đề xuất của LLM có thể cải thiện đáng kể tốc độ hoàn thành nhiệm vụ, nhưng đồng thời cũng đưa ra sự thiên vị neo giữ, có khả năng ảnh hưởng đến chiều sâu và sắc thái của phân tích, đặt ra câu hỏi quan trọng về sự đánh đổi giữa hiệu quả gia tăng và rủi ro phân tích thiên vị.[2]

1.3.7. Hạn chế của mô hình ngôn ngữ lớn không ứng dụng RAG

Mô hình ngôn ngữ lớn (LLM - Large Language Model) khi không có sự hỗ trợ từ RAG (Retrieval-Augmented Generation) chủ yếu hoạt động dựa vào dữ liệu mà nó đã được huấn luyện. Các LLM như GPT, BERT, hay T5 thường được huấn luyện trên các tập dữ liệu lớn với hàng tỷ từ, giúp chúng học cách tạo ra văn bản, trả lời câu hỏi, và thực hiện các nhiệm vụ liên quan đến ngôn ngữ tự nhiên. Tuy nhiên, việc này đi kèm với một số hạn chế:

- Giới hạn kiến thức: LLM không có khả năng truy cập dữ liệu ngoài phạm vi nó đã được huấn luyện. Kiến thức của mô hình bị giới hạn ở thời điểm và nội dung dữ liệu đã được đưa vào huấn luyện, điều này có nghĩa là mô hình không thể cung cấp thông tin mới nhất hoặc cập nhật.
- Sinh ra thông tin không chính xác: Khi gặp những câu hỏi yêu cầu thông tin chi tiết, đặc biệt là các thông tin mới hoặc hiếm gặp, LLM có thể sinh ra các câu trả lời không chính xác hoặc suy đoán. Vì mô hình không có khả năng truy cập vào dữ liệu bên ngoài, các câu trả lời chỉ dựa vào kiến thức "cố định" từ khi mô hình được huấn luyện.

Khả năng nhớ lâu dài: Mặc dù LLM rất mạnh trong việc xử lý ngôn ngữ và tạo ra các câu trả lời trôi chảy, mô hình có thể gặp khó khăn trong việc đưa ra thông tin chính xác nếu câu hỏi yêu cầu kiến thức chi tiết hoặc có tính lịch sử, vì mô hình không có cơ chế nhớ dữ liệu ngoài phạm vi đã học.

1.4. KẾT LUẬN CHƯƠNG

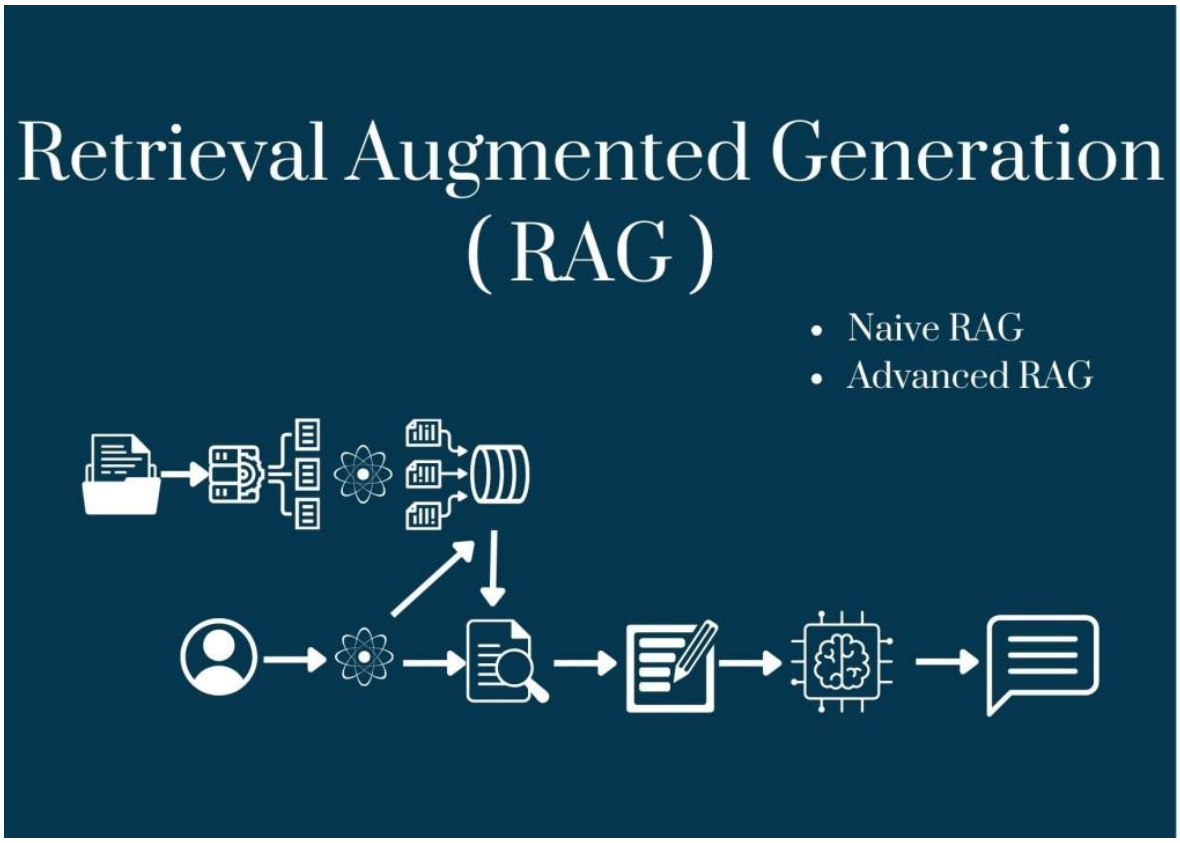
Chương 1 đã đưa ra cái nhìn tổng quan về các cơ sở lý thuyết của đề án với những nội dung tổng quát liên quan đến lĩnh vực hỏi đáp tự động. Những kiến thức tổng quan về trí tuệ nhân tạo, xử lý ngôn ngữ tự nhiên và mô hình ngôn ngữ lớn có thể giúp người đọc có cái nhìn sâu rộng hơn khi tiến tới ứng dụng cụ thể vào bài toán xây dựng một hệ thống hỏi đáp tự động như thế nào.

CHƯƠNG 2. CẢI THIỆN MÔ HÌNH NGÔ NGỮ LỚN DỰA TRÊN RAG FRAMEWORK

2.1. RAG FRAMEWORK

2.1.1. Giới thiệu về RAG Framework

Trong lĩnh vực trí tuệ nhân tạo (AI) và học máy (Machine Learning), việc kết hợp các mô hình học sâu với hệ thống truy xuất thông tin là một hướng đi ngày càng phổ biến. Một trong những framework tiên tiến và hiệu quả nhất trong việc này là RAG (Retrieval-Augmented Generation) framework. 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.



Hình 2.1: Retrieval Augmented Generation (RAG) (Nguồn: linkedin)

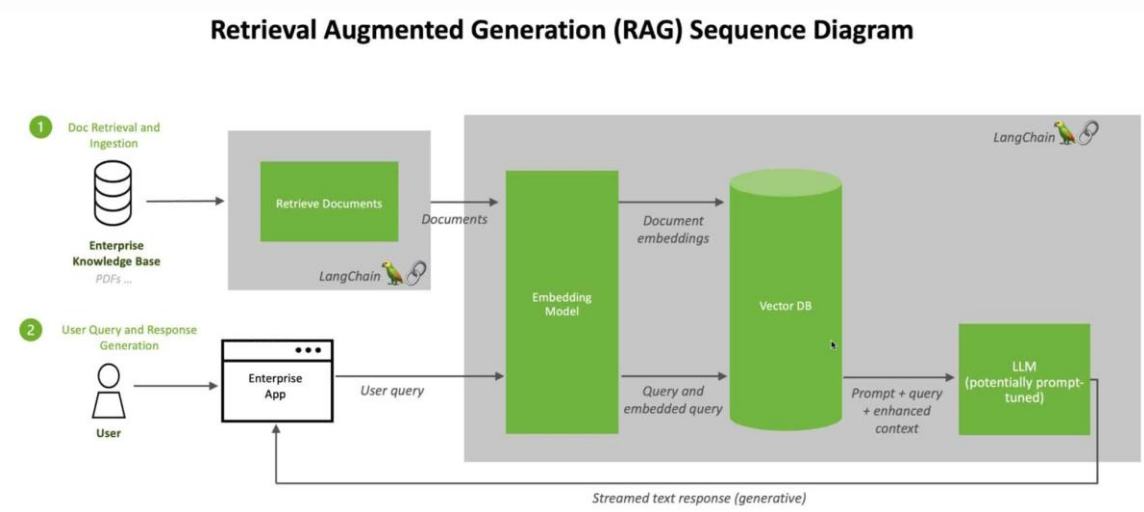
RAG là sự kết hợp giữa hai phương pháp chính: truy xuất thông tin (retrieval) và tạo sinh văn bản (generation). Điều này cho phép RAG tạo ra các câu trả lời có tính ngữ cảnh cao và chất lượng từ các nguồn thông tin ngoài phạm vi dữ liệu đã huấn luyện ban đầu.

Khái niệm về mô hình RAG được xây dựng dựa trên việc tích hợp hai thành phần cốt lõi của NLP: Truy xuất thông tin (IR) và Tạo ngôn ngữ tự nhiên (NLG). Bằng cách truy xuất rõ ràng các đoạn có liên quan từ một kho văn bản lớn và tăng cường thông tin này trong quá trình tạo, các mô hình RAG nâng cao nền tảng thực tế của kết quả đầu ra từ kiến thức cập nhật. Quy trình làm việc chung của hệ thống Tạo tăng cường truy xuất (RAG), cho thấy cách nó tăng cường cơ bản khả năng của Mô hình ngôn ngữ lớn (LLM) bằng cách căn cứ đầu ra của chúng theo thời gian thực.[3]

2.1.2. Cấu trúc và nguyên lý hoạt động của RAG

Với RAG, khả năng của LLM sử dụng kiến thức và thông tin không phải được lưu trữ trong trọng số của mô hình (không chỉ dựa vào những gì đã học) được thúc đẩy bằng cách cung cấp các nguồn tri thức bên ngoài như cơ sở dữ liệu, sách, báo, trang web. Điều này thúc đẩy hoạt động của trình tìm kiếm thông tin (retriever) để khám phá các tri thức liên quan để điều chỉnh LLMs. Nhờ vào cách tiếp cận này, RAG có thể mở rộng nền tảng tri thức của LLM bằng cách tích hợp nguồn tri thức từ bên ngoài.

RAG framework bao gồm hai thành phần chính: retriever và generator. Đầu tiên, retriever sẽ tìm kiếm các đoạn văn bản hoặc thông tin liên quan từ một tập dữ liệu lớn, thường là từ cơ sở dữ liệu hoặc các tài liệu văn bản đã được lưu trữ. Thông tin này được cung cấp cho generator, là một mô hình ngôn ngữ học sâu, thường là một biến thể của mô hình Transformer như BERT hoặc GPT, để tạo ra các phản hồi tự nhiên và chính xác dựa trên thông tin được truy xuất.



Hình 2.2: Sơ đồ trình tự một hệ thống hỏi đáp tự động sử dụng RAG

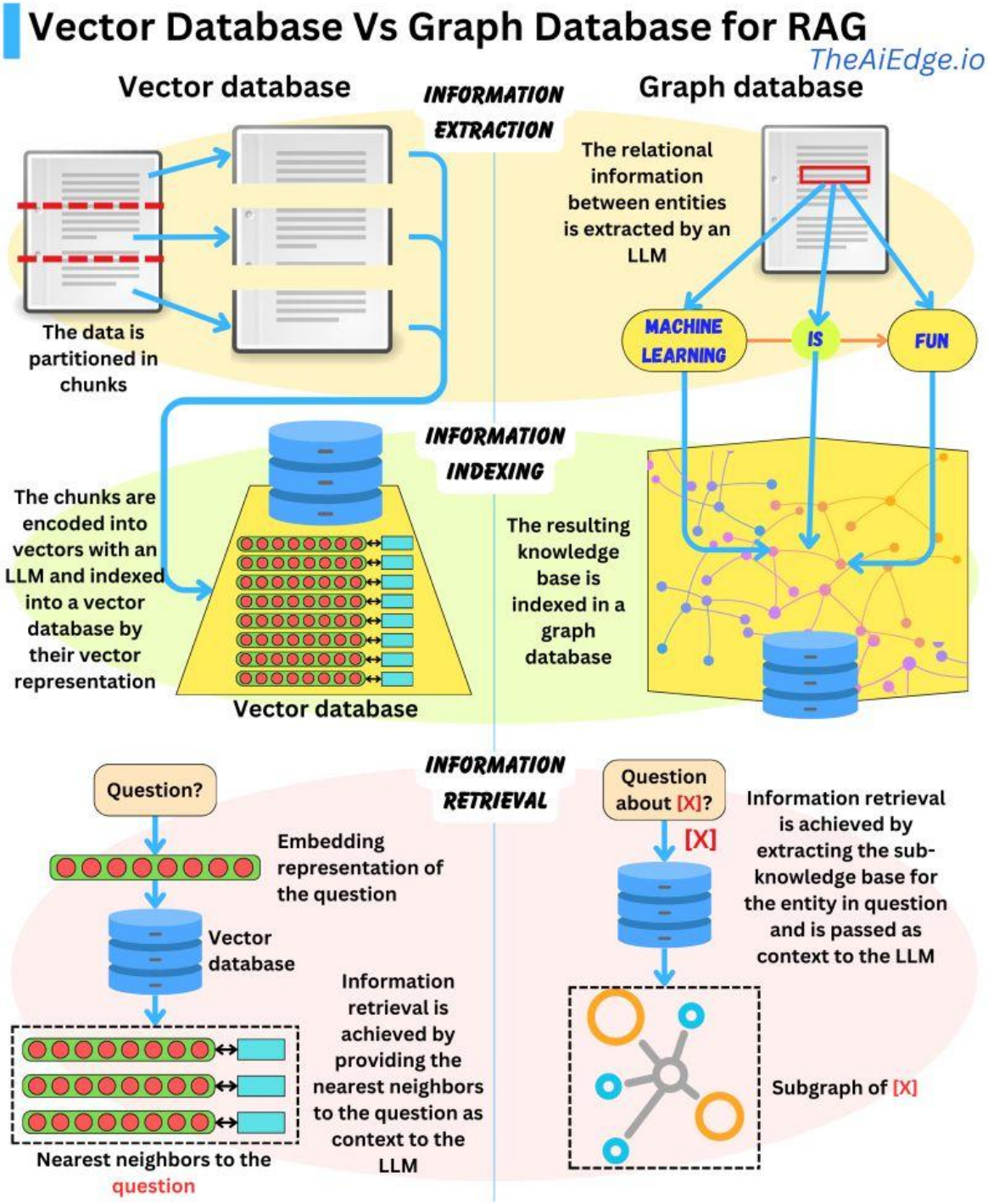
Cụ thể, khi một truy vấn được đưa vào hệ thống, retriever sẽ tìm kiếm các đoạn văn có liên quan từ cơ sở dữ liệu ngoài (ví dụ như Wikipedia). Sau đó, các đoạn văn này sẽ được đưa vào mô hình generator để tạo ra câu trả lời hoàn chỉnh. Điều này giúp hệ thống có khả năng không chỉ dựa vào kiến thức nội bộ của mô hình mà còn tận dụng các thông tin mới nhất, cập nhật từ nhiều nguồn khác nhau.

Công cụ truy xuất (retriever) ở đây có thể là bất kỳ công cụ nào sau đây tùy thuộc vào nhu cầu về ngữ nghĩa hay không:

- **Cơ sở dữ liệu Vector:** Các câu truy vấn được nhúng bằng cách sử dụng mô hình như BERT (dựa trên Transformers) để tạo ra nhúng vector dense. Ngoài ra, các phương pháp truyền thống như TF-IDF cũng có thể được sử dụng như nhúng thưa. Sau đó, quá trình tìm kiếm thường được thực hiện dựa trên sự tương đồng ngữ nghĩa (semantic similarity) hoặc tần suất của các thuật ngữ (term frequency).
- **Cơ sở dữ liệu Graph:** Việc xây dựng cơ sở tri thức từ các mối quan hệ giữa các thực thể được trích xuất từ văn bản. Phương pháp này đảm bảo tính chính xác của tri thức, nhưng yêu cầu truy vấn chính xác, điều này cũng có thể là một hạn chế trong một số ứng dụng.
- **Cơ sở dữ liệu Regular SQL:** Có thể giúp lưu trữ và tìm kiếm thông tin có cấu trúc nhưng có thể thiếu tính linh hoạt về mặt ý nghĩa.

Cơ sở dữ liệu đồ thị được xem là lựa chọn ưu tiên hơn cho RAG so với cơ sở dữ liệu vector vì nó chứa những tri thức chính xác cao. Trái với cơ sở dữ liệu vector, cơ sở dữ liệu đồ thị xây dựng một cơ sở tri thức từ các mối quan hệ giữa các thực thể được trích xuất từ văn bản. Điều này giúp việc truy xuất thông tin trở nên hiệu quả hơn, mặc dù yêu cầu các truy vấn phải chính xác với cách dữ liệu được cấu trúc và liên kết. Nếu không hiểu rõ cách thông tin được tổ chức và liên kết trong cơ sở dữ liệu, việc truy xuất thông tin cần thiết có thể gặp khó khăn. Graph database là một dạng cơ sở dữ liệu được thiết kế đặc biệt để lưu trữ thông tin dưới dạng đồ thị, với các đỉnh biểu diễn thực thể và các cạnh biểu diễn mối quan hệ giữa chúng. Trong ngữ cảnh của hệ thống hỏi đáp tự động, Graph database có thể được sử dụng để lưu trữ tri thức và mối quan hệ giữa các thực thể, từ đó giúp hệ thống hiểu rõ ngữ cảnh và cung cấp câu trả lời chính xác hơn. Việc sử dụng Graph database trong hệ thống hỏi đáp tự động cho phép hệ thống phân tích các mối quan hệ phức tạp giữa các thực thể và dựa vào đó để đưa ra câu trả lời thông minh. Thay vì chỉ dựa vào dữ liệu cấu trúc như trong các hệ cơ sở dữ liệu quan hệ truyền thống, Graph database cho phép hệ thống xử lý dữ liệu phi cấu trúc một cách linh hoạt và hiệu quả.

Mặt khác, cơ sở dữ liệu vector chia và lập chỉ mục dữ liệu bằng cách sử dụng vector dựa trên LLM để mã hóa, cho phép tìm kiếm dữ liệu dựa trên ngữ nghĩa nhưng có thể mang theo thông tin không cần thiết, dẫn đến việc có thể xảy ra sai số. Khi hệ thống nhận được một câu hỏi mới, cơ sở dữ liệu vector được sử dụng để tìm kiếm các câu có liên quan nhất dựa trên sự tương đồng vector giữa câu hỏi mới và các câu trong cơ sở dữ liệu. Quá trình này giúp hệ thống xác định được các câu trả lời tiềm năng và cung cấp cho người dùng các thông tin chính xác và đáng tin cậy. Việc sử dụng Vector database trong RAG framework giúp hệ thống hiểu rõ hơn ngữ cảnh của câu hỏi, từ đó cải thiện khả năng tìm kiếm và trả lời câu hỏi một cách hiệu quả. Bằng cách ánh xạ văn bản vào không gian vector, hệ thống có khả năng xử lý thông tin một cách nhanh chóng và chính xác, đồng thời tối ưu hóa trải nghiệm người dùng khi tương tác với hệ thống hỏi đáp tự động.

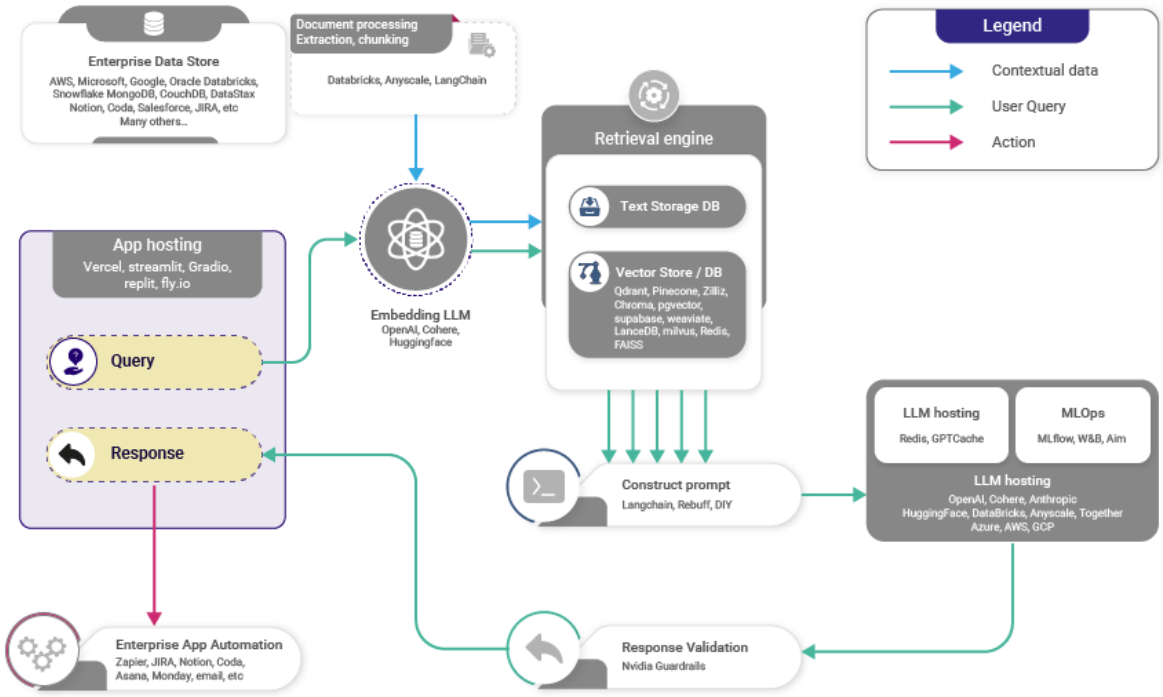


Hình 2.3: So sánh Graph với Vector database (Nguồn: TheAiEdge.io)

Sau quá trình trích xuất thông tin, có thể xem xét và lọc các kiến thức liên quan bằng cách áp dụng các bộ lọc để loại bỏ những kiến thức không phù hợp với các quy tắc cụ thể mà người dùng đang sử dụng, hoặc không phù hợp với sở thích cá nhân, bối cảnh hiện tại hoặc giới hạn về độ dài.

Một quá trình hoạt động của RAG sử dụng vector database được tóm gọn như sau:

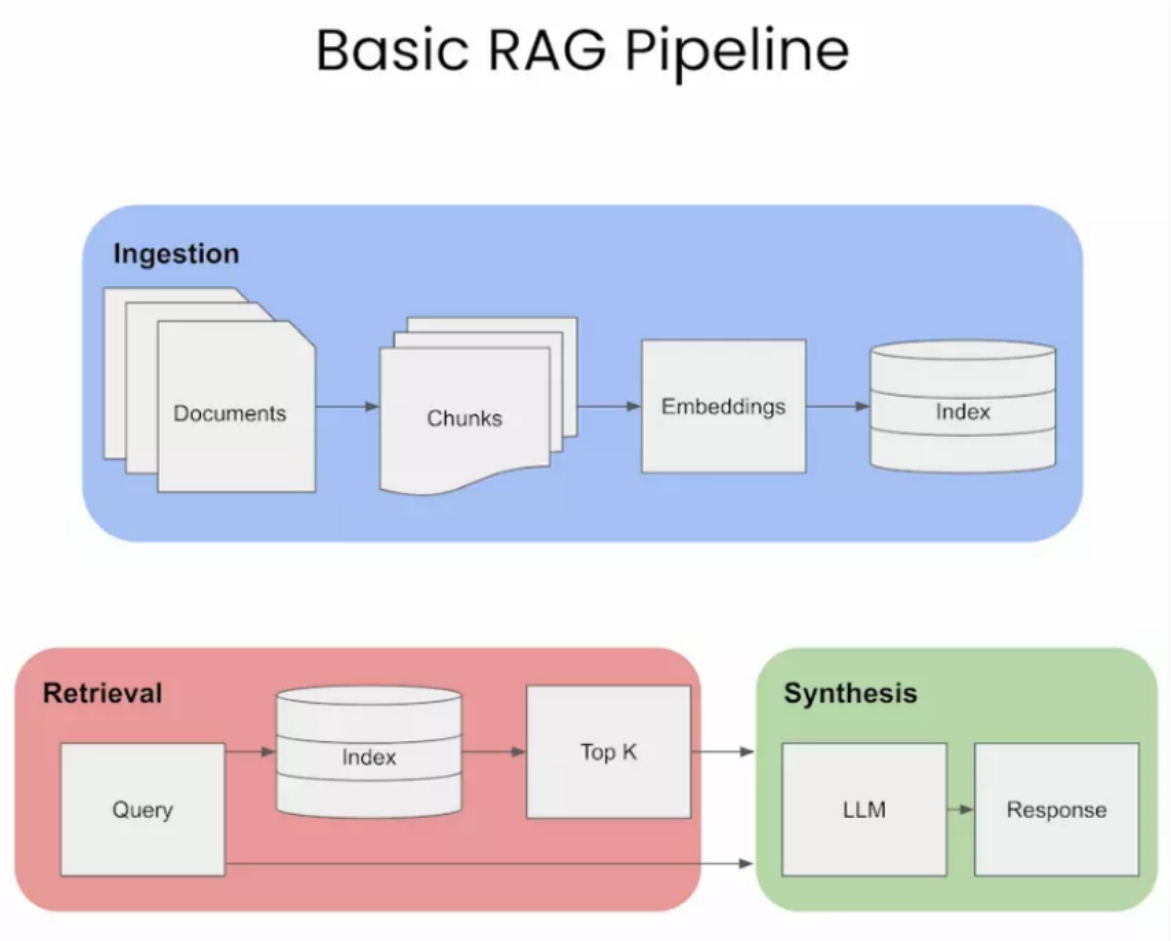
- **Create Vector database:** Đầu tiên, chuyển đổi những kiến thức trong tài liệu thành các vector và lưu trữ chúng vào một cơ sở dữ liệu vector.
- **User input:** Người dùng nhập các câu hỏi bằng ngôn ngữ tự nhiên nhằm tìm kiếm phản hồi phù hợp hoặc để hoàn thành câu hỏi đó.
- **Information retrieval:** Cơ chế truy xuất thông tin quét toàn bộ vector trong cơ sở dữ liệu để xác định các đoạn tri thức (cụ thể là các đoạn văn) nào có ngữ nghĩa tương đồng với câu truy vấn của người dùng. Những đoạn văn này sau đó được đưa vào mô hình ngôn ngữ lớn (LLM) để mở rộng bối cảnh cho quá trình tạo ra câu trả lời.
- **Combining data:** Các đoạn văn được trích xuất từ cơ sở dữ liệu sau quá trình truy xuất được kết hợp với câu truy vấn ban đầu của người dùng để tạo thành một câu hỏi hoàn chỉnh.
- **Generate text:** Câu hỏi hoàn chỉnh sau khi được bổ sung thêm bối cảnh sau đó được đưa qua mô hình ngôn ngữ lớn (LLM) để sinh ra các câu trả lời cuối cùng dựa trên ngữ cảnh bổ sung đó.



Hình 2.4: Quá trình hoạt động của RAG

2.1.3. Các thành phần của RAG framework

Đánh giá một cách tổng quan, RAG cần có 3 bước để hoàn thành chu trình làm việc: Ingestion (Biến đổi dữ liệu), Retrieval (Truy xuất thông tin), và Synthesis/ Response Generation (Tổng hợp và sinh câu trả lời)



Hình 2.5: Đường ống RAG cơ bản

- **Ingestion**

Ingestion hay còn gọi là quá trình nhập dữ liệu, là quá trình tổng quát của việc biến đổi dữ liệu. Nó bao gồm các giai đoạn cơ bản sau:

- Thu thập dữ liệu
- Tiền xử lý dữ liệu
- Lập chỉ mục và lưu trữ vào database (Indexing/ Embedding/ Storage DB)

Chunking

Chunking (phân đoạn) là quá trình chia các câu hỏi hoặc tài liệu thành các phần nhỏ hơn, được gọi là "chunks", sao cho mỗi phân đoạn vẫn giữ nguyên ý nghĩa ban đầu. Mỗi chunk sau đó được thay đổi thành một vector nhúng để sử dụng trong quá trình truy xuất thông tin trong RAG. Việc chia các phân đoạn một cách chính xác và đủ nhỏ giúp cải thiện khả năng kết nối giữa câu truy vấn của

người dùng và nội dung chunk tương ứng. Nếu các phân đoạn quá lớn, chúng có thể chứa quá nhiều thông tin không liên quan, dẫn đến nhiễu và làm giảm độ chính xác của quá trình truy xuất thông tin. Bằng cách kiểm soát kích thước của các phân đoạn, RAG có thể đạt được sự cân bằng giữa lượng thông tin đầy đủ và độ chính xác trong quá trình xử lý dữ liệu. Việc lựa chọn kích thước của chunking trong RAG là rất quan trọng. Nó cần đủ nhỏ để đảm bảo tính liên quan và giảm nhiễu nhưng cũng cần đủ lớn để duy trì đủ dữ liệu và ngữ cảnh.

Có một số phương pháp Chunking thường được sử dụng:

- **Fixed-size chunking:** Lựa chọn số lượng tokens trong mỗi phân đoạn là quyết định quan trọng trong quá trình chunking. Việc quyết định có nên tạo overlap giữa các chunks hay không cũng đóng vai trò quan trọng trong việc bảo toàn ngữ cảnh giữa chúng.
- **Context-aware chunking:** là phương pháp sử dụng cấu trúc nội bộ của văn bản để chia thành các chunks, tạo ra các đơn vị văn bản nhỏ hơn mang ý nghĩa và phù hợp hơn với ngữ cảnh. Điều này giúp đảm bảo rằng các phân đoạn được tạo ra từ cùng một phần của văn bản hoặc từ các phần có liên quan trong văn bản, giữ cho thông tin có liên kết và dễ hiểu hơn. Việc áp dụng context-aware chunking giúp cải thiện khả năng hiểu và xử lý dữ liệu, đặc biệt là trong các tác vụ liên quan đến ngôn ngữ tự nhiên. Bằng cách này, các chunks sẽ bám sát với cấu trúc và ý nghĩa của văn bản gốc, giúp tạo ra thông tin được tổ chức một cách logic và dễ dàng trích xuất ngữ cảnh cần thiết. Có các kỹ thuật cơ bản được sử dụng cho phương pháp này
    - **Sentence Splitting:** Phương pháp này được đánh giá là tối ưu với các mô hình ưu tiên cho việc nhúng các nội dung của tài liệu theo cấp độ câu (sentence-level) bao gồm:
        - Naive Splitting: Phương pháp chia câu thành các phần nhỏ bằng cách sử dụng dấu chấm hoặc dấu xuống dòng là một cách tiếp cận cơ bản để tách văn bản thành các đơn vị nhỏ hơn. Phương pháp này đơn giản và nhanh chóng, giúp tạo ra các

chunks dựa trên cấu trúc câu gốc. Tuy nhiên, nhược điểm của phương pháp này là có thể bỏ qua các câu có cấu trúc phức tạp hoặc không tuân thủ theo cấu trúc tiêu chuẩn của ngôn ngữ, dẫn đến việc mất mát context quan trọng của các câu đó. Các câu phức tạp hoặc câu không tuân thủ cấu trúc chung có thể bị chia nhỏ thành các phần không hợp lý, làm mất đi sự liên kết và ý nghĩa của chúng.

- NLTK (Natural Language Toolkit): Là một thư viện phổ biến và toàn diện được sử dụng để xử lý ngôn ngữ tự nhiên trong Python. NLTK cung cấp các công cụ và tài nguyên tiện ích để thực hiện các tác vụ xử lý ngôn ngữ tự nhiên, bao gồm phân tích ngữ pháp, phân loại văn bản, và nhiều tác vụ khác. Một trong những tính năng quan trọng của NLTK là bộ phân tách câu (sentence tokenizer) giúp phân tách văn bản thành các câu một cách hiệu quả. Bằng cách sử dụng bộ sentence tokenizer của NLTK, người dùng có thể dễ dàng phân chia văn bản thành các câu riêng biệt để tiến hành xử lý và phân tích ngữ cảnh của từng câu một cách chi tiết.
- spaCy: Là một thư viện nâng cao cho các tác vụ trong NLP, cung cấp khả năng phân đoạn các câu hiệu quả.
- Vncorenlp: Là một thư viện phổ biến sử dụng cho Tiếng Việt. Cung cấp các chức năng như phân tích ngữ cảnh, phân tách từ, phân loại từ loại, phân tích cú pháp, và nhiều tính năng khác để giúp xử lý văn bản tiếng Việt một cách hiệu quả. Bộ tokenizer của VnCoreNLP giúp phân tách văn bản thành các cụm từ và câu một cách chính xác, hỗ trợ trong việc xử lý và phân tích ngôn ngữ tự nhiên.

- **Recursive Chunking:** Là một phương pháp trong xử lý ngôn ngữ tự nhiên giúp chia văn bản theo cấp độ bằng cách sử dụng các dấu phân

tách khác nhau. Phương pháp này cho phép tạo ra các phần văn bản có độ lớn hoặc cấu trúc gần giống nhau bằng cách ứng dụng phương pháp đệ quy với nhiều tiêu chí. Khi sử dụng recursive chunking, văn bản được chia nhỏ thành các phần con dựa trên các quy tắc cú pháp hoặc ngữ cảnh cụ thể, và quá trình này có thể được lặp lại để tạo ra các chunk chi tiết hơn hoặc theo cấp độ cao hơn tùy thuộc vào mục tiêu xử lý. Phương pháp này linh hoạt và có thể được điều chỉnh để phù hợp với nhu cầu xử lý dữ liệu và phân tích ngôn ngữ của mỗi ứng dụng cụ thể.

- **Specialized Chunking:** Được sử dụng cho các dữ liệu có định dạng như Markdown hoặc LaTeX, các quy tắc phân tách và xử lý dữ liệu được tinh chỉnh để phù hợp với cấu trúc đặc biệt của định dạng đó. Điều này giúp đảm bảo rằng các phần của văn bản được chia thành chunks vẫn giữ nguyên cấu trúc và ý nghĩa ban đầu của nó, mà không làm mất thông tin quan trọng hoặc định dạng đặc biệt.
  - Markdown Chunking: Xác định cú pháp Markdown trong văn bản và chia chúng dựa theo cấu trúc.
  - LaTeX Chunking: Có thể phân tích các command có trong văn bản để chia phần nội dung mà vẫn giữ được cấu trúc của nó.

Embeddings

Việc nhúng tài liệu vào mô hình RAG là quá trình biến đổi cả câu hỏi từ người dùng và các tài liệu trong kho tri thức thành các vector trong không gian đa chiều để có thể đánh giá mức độ tương quan giữa chúng. Quá trình này đóng vai trò quan trọng trong khả năng của RAG để truy xuất thông tin phù hợp nhất từ kho tri thức với câu hỏi của người dùng. Để lựa chọn mô hình nhúng phù hợp cho vấn đề, việc tham khảo các bộ benchmark của các mô hình nhúng trên các tập dữ liệu lớn cho các nhiệm vụ cụ thể là rất quan trọng. Thông thường, việc dựa vào kết quả của các mô hình nhúng trên các benchmark giúp xác định mô hình nào phù hợp nhất cho vấn đề cụ thể cần giải quyết. Việc này giúp tăng cường hiệu suất

và chất lượng của quá trình nhúng tài liệu và tối ưu hóa khả năng truy xuất thông tin của mô hình RAG.

- Sparse embedding thường được ứng dụng để đối chiếu từ vựng giữa câu truy vấn và tài liệu. Đây là sự lựa chọn thích hợp giữa các từ khóa có sự tương đồng lớn. Sparse embedding tốn ít chi phí tính toán hơn so với các phương pháp khác, tuy nhiên, nhược điểm là có thể không thể hiện được ngữ nghĩa của nội dung.
- Semantic embedding (hoặc Dense embedding) thường được sử dụng vì cs thể hiểu ngữ cảnh của câu. Loại nhúng này được sử dụng phổ biến trong các hệ thống RAG vì nắm bắt ngữ cảnh và ngữ nghĩa hiệu quả.

- **Retrieval**

Dựa vào đặc điểm của bộ dữ liệu, mức độ phức tạp của câu truy vấn và sự cân bằng giữa tính cụ thể và hiểu biết ngữ cảnh trong việc xử lý và đáp ứng các câu hỏi, có thể phân loại Retrieval thành 2 loại sau:

- Specificity (Độ cụ thể): Là khả năng của một hệ thống cung cấp thông tin chính xác và đầy đủ cho một truy vấn cụ thể. Hệ thống có độ chi tiết cao sẽ cung cấp thông tin chính xác, thường là câu trả lời chi tiết và trực tiếp liên quan đến câu hỏi được đặt ra.
- Contextual Understanding (Hiểu biết ngữ cảnh): Hiểu biết ngữ cảnh là khả năng của hệ thống tích hợp thông tin ngữ cảnh vào câu trả lời. Hệ thống có hiểu biết ngữ cảnh tốt sẽ không chỉ cung cấp câu trả lời dựa trên thông tin cụ thể trong câu hỏi, mà còn xem xét bối cảnh, ý định ẩn, và thông tin liên quan không được nêu rõ. Điều này giúp câu trả lời trở nên sâu sắc và kết nối hơn.

- **Synthesis**

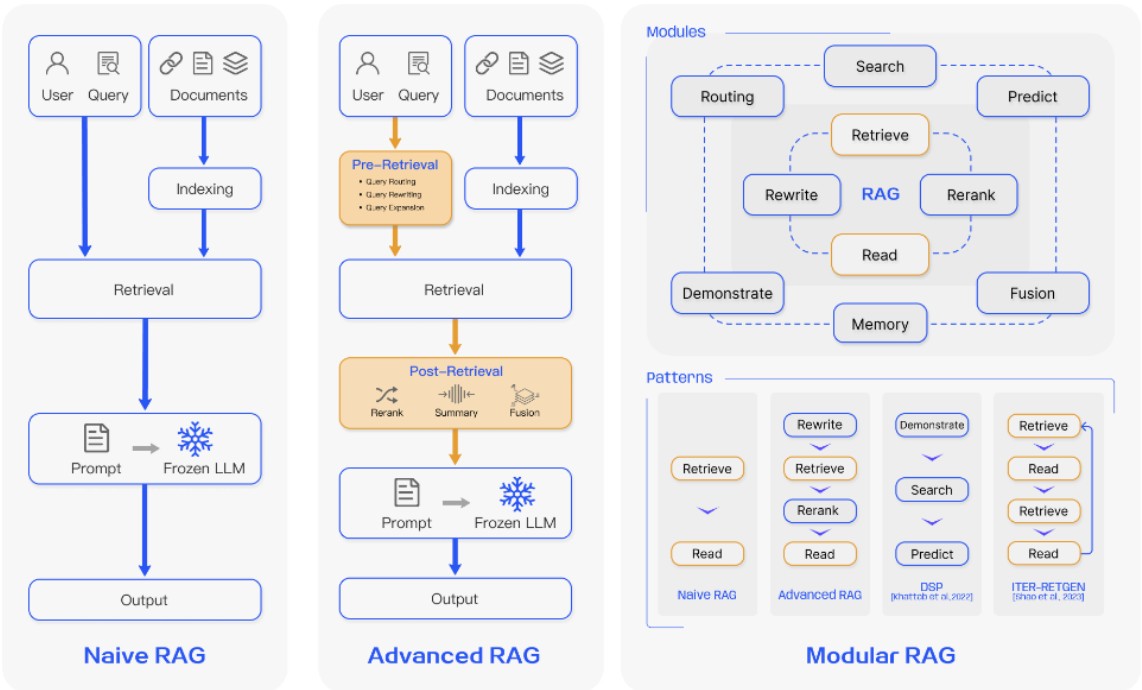
Trong hệ thống RAG, bước cuối cùng là việc sinh ra câu trả lời cho người dùng (Synthesis hoặc Response Generation). Ở bước này, mô hình tổng hợp dữ liệu thu thập được cùng với kiến thức đã được tiền huấn luyện để sinh ra các câu

trả lời tự nhiên và phù hợp với ngữ cảnh. Việc này có thể liên quan đến quá trình tích hợp tri thức từ các nguồn dữ liệu, có độ chính xác và tương đồng, từ đó sinh ra câu trả lời đảm bảo chứa thông tin và đáp ứng với câu hỏi nhận được từ người dùng, mang tính tương tác cao và có ngữ điệu tự nhiên.

Trong quá trình tạo ra câu mô tả cho mô hình ngôn ngữ lớn, để sinh ra câu trả lời, việc sắp xếp chiến lược để đặt thông tin quan trọng ở đầu hoặc cuối chuỗi đầu vào có thể tăng cường hiệu quả của hệ thống RAG và giúp hệ thống hoạt động một cách hiệu quả hơn.[4]

2.1.4. Phân loại RAG framework

Mô hình nghiên cứu RAG liên tục phát triển và hiện nay có thể phân loại nó thành ba giai đoạn: Naïve RAG, Advanced RAG và Modular RAG. Mặc dù phương pháp RAG tiết kiệm chi phí và vượt trội hơn hiệu suất của LLM gốc, nhưng chúng cũng bộc lộ nhiều hạn chế. Sự phát triển của Advanced RAG và Modular RAG là phản hồi cho những thiếu sót cụ thể này trong Naive RAG.



Hình 2.6: Các loại RAG cơ bản

Mô hình nghiên cứu Naive RAG đại diện cho phương pháp sớm nhất, đã đạt được sự nổi bật ngay sau khi áp dụng rộng rãi ChatGPT. Naive RAG tuân theo quy trình truyền thống bao gồm lập chỉ mục, truy xuất và tạo, quy trình này cũng được đặc trưng là khung "Retrieve-Read".

Advanced RAG giới thiệu những cải tiến cụ thể nhằm khắc phục những hạn chế của Naive RAG. Tập trung vào việc nâng cao chất lượng truy xuất, nó sử dụng các chiến lược truy xuất trước và sau truy xuất. Để giải quyết các vấn đề về lập chỉ mục, Advanced RAG tinh chỉnh các kỹ thuật lập chỉ mục của mình thông qua việc sử dụng phương pháp cửa sổ trượt, phân đoạn chi tiết và kết hợp siêu dữ liệu. Ngoài ra, nó kết hợp một số phương pháp tối ưu hóa để hợp lý hóa quá trình truy xuất.

Kiến trúc Modular RAG tiến bộ hơn hai mô hình RAG trước đây, mang lại khả năng thích ứng và tính linh hoạt nâng cao. Nó kết hợp các chiến lược đa dạng để cải thiện các thành phần của nó, chẳng hạn như thêm mô-đun tìm kiếm cho các tìm kiếm tương tự và tinh chỉnh trình truy xuất thông qua tinh chỉnh. Những cải tiến như mô-đun RAG được tái cấu trúc và đường ống RAG được sắp xếp lại đã được giới thiệu để giải quyết những thách thức cụ thể. Sự thay đổi theo hướng tiếp cận Modular RAG đang trở nên phổ biến, hỗ trợ cả xử lý tuần tự và đào tạo từ đầu đến cuối tích hợp trên các thành phần của nó. Bất chấp sự khác biệt của nó, Modular RAG được xây dựng dựa trên các nguyên tắc nền tảng của Advanced RAG và Native RAG, minh họa cho sự phát triển và cải tiến trong dòng RAG. [5]

2.1.5. Lợi ích của RAG framework

Một trong những lợi ích lớn nhất của RAG framework là khả năng kết hợp giữa kiến thức cố định của mô hình ngôn ngữ với kiến thức được cập nhật từ hệ thống truy xuất thông tin. Điều này rất hữu ích trong các trường hợp yêu cầu phản hồi với các thông tin mới hoặc trong các lĩnh vực có sự thay đổi nhanh chóng như y tế, kinh tế hay khoa học kỹ thuật.

Hơn nữa, việc sử dụng các đoạn văn bản được truy xuất giúp giảm thiểu khả năng mô hình tạo ra các thông tin sai lệch hoặc không chính xác, bởi thông tin được lấy từ nguồn cụ thể và đã được kiểm chứng. Điều này không chỉ cải thiện độ chính xác của mô hình mà còn tăng độ tin cậy của các kết quả đầu ra.

Ngoài ra, RAG framework còn rất linh hoạt trong việc mở rộng và tích hợp với các hệ thống khác. Bằng cách kết hợp mô hình truy xuất với mô hình sinh, người dùng có thể dễ dàng mở rộng phạm vi thông tin mà mô hình có thể truy cập mà không cần phải huấn luyện lại toàn bộ mô hình.

2.1.6. Các ứng dụng của RAG framework

RAG framework được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau, từ trợ lý ảo, chatbot, đến các hệ thống hỗ trợ quyết định. Trong lĩnh vực y tế, RAG có thể giúp bác sĩ truy xuất thông tin từ hàng loạt tài liệu nghiên cứu và sau đó cung cấp những phân tích và đề xuất phù hợp cho từng trường hợp cụ thể. Trong kinh doanh, các trợ lý ảo được hỗ trợ bởi RAG có thể trả lời các câu hỏi phức tạp từ khách hàng bằng cách truy xuất dữ liệu từ các tài liệu kinh doanh, chính sách công ty, hoặc các báo cáo tài chính.

Hơn nữa, RAG còn có thể được sử dụng trong lĩnh vực giáo dục, giúp cung cấp tài liệu học tập phong phú và đa dạng cho học sinh, sinh viên, đồng thời tạo ra các phản hồi chính xác và mang tính học thuật cao.

RAG framework là một giải pháp tiên tiến, kết hợp giữa truy xuất thông tin và mô hình sinh văn bản, giúp tăng cường khả năng trả lời câu hỏi của các hệ thống AI và NLP. Với cấu trúc độc đáo và linh hoạt, RAG có thể cung cấp các câu trả lời chính xác, cập nhật và đáng tin cậy hơn, nhờ vào việc tận dụng kiến thức từ nhiều nguồn dữ liệu khác nhau. Nhờ vào những ưu điểm nổi bật này, RAG framework hứa hẹn sẽ là một công cụ quan trọng trong việc phát triển các ứng dụng AI tiên tiến và hiệu quả trong tương lai.

2.1.7. RAG cải thiện mô hình ngôn ngữ lớn

Các mô hình LLM luôn có một số hạn chế nhất định bao gồm:

- Vấn đề thiếu thông tin cụ thể: Các mô hình ngôn ngữ thường chỉ đưa ra câu trả lời tổng quát dựa trên dữ liệu huấn luyện. Khi người dùng yêu cầu thông tin chi tiết về sản phẩm phần mềm bạn cung cấp hoặc cần hướng dẫn giải quyết vấn đề cụ thể, mô hình ngôn ngữ truyền thống có thể không đáp ứng được yêu cầu chính xác. Nguyên nhân là do chúng không được huấn luyện trên dữ liệu cụ thể của tổ chức của bạn. Hơn nữa, dữ liệu này cũng có thể bị lỗi thời, giới hạn khả năng cập nhật thông tin mới nhất.
- Ảo giác (Hallucination): Mô hình ngôn ngữ lớn (LLM) đôi khi gặp hiện tượng "ảo giác", có nghĩa là mô hình tự tin đưa ra các phản hồi sai dựa trên thông tin không có thật. Khi không có câu trả lời chính xác cho câu hỏi của người dùng, LLM có thể tạo ra những câu trả lời không liên quan hoặc mâu thuẫn.
- Phản hồi chung: Các mô hình ngôn ngữ thường tạo ra các phản hồi tổng quát, không đặc thù cho ngữ cảnh cụ thể đang được đề cập. Điều này tạo ra một hạn chế lớn trong việc cung cấp hỗ trợ cho khách hàng, vì mỗi người có những sở thích và nhu cầu riêng biệt mà cần được đáp ứng để tạo ra trải nghiệm cá nhân hóa. RAG (Retrieval-Augmented Generation) giải quyết vấn đề này bằng cách kết hợp kiến thức tổng quát từ mô hình ngôn ngữ lớn với khả năng truy cập vào dữ liệu cụ thể, chẳng hạn như thông tin về sản phẩm và hướng dẫn sử dụng mà bạn cung cấp. Kết hợp này giúp tạo ra các phản hồi chính xác và đáng tin cậy hơn, phù hợp với yêu cầu riêng của tổ chức và người dùng.

Thông qua việc tích hợp thông tin bên ngoài, RAG cung cấp cho mô hình khả năng tương tác với người dùng một cách linh hoạt hơn, cung cấp câu trả lời chính xác và đáng tin cậy hơn. Điều này giúp tăng cường khả năng học và hiệu suất của mô hình ngôn ngữ trong việc tạo ra các câu văn tự nhiên và chính xác

hơn. Với RAG, mô hình LLM trở nên linh hoạt hơn trong việc sử dụng thông tin bên ngoài để nâng cao chất lượng của các câu được sinh ra. Từ đó, LLM có khả năng vượt qua nhiều giới hạn trên. Sự kết hợp này tạo ra một hệ thống thông minh hơn, không chỉ dựa vào các kiến thức sẵn có mà còn có thể truy cập thông tin mới từ các nguồn dữ liệu bên ngoài.

- Khả năng truy cập thông tin mới và chính xác: Với RAG, LLM có thể truy xuất thông tin từ các tài liệu hoặc cơ sở dữ liệu bên ngoài, giúp mô hình trả lời các câu hỏi yêu cầu kiến thức cập nhật. Điều này đặc biệt quan trọng khi làm việc với các chủ đề thay đổi nhanh như tin tức, y học, hoặc các lĩnh vực nghiên cứu mới.
- Tính linh hoạt và mở rộng: RAG cho phép LLM truy cập vào kho kiến thức rộng lớn mà không cần phải huấn luyện lại mô hình với dữ liệu mới. Điều này làm cho hệ thống trở nên linh hoạt hơn, có thể dễ dàng mở rộng với các nguồn thông tin khác nhau mà không làm thay đổi cấu trúc mô hình ban đầu.
- Giảm thiểu thông tin sai lệch: Thay vì chỉ dựa vào việc sinh văn bản từ kiến thức có sẵn, LLM sử dụng RAG có thể dựa vào các đoạn văn bản truy xuất được từ các tài liệu có sẵn, giúp tăng tính chính xác và tin cậy của thông tin. Điều này làm giảm khả năng sinh ra những câu trả lời sai hoặc không chính xác.
- Tối ưu hóa hiệu suất: Với RAG, LLM có thể xử lý các truy vấn phức tạp và chi tiết hơn. Ví dụ, khi người dùng yêu cầu một câu trả lời về một chủ đề khó, retriever của RAG sẽ tìm kiếm các thông tin liên quan từ cơ sở dữ liệu và cung cấp cho generator để tạo ra câu trả lời phù hợp và chính xác hơn.

Với khả năng kết hợp giữa truy xuất thông tin và sinh câu, RAG đóng vai trò quan trọng trong việc cải thiện hiệu suất và khả năng tương tác của các mô hình ngôn ngữ lớn, mở ra những tiềm năng mới trong lĩnh vực xử lý ngôn ngữ tự nhiên.

2.2. HỆ THỐNG HỎI ĐÁP

2.2.1. Hệ thống hỏi đáp là gì

Định nghĩa cổ điển của hệ thống hỏi đáp (QAS) là một chương trình máy tính xử lý đầu vào ngôn ngữ tự nhiên từ người dùng và tạo ra các phản hồi thông minh và tương đối sau đó gửi lại cho người dùng. [6]

Hệ thống hỏi đáp được phát triển để trả lời các câu hỏi được trình bày bằng ngôn ngữ tự nhiên bằng cách trích xuất câu trả lời. Sự phát triển của QAS nhằm mục đích làm cho Web phù hợp hơn với việc sử dụng của con người bằng cách loại bỏ nhu cầu sàng lọc rất nhiều kết quả tìm kiếm theo cách thủ công để xác định câu trả lời chính xác cho một câu hỏi.[7]

Hệ thống hỏi đáp liên quan đến 3 lĩnh vực lớn là xử lý ngôn ngữ tự nhiên (Natural Language Processing), tìm kiếm thông tin (Information Retrieval) và rút trích thông tin (Information Extraction).

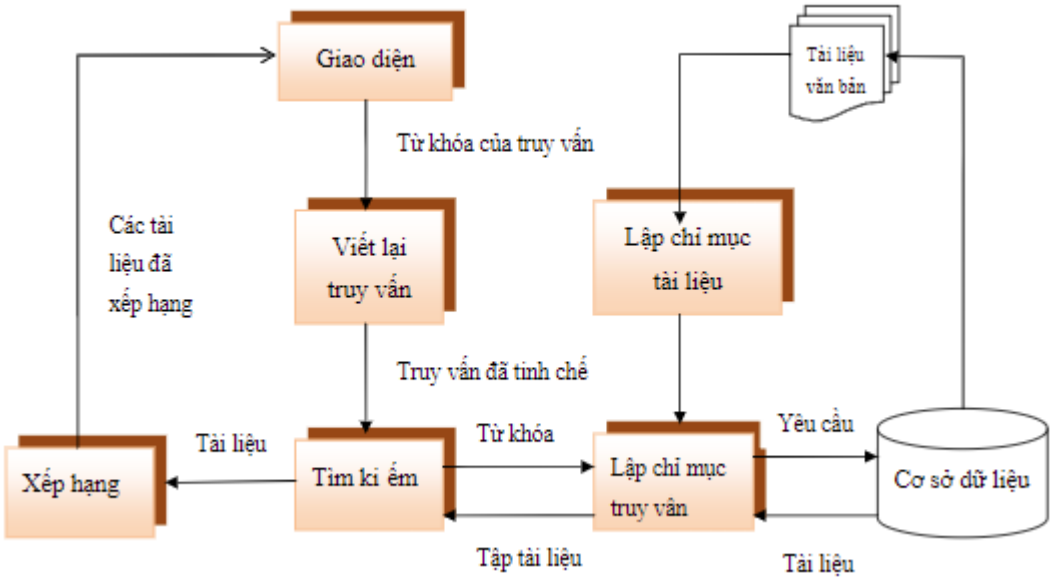
2.2.2. Các loại hệ thống hỏi đáp

Có thể chia hệ thống hỏi đáp thành 2 loại:

- Hệ thống hỏi đáp lĩnh vực hẹp (Closed-domain Question Answering): Là một loại hệ thống trả lời câu hỏi được thiết kế để hoạt động trong một lĩnh vực cụ thể hoặc trên một tập hợp hẹp các chủ đề. Thay vì xử lý mọi loại câu hỏi như các hệ thống trả lời câu hỏi tổng quát, hệ thống này chuyên tâm vào việc cung cấp câu trả lời chính xác cho các câu hỏi liên quan đến một lĩnh vực cụ thể, như y học, pháp lý, công nghệ, v.v. Điều này giúp cải thiện độ chính xác và hiệu suất của hệ thống trong việc cung cấp thông tin cho người dùng.
- Hệ thống hỏi đáp lĩnh vực rộng (Open-domain Question Answering): là một loại hệ thống trả lời câu hỏi được thiết kế để xử lý và cung cấp câu trả lời cho các câu hỏi từ một loạt các lĩnh vực và chủ đề khác nhau mà không bị hạn chế bởi một lĩnh vực cụ thể. Điều này đòi hỏi hệ thống có khả năng trích xuất thông tin từ nguồn dữ liệu đa dạng và rộng lớn để cung cấp câu trả lời chính

xác và đa chiều cho người dùng. Hệ thống hỏi đáp lĩnh vực rộng thường sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên, tìm kiếm thông tin và rút trích thông tin để hiểu và trả lời câu hỏi một cách hiệu quả. Mục tiêu của hệ thống này là cung cấp thông tin đa dạng và phong phú dựa trên nhu cầu của người dùng mà không bị giới hạn bởi một lĩnh vực hay chủ đề cụ thể. Rất nhiều phương pháp khác nhau được sử dụng như các phương pháp thống kê, dựa trên luật và phương pháp tổng hợp được sử dụng để xây dựng hệ thống hỏi đáp. Do sự phát triển ngày càng phức tạp, các hệ thống này thường sử dụng nhiều mô-đun khác nhau, như tìm kiếm thông tin, phân tích cú pháp câu, phân tích loại câu hỏi, phân tích ngữ nghĩa, và thậm chí các phương pháp suy luận để đánh giá và xếp hạng câu trả lời. Với sự gia tăng về mức độ phức tạp của các hệ thống này, việc xác định giai đoạn nào đó góp phần lớn vào hiệu quả của hệ thống trở nên khó khăn.

2.2.3. Kiến trúc hệ thống hỏi đáp



Hình 2.7:Kiến trúc hệ thống hỏi đáp (Nguồn: Vietdq)

Một kiến trúc hệ thống hỏi đáp tự động thường có:

– Giao diện người dùng (User Interface): Là một phần quan trọng của hệ thống hỏi đáp. Hầu hết các hệ thống hỏi đáp hiện đại cung cấp người dùng một giao

diện web để nhập câu hỏi. Sau khi câu hỏi được gửi, hệ thống sẽ xử lý và trả về câu trả lời dưới dạng tương tự.

- Phân tích câu hỏi (Question Analyzer) và tìm kiếm dữ liệu (Data Retrieval): Đóng vai trò quan trọng trong hệ thống hỏi đáp. Trong giai đoạn phân tích câu hỏi, thông tin cần thiết được trích lọc để sử dụng trong quá trình tìm kiếm dữ liệu. Trong giai đoạn tìm kiếm dữ liệu, việc lấy thông tin liên quan đến câu hỏi là điểm quan trọng.
- Rút trích câu trả lời (Answer Extraction): Là quá trình rút trích thông tin liên quan nhất với câu hỏi để tạo thành câu trả lời hoặc các câu trả lời.
- Xếp hạng (Ranking): Được sử dụng khi có nhiều câu trả lời. Các câu trả lời sẽ được xếp hạng dựa trên mức độ liên quan với câu hỏi của người dùng.
- Xác minh câu trả lời (Answer Verification): Giúp cải thiện chính xác bằng cách phân tích sâu hơn và so sánh câu hỏi và câu trả lời để xác minh tính hợp lý của câu trả lời.

2.2.4. Đặc điểm hệ thống hỏi đáp

Mặc dù những hệ thống này đóng vai trò quan trọng trong việc xử lý các loại câu hỏi cụ thể và đáp ứng nhu cầu thông tin cần thiết, hệ thống hỏi đáp vẫn có những hạn chế nhất định. Các hệ thống hỏi đáp có đặc điểm chung bao gồm:

- Tiếp cận dựa trên quy tắc: Hoạt động dựa vào các quy tắc cụ thể đã được lập trình trước. Thay vì sử dụng mô hình ngôn ngữ lớn, hệ thống này áp dụng các quy tắc để xử lý câu hỏi và cung cấp câu trả lời. Tuy tiếp cận này giúp hệ thống hoạt động một cách dễ dàng và dự đoán được, nhưng cũng đồng nghĩa với việc giới hạn khả năng hiểu biết tự nhiên của ngôn ngữ so với các mô hình học máy phức tạp.
- Hạn chế về từ vựng: Chỉ có khả năng xử lý một số lượng hạn chế các câu hỏi và từ ngữ. Khả năng mở rộng từ vựng của hệ thống này rất hạn chế, làm giảm tính linh hoạt và khả năng đáp ứng yêu cầu của người dùng.

- Thiếu khả năng học từ dữ liệu mới: Không có quá trình huấn luyện liên tục để cải thiện hiệu suất, hệ thống này không thể tự cải thiện theo thời gian như các mô hình có LLM. Điều này làm giảm khả năng hệ thống thích ứng với các yêu cầu mới và cải thiện chất lượng phản hồi.
- Giới hạn trong việc xử lý ngôn ngữ phức tạp: Thường gặp khó khăn trong việc xử lý ngôn ngữ tự nhiên phức tạp. Do không có khả năng học và cải thiện, hệ thống này không thể sinh ra câu trả lời linh hoạt và tự nhiên như các mô hình học máy tiên tiến. Điều này có thể dẫn đến việc hệ thống không thể hiểu hoặc đáp ứng đúng các câu hỏi phức tạp từ người dùng.

2.3. HỆ THỐNG HỎI ĐÁP TỰ ĐỘNG ỨNG DỤNG MÔ HÌNH NGÔN NGỮ LỚN

2.3.1. Khái niệm hệ thống hỏi đáp tự động ứng dụng LLM

Hiện nay, những hệ thống hỏi đáp tự động được hỗ trợ bởi các công cụ điều khiển theo quy tắc hoặc công cụ thông minh nhân tạo (AI) tương tác với người dùng chủ yếu thông qua giao diện dựa trên văn bản. Đây là một lĩnh vực nghiên cứu trong trí tuệ nhân tạo và xử lý ngôn ngữ tự nhiên. Mục tiêu chính của hệ thống này là hiểu được nội dung câu hỏi, tìm kiếm thông tin liên quan từ các nguồn dữ liệu có sẵn, và đưa ra câu trả lời chính xác, ngắn gọn và dễ hiểu.

Với việc sử dụng LLM, hệ thống có khả năng hiểu ngữ cảnh, logic và sáng tạo trong việc tạo ra câu trả lời, cung cấp cho người dùng những phản hồi chính xác và đa dạng. Không chỉ tăng cường độ chính xác và tính linh hoạt trong trả lời câu hỏi, mà còn giúp hệ thống tự động học và cải thiện mà không cần sự can thiệp của con người.

Ứng dụng LLM trong hệ thống hỏi đáp tự động không chỉ giúp tối ưu hóa trải nghiệm người dùng mà còn giúp tạo ra các dịch vụ hỗ trợ tự động hiệu quả, như chatbot, trợ lý ảo, diễn đàn trực tuyến hay các trang web cung cấp thông tin.

Điều này mở ra những triển vọng hứa hẹn về việc áp dụng công nghệ tiên tiến để cải thiện tương tác giữa con người và máy tính trong các lĩnh vực khác nhau.

Hình 2.8: Hệ thống hỏi đáp tự động (Chatbot)

Hệ thống hỏi đáp tự động có thể được phân loại theo nhiều tiêu chí khác nhau. Dựa trên nguồn dữ liệu, có thể chia thành hệ thống dựa trên tri thức đã được cấu trúc hóa (như cơ sở dữ liệu hoặc ontology) và hệ thống dựa trên văn bản không cấu trúc (như tài liệu web hoặc sách điện tử). Dựa trên loại câu hỏi, có thể phân biệt giữa hệ thống trả lời câu hỏi đơn giản (ví dụ: Ai? Cái gì? Khi nào? Ở đâu?) và hệ thống xử lý câu hỏi phức tạp đòi hỏi suy luận hoặc tổng hợp thông tin.

Sự phát triển của hệ thống hỏi đáp tự động có ý nghĩa quan trọng trong việc cải thiện tương tác giữa người và máy, nâng cao hiệu quả tìm kiếm thông tin, và hỗ trợ ra quyết định trong nhiều lĩnh vực như y tế, giáo dục, và dịch vụ khách hàng. Tuy nhiên, việc xây dựng một hệ thống hỏi đáp hiệu quả vẫn còn nhiều thách thức, đặc biệt là trong việc xử lý ngôn ngữ tự nhiên với tất cả sự phức tạp và đa nghĩa của nó.

2.3.2. So sánh hệ thống hỏi đáp tự động với hệ thống hỏi đáp

Hệ thống hỏi đáp tự động tiên tiến sử dụng trí tuệ nhân tạo và mô hình ngôn ngữ để tự động trả lời câu hỏi mà không cần can thiệp của con người, trong khi hệ thống hỏi đáp truyền thống phụ thuộc vào cơ sở dữ liệu và quy tắc lập trình cố định, thường cần sự can thiệp thủ công để cập nhật thông tin.

So sánh giữa hai loại hệ thống này giúp làm rõ sự linh hoạt, khả năng học hỏi và hiệu suất xử lý thông tin của hệ thống hỏi đáp tự động so với hệ thống truyền thống.

Bảng 2.1: Bảng so sánh hệ thống hỏi đáp có và không ứng dụng LLM

Tiêu Chí	Ứng dụng LLM	Không Ứng dụng LLM
Độ chính xác	Cao hơn, đặc biệt trong câu trả lời phức tạp và sáng tạo	Có thể hạn chế trong việc đưa ra câu trả lời phức tạp và sáng tạo
Đa dạng và linh hoạt	Có khả năng tạo ra câu trả lời đa dạng và sáng tạo	Có thể thiếu tính đa dạng và sự sáng tạo trong câu trả lời
Chi phí tính toán	Tốn kém, đòi hỏi cấu hình phần cứng mạnh mẽ	Đơn giản, không đòi hỏi tài nguyên tính toán lớn
Khả năng hiểu ngữ cảnh	Có thể gặp khó khăn trong việc hiểu ngữ cảnh phức tạp	Có thể dễ dàng kiểm soát chất lượng câu trả lời
Tính linh hoạt và mở rộng	Có khả năng mở rộng và linh hoạt trong xử lý các loại câu hỏi	Có thể hạn chế trong việc xử lý các câu hỏi phức tạp và yêu cầu sự sáng tạo

2.3.3. Kiến trúc của hệ thống hỏi đáp tự động ứng dụng LLM

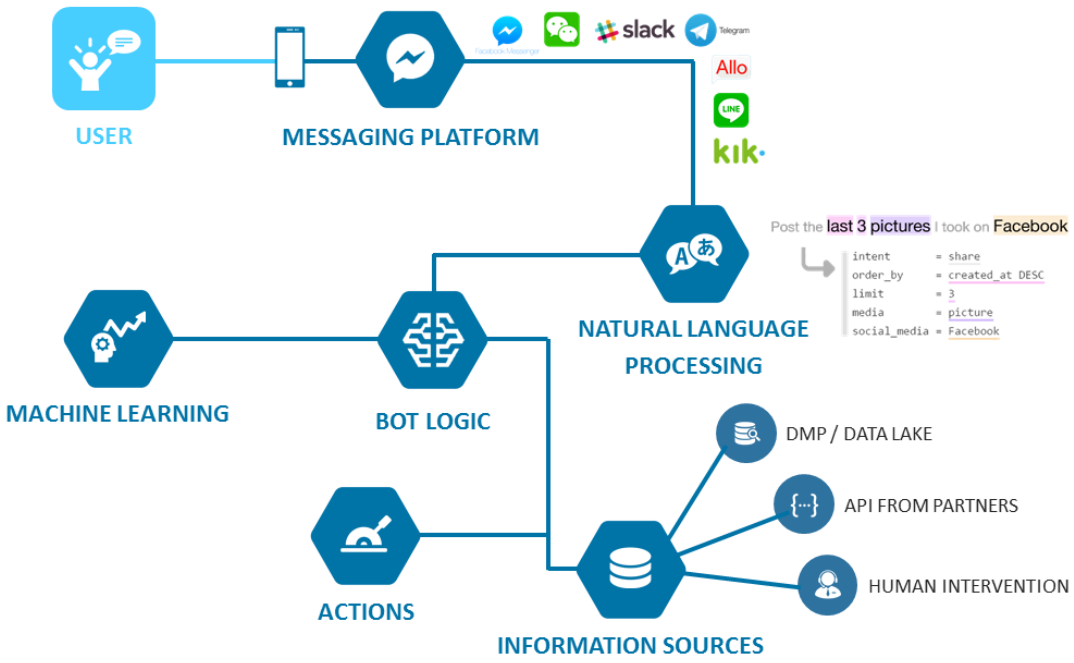
Kiến trúc của hệ thống hỏi đáp tự động ứng dụng Large Language Model (LLM) là một cấu trúc phức tạp và hiệu quả, kết hợp nhiều thành phần quan trọng để cung cấp trả lời tự động chính xác và linh hoạt cho người dùng. Mô-đun LLM có khả năng hiểu ngữ cảnh, sáng tạo và tạo ra các câu trả lời phù hợp với câu hỏi được đặt ra.

Trong hệ thống hỏi đáp tự động, bộ xử lý ngôn ngữ tự nhiên (NLP) đóng vai trò trọng tâm trong quá trình xử lý thông tin ngôn ngữ. Qua việc áp dụng kỹ thuật tokenization, câu hỏi từ người dùng được phân chia thành các đơn vị nhỏ hơn, giúp mô hình hiểu rõ cấu trúc và nội dung của câu hỏi. Tiếp theo, qua việc sử dụng embedding layer, các từ và cụm từ trong câu hỏi được biểu diễn dưới dạng các vector số học, giúp mô hình hiểu được ý nghĩa ngữ cảnh của từng từ và cụm từ trong câu hỏi.

Ngoài ra, hệ thống giao diện người dùng đóng vai trò quan trọng trong việc tương tác giữa người dùng và hệ thống. Giao diện này không chỉ giúp người dùng đưa ra câu hỏi một cách dễ dàng mà còn đảm bảo việc truyền tải thông tin giữa người dùng và hệ thống một cách hiệu quả và thuận lợi. Qua đó, có thể tương tác với hệ thống một cách tự nhiên và thuận tiện, nhận được câu trả lời chính xác và đầy đủ cho những câu hỏi mà người dùng đặt ra.

Hệ thống lưu trữ và quản lý dữ liệu chứa các thông tin quan trọng như dữ liệu huấn luyện mô hình, dữ liệu truy vấn từ người dùng và câu trả lời sinh ra. Cuối cùng, công việc tối ưu hóa mô hình và đánh giá hiệu suất giúp hệ thống ngày càng hoàn thiện và cung cấp những câu trả lời chất lượng cao cho người dùng, mở ra triển vọng sử dụng công nghệ tiên tiến để tối ưu hóa trải nghiệm tương tác giữa con người và máy tính.

Kiến trúc của một hệ thống hỏi đáp tự động thường bao gồm ba thành phần chính: xử lý câu hỏi, truy xuất thông tin, và tạo câu trả lời. Mỗi thành phần này đóng vai trò quan trọng trong quá trình chuyển đổi từ câu hỏi của người dùng thành câu trả lời cuối cùng.



Hình 2.9: Kiến trúc hệ thống hỏi đáp tự động

Xử lý câu hỏi: Đây là bước đầu tiên và quan trọng trong hệ thống. Mục tiêu của bước này là phân tích và hiểu câu hỏi của người dùng. Quá trình này thường bao gồm các công đoạn như:

- Tiền xử lý: loại bỏ nhiễu, chuẩn hóa văn bản.
- Phân tích cú pháp: xác định cấu trúc ngữ pháp của câu hỏi.
- Phân loại câu hỏi: xác định loại câu hỏi (ví dụ: Ai? Cái gì? Khi nào?).
- Trích xuất từ khóa: xác định các từ khóa quan trọng trong câu hỏi.
- Xác định loại câu trả lời mong đợi: dự đoán dạng câu trả lời (ví dụ: một ngày tháng, một tên người, một giải thích).

Truy xuất thông tin: Sau khi hiểu được câu hỏi, hệ thống sẽ tìm kiếm thông tin liên quan từ nguồn dữ liệu có sẵn. Quá trình này có thể bao gồm:

- Tìm kiếm tài liệu: sử dụng các kỹ thuật tìm kiếm thông tin để xác định các tài liệu có khả năng chứa câu trả lời.

- Lọc và xếp hạng: sắp xếp các tài liệu theo mức độ liên quan và loại bỏ những tài liệu không phù hợp.
- Trích xuất đoạn văn: xác định các đoạn văn cụ thể trong tài liệu có khả năng chứa câu trả lời.

Tạo câu trả lời: Đây là bước cuối cùng, trong đó hệ thống tổng hợp thông tin đã thu thập để tạo ra câu trả lời. Quá trình này có thể bao gồm:

- Trích xuất câu trả lời: xác định phần thông tin cụ thể trong đoạn văn đã trích xuất mà có thể trả lời câu hỏi.
- Tổng hợp câu trả lời: trong trường hợp câu hỏi phức tạp, hệ thống có thể cần tổng hợp thông tin từ nhiều nguồn.
- Tạo câu trả lời tự nhiên: chuyển đổi thông tin đã trích xuất thành một câu trả lời có cấu trúc ngôn ngữ tự nhiên.
- Xác nhận và đánh giá: kiểm tra tính nhất quán và độ tin cậy của câu trả lời.

Việc xây dựng một hệ thống hỏi đáp tự động hiệu quả đòi hỏi sự kết hợp của nhiều kỹ thuật và phương pháp khác nhau. Một số phương pháp chính được sử dụng có thể kể đến bao gồm:

Phương pháp dựa trên quy tắc (Rule-based Approach): Đây là phương pháp truyền thống, sử dụng các quy tắc được định nghĩa sẵn để xử lý câu hỏi và tìm kiếm câu trả lời. Phương pháp này hiệu quả cho các loại câu hỏi đơn giản và trong những lĩnh vực có cấu trúc rõ ràng. Tuy nhiên, nó kém linh hoạt và khó mở rộng cho các trường hợp phức tạp.

Phương pháp học máy (Machine Learning approach): Sử dụng các thuật toán học máy để tự động học cách xử lý câu hỏi và tìm kiếm câu trả lời từ dữ liệu huấn luyện. Phương pháp này bao gồm:

- Học có giám sát: sử dụng bộ dữ liệu câu hỏi-câu trả lời đã được gán nhãn để huấn luyện mô hình.

- Học không giám sát: tìm kiếm các mẫu và cấu trúc trong dữ liệu mà không cần gán nhãn trước.
- Học bán giám sát: kết hợp dữ liệu có gán nhãn và không gán nhãn.

Phương pháp học sâu (Deep Learning approach): Sử dụng các mô hình mạng nơ-ron sâu để xử lý ngôn ngữ tự nhiên và tìm kiếm câu trả lời. Các kiến trúc phổ biến bao gồm:

- Mạng nơ-ron hồi quy (RNN) và biến thể như LSTM, GRU: hiệu quả trong việc xử lý dữ liệu chuỗi như văn bản.
- Mạng nơ-ron tích chập (CNN): có thể được sử dụng để trích xuất đặc trưng từ văn bản.
- Mô hình Transformer: đặc biệt hiệu quả trong việc xử lý ngữ cảnh dài và đã dẫn đến sự phát triển của các mô hình tiên tiến như BERT, GPT.

Phương pháp kết hợp (Hybrid approach): Kết hợp ưu điểm của các phương pháp trên để tạo ra hệ thống mạnh mẽ và linh hoạt hơn. Ví dụ, có thể sử dụng quy tắc cho việc xử lý câu hỏi đơn giản và học sâu cho các trường hợp phức tạp.

Phương pháp dựa trên tri thức (Knowledge-based approach): Sử dụng các cơ sở tri thức như ontology hoặc đồ thị tri thức để hỗ trợ quá trình tìm kiếm và suy luận. Phương pháp này đặc biệt hữu ích cho các câu hỏi đòi hỏi suy luận logic hoặc kiến thức chuyên ngành.

Trong quá trình xây dựng, việc thu thập và chuẩn bị dữ liệu huấn luyện chất lượng cao là rất quan trọng. Điều này có thể bao gồm việc tạo ra bộ dữ liệu câu hỏi-câu trả lời, thu thập văn bản từ nhiều nguồn đáng tin cậy, và xây dựng cơ sở tri thức chuyên ngành.

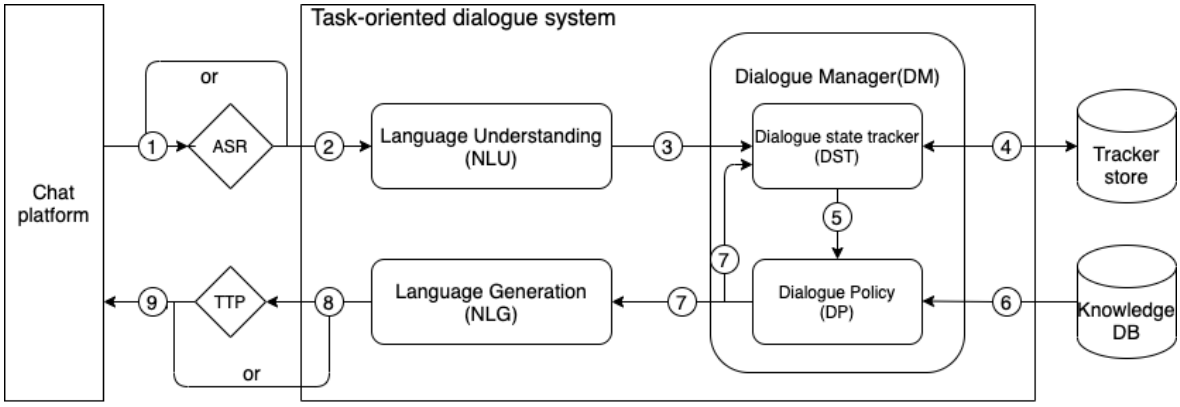
Ngoài ra, việc đánh giá và cải thiện liên tục hệ thống cũng là một phần quan trọng trong quá trình xây dựng. Điều này bao gồm việc sử dụng các metrics đánh

giá phù hợp (như độ chính xác, độ bao phủ, F1-score), thực hiện kiểm thử A/B, và thu thập phản hồi từ người dùng thực tế.

2.3.4. Phân loại hệ thống hỏi đáp tự động

- **Task-oriented dialogue systems - TODs**

TODs là các chatbot chuyên biệt được thiết kế để thực hiện những công việc cụ thể trong các lĩnh vực nhất định, chẳng hạn như đặt vé máy bay hay cung cấp thông tin về COVID-19. Các doanh nghiệp thường phát triển TODs để hỗ trợ khách hàng liên tục 24/7.



Hình 2.10: Kiến trúc một hệ thống TODs

Kiến trúc của một pipeline TODs gồm 4 module đảm nhận các vai trò khác nhau, đầu ra của module trước sẽ là đầu vào của module sau. Ngoài ra một số hệ thống TODs có thể được tích hợp thêm các dịch vụ Automatic Speech Recognition (ASR) và Text to Speech (TTS) để cung cấp khả năng tương tác bằng giọng nói cho Chatbot.

Hiện nay, TODs được ứng dụng rộng rãi trong nhiều ngành như tài chính, y tế, thương mại điện tử và viễn thông. Chúng thường được tích hợp vào các nền tảng nhắn tin phổ biến như Messenger, WhatsApp, Slack hoặc các nền tảng chat riêng của doanh nghiệp. Một số hệ thống có khả năng tương tác bằng giọng nói và được tích hợp vào các tổng đài tự động để cung cấp dịch vụ chăm sóc khách hàng.

- **Intelligent Personal Assistants - IPAs**

IPAs đóng vai trò như những trợ lý ảo, hỗ trợ người dùng trong các công việc hàng ngày. Ví dụ, Siri của Apple có thể thực hiện các tác vụ như gọi điện, nhắn tin, tìm nhà hàng, quản lý lịch hẹn. Tương tự có các trợ lý khác như Cortana của Microsoft, Google Assistant và Alexa của Amazon. Khi gặp yêu cầu ngoài khả năng, IPAs thường chuyển sang tìm kiếm web. IPAs có thể thực hiện nhiều tác vụ đa dạng, không giới hạn trong một lĩnh vực cụ thể, tạo cảm giác luôn sẵn sàng hỗ trợ người dùng. Để làm được điều này, IPAs thường được tích hợp vào các thiết bị cá nhân như điện thoại thông minh, máy tính, thiết bị nhà thông minh, đồng hồ thông minh và được cấp quyền truy cập vào thông tin cá nhân của người dùng như danh bạ, email, lịch và hành vi sử dụng thiết bị.

- **Chit-chat dialogue systems - CDDs**

Khác với TODs và IPAs tập trung vào hiệu quả và tốc độ hoàn thành công việc, CDDs được thiết kế để trở thành "bạn ảo" đồng hành lâu dài với người dùng. Chúng lắng nghe và thảo luận về mọi chủ đề trong cuộc sống. Mặc dù ấn tượng về khả năng, CDDs đang gặp thách thức trong việc kiểm soát nội dung khi thảo luận các chủ đề nhạy cảm như phân biệt chủng tộc, tôn giáo, chiến tranh, chính trị hoặc đưa ra nội dung không phù hợp với người dùng vị thành niên.

2.3.5. Dữ liệu và huấn luyện hệ thống Hỏi đáp tự động

Một hệ thống chatbot thông thường bao gồm hai chức năng chính: nhận dạng Intent và nhận dạng Entities. Intent giúp hệ thống hiểu ý định của người dùng trong mỗi tin nhắn, trong khi Entities giúp trích xuất thông tin cụ thể liên quan đến Intent đó. Để thực hiện hai nhiệm vụ này, cần sử dụng hai mô hình học máy riêng biệt: một để phân loại #Intents và một để phân loại @Entities. Kết hợp hai mô hình này giúp chatbot hiểu rõ hơn ngữ cảnh và yêu cầu của người dùng, từ đó cung cấp phản hồi chính xác và hiệu quả.

- **Phân loại Intent**

Để thực hiện nhiệm vụ này, cần sử dụng một mô hình phân loại chủ đề văn bản (Topic classification). Dữ liệu đầu vào là các văn bản đã được gán nhãn với các chủ đề tương ứng, ví dụ như trong trường hợp của chatbot, đó có thể là các tin nhắn từ người dùng. Trong phân loại văn bản, một kỹ thuật phổ biến là sử dụng Bag of Words để trích xuất đặc trưng. Các mô hình tiên tiến hơn có thể sử dụng Continuous Bag of Words kết hợp với deep learning. Ý tưởng cơ bản là đếm tần suất xuất hiện của các từ trong văn bản, dựa trên một từ điển chứa tất cả các từ xuất hiện trong toàn bộ tập văn bản.

Tính chất cốt lõi của xử lý ngôn ngữ tự nhiên là sử dụng các biến đổi để chuyển đổi văn bản sang số, sao cho vẫn giữ được các đặc trưng quan trọng của văn bản. Bài toán phân loại chủ đề (và phân loại Intent cho chatbot) tương tự như bài toán phân tích tâm lý khách hàng (sentiment analysis), trong việc áp dụng các mô hình học máy để hiểu và phân loại dữ liệu văn bản theo các chủ đề hoặc ý định cụ thể.

- **Phân loại Entities**

Trong xử lý ngôn ngữ tự nhiên, việc nhận diện và phân loại các thực thể (Entities) như tên người, địa danh, ngày tháng, con số và các loại thực thể khác trong văn bản được thực hiện thông qua kỹ thuật học máy gọi là Named Entity Recognition (NER) hoặc còn được gọi là Entity Extraction. Named Entity Recognition (NER) là quy trình tự động xác định và phân loại các thực thể định danh khác nhau trong văn bản thành các loại cụ thể như tên riêng, địa danh, thời gian, số lượng, v.v. NER đóng vai trò quan trọng trong việc trích xuất thông tin quan trọng từ văn bản và hỗ trợ các ứng dụng xử lý ngôn ngữ tự nhiên như chatbot, tóm tắt văn bản, trích xuất thông tin, và nhiều ứng dụng khác.

Qua việc áp dụng các mô hình học máy và kỹ thuật xử lý ngôn ngữ tự nhiên, NER giúp tự động hóa quá trình nhận dạng và phân loại thực thể, giúp cải thiện hiệu suất và chính xác của các ứng dụng xử lý ngôn ngữ tự nhiên. Những mô hình hiện đại nhất hiện nay ứng dụng Deep learning cho bài toán NER. Mô hình cho

độ chính xác cao nhất tới giờ là mô hình Deep learning kết hợp CRF mang tên Bi-LSTM-CRF.

- **Huấn luyện hệ thống Hỏi đáp tự động**

Tập huấn luyện có sự đồng nhất càng cao thì khả năng dự đoán càng chính xác.

Độ đồng nhất trong tập huấn luyện có mức độ ảnh hưởng trực tiếp đến khả năng dự đoán chính xác của mô hình. Ví dụ, khi xác định ngày tháng (@Date), định dạng thông thường như DD/MM/YYYY hoặc DD-MM-YYYY giúp làm cho việc dự đoán trở nên trực quan và dễ dàng hơn. Tuy nhiên, đối với thực thể @Term_Insurance (thời hạn bảo hiểm), việc xác định định dạng có thể phức tạp hơn nhiều. Các giá trị như "Trung hạn", "dài hạn", "ngắn hạn", "3 năm", "5 năm", "12 năm", "20 năm" có đa dạng về cả số và chữ. Ví dụ, @Term_Insurance có thể bao gồm cả token 2 từ tiếng Anh hoặc một sự kết hợp giữa số và chữ cái. Vì sự phức tạp và đa dạng của các giá trị, việc dự đoán cho @Term_Insurance trở nên khó khăn hơn nhiều so với việc xử lý các thực thể @Date đơn giản. Để cải thiện khả năng dự đoán trong trường hợp này, việc có tập dữ liệu huấn luyện đa dạng và đồng nhất là cực kỳ quan trọng. Áp dụng các kỹ thuật học máy tiên tiến và xử lý ngôn ngữ tự nhiên có thể giúp nâng cao chính xác của mô hình trong việc dự đoán các thực thể phức tạp như @Term_Insurance.

Đúng chính tả và ngữ pháp sẽ dễ dàng dự đoán câu trả lời hơn

Trong việc phân loại các thực thể (Entities), việc ghi nhận thông tin từ người dùng với chính tả chính xác đóng vai trò quan trọng vì thông thường, cấu trúc của từ được xem xét rộng rãi trong quá trình phân loại. Mô hình NER (Named Entity Recognition) dựa vào việc phát âm của từ để xác định các đặc trưng cho việc phân loại, ví dụ "giầy mới" khác với "giay moi" (giấy mời). Ngôn ngữ Nhật Bản có thể không nhạy cảm với việc sử dụng dấu và việc viết hoa, nhưng khi được xử lý bởi máy tính, vẫn sẽ phải mã hóa thành các ký tự máy tính hiểu được thông qua các bộ mã hóa như ASCII, UTF-8. Do đó cần có sự chính xác trong nhập liệu. Ngoài

ra, yếu tố ngữ cảnh (Context) cũng đóng vai trò quan trọng trong quá trình huấn luyện các thực thể. Ví dụ, vị trí của một từ trong câu cũng có thể cung cấp thông tin quan trọng. Đối với thực thể @Location, các từ như "tại", "ở", "từ", "đến" có thể là đặc trưng quan trọng để dự đoán được câu trả lời.

Synonym (từ đồng nghĩa)

Khi sử dụng Synonym trong trường hợp từ đồng nghĩa, cần phải áp dụng phương pháp này ngay từ bước tách từ (Tokenizer). Khi gặp một từ được xác định là Synonym, token sẽ được thay thế bằng một từ gốc tương ứng. Để xác định một Entity hiệu quả, việc xác định các từ đồng nghĩa của token vô cùng quan trọng.

2.3.6. Thách thức xây dựng hệ thống hỏi đáp tự động

Xây dựng chatbot là một quá trình phức tạp đòi hỏi sự kết hợp giữa khoa học máy tính, ngôn ngữ học và trải nghiệm người dùng. Dưới đây là một số thách thức phổ biến mà các nhà phát triển chatbot thường phải đối mặt:

- Hiểu Ngôn Ngữ Tự Nhiên (NLP): Một trong những thách thức lớn nhất là thiết lập hệ thống NLP đủ mạnh để hiểu và xử lý ngôn ngữ tự nhiên của người dùng. Việc xử lý từ ngữ, ngữ cảnh, ngữ pháp và cảm xúc đòi hỏi sự chính xác và linh hoạt từ phía chatbot.
- Tích Hợp Hệ Thống: Chatbot thường phải tích hợp với các hệ thống khác nhau như cơ sở dữ liệu, CRM (Quản lý mối quan hệ khách hàng), và các ứng dụng khác để cung cấp thông tin chính xác và hữu ích cho người dùng.
- Tương Tác Người-Máy: Tạo ra một tương tác tự nhiên và hấp dẫn giữa chatbot và người dùng là một thách thức khó khăn. Chatbot cần phát triển các câu trả lời linh hoạt, không gian và thân thiện để tạo ra trải nghiệm tương tác tốt.
- Bảo Mật và Quyền Riêng Tư: Bảo vệ thông tin cá nhân của người dùng và đảm bảo an toàn thông tin là một yếu tố quan trọng khi xây dựng chatbot, đặc biệt khi xử lý thông tin nhạy cảm.

- Học Máy Liên Tục: Chatbot cần có khả năng học từ dữ liệu mới và cải thiện hiệu suất theo thời gian, điều này đòi hỏi sự theo dõi và cập nhật liên tục từ phía nhà phát triển.
- Đa Nền Tảng: Chatbot cần được phát triển để hoạt động trên nhiều nền tảng khác nhau như website, ứng dụng di động, các ứng dụng tin nhắn như Facebook Messenger, WhatsApp, và các nền tảng khác.
- Định Hình Mục Tiêu: Xác định rõ mục tiêu và chức năng chính của chatbot để đảm bảo rằng nó cung cấp giá trị cho người dùng và hỗ trợ mục tiêu kinh doanh của tổ chức.

Có thể nhận thấy việc xây dựng chatbot không chỉ là việc triển khai công nghệ, mà còn đòi hỏi sự hiểu biết sâu rộng về ngôn ngữ, trải nghiệm người dùng và mục tiêu kinh doanh để tạo ra một sản phẩm hiệu quả và hấp dẫn.

2.4. KẾT LUẬN CHƯƠNG

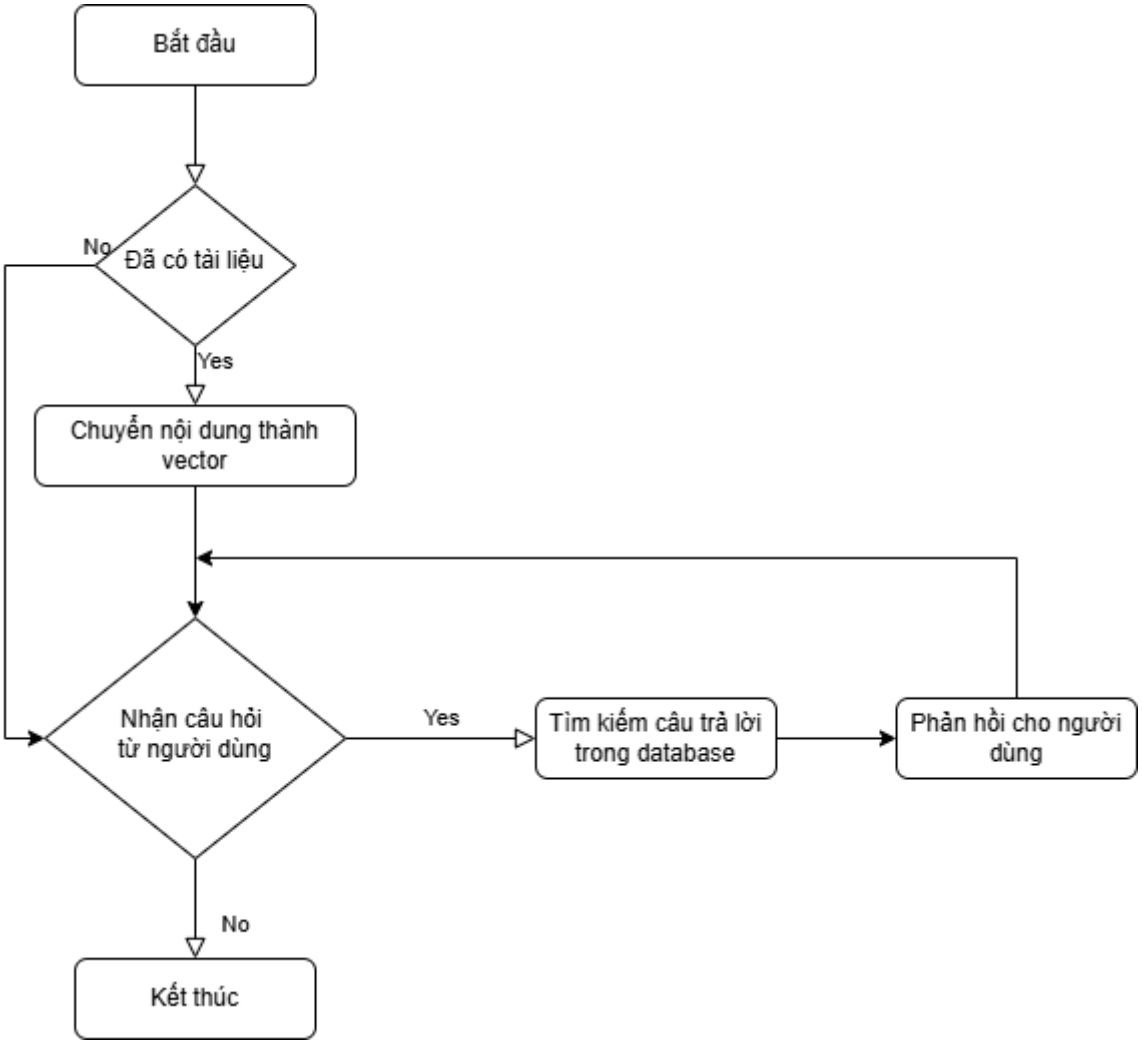
Chương 2 đã cung cấp cái nhìn toàn diện về bài toán ứng dụng RAG và LLM vào xây dựng hệ thống hỏi đáp tự động. Cải thiện hệ thống hỏi đáp dựa trên RAG framework mang lại sự nâng cao đáng kể về hiệu suất và chất lượng của việc trả lời câu hỏi tự động. Kết hợp RAG framework giúp hệ thống hiểu rõ ngữ cảnh của câu hỏi và cung cấp câu trả lời chính xác hơn. Việc áp dụng mô hình ngôn ngữ và trí tuệ nhân tạo giúp cải thiện khả năng học tập và linh hoạt của hệ thống, tạo ra trải nghiệm tương tác tốt hơn cho người dùng và nâng cao khả năng xử lý thông tin hiệu quả.

CHƯƠNG 3. XÂY DỰNG HỆ THỐNG HỎI ĐÁP TỰ ĐỘNG BẰNG MÔ HÌNH NGÔN NGỮ LỚN VÀ RAG FRAMEWORK

3.1. XÂY DỰNG HỆ THỐNG HỎI ĐÁP TỰ ĐỘNG

3.1.1. Sơ đồ khối chương trình

Hệ thống được xây dựng toàn bộ bằng ngôn ngữ lập trình python. Sơ đồ khối hoạt động được thể hiện trong hình 3.1:



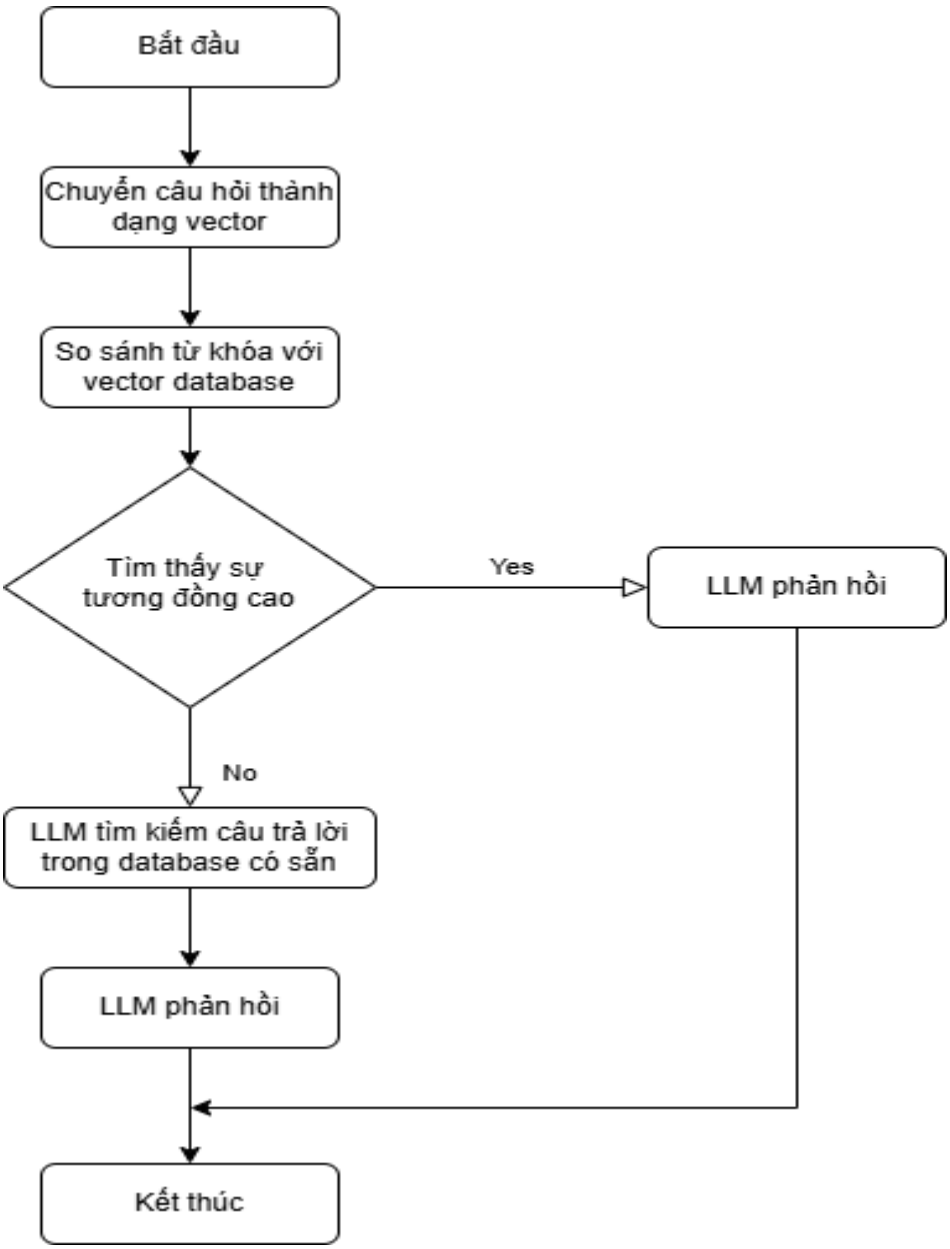
Hình 3.1: Sơ đồ khối hệ thống hỏi đáp tự động

Đầu tiên khi bắt đầu một phiên làm việc, hệ thống sẽ kiểm tra người dùng đã thực hiện thêm tài liệu PDF vào hay chưa. Nếu việc này chưa được thực hiện, sẽ có một cảnh báo tới người dùng yêu cầu Import file PDF để tiếp tục sử dụng.

Sau khi người dùng đã Import file PDF vào hệ thống, sẽ được chuyển đến bước tiếp theo.

Việc “Convert file PDF into data in Vector Database” sẽ biến những kiến thức thu được từ file pdf trở thành những vector data và gắn vào một nơi lưu trữ được gọi là Vector Database. Bước chuyển đổi này sẽ được thực hiện trong phần Back-end, người dùng chỉ cần chờ việc chuyển đổi hoàn thành là có thể bắt đầu đặt câu hỏi cho hệ thống.

Tại bước “Search for answers in the database”, hệ thống có thể được chia nhỏ như hình 3.2. Bước này bao gồm cả việc RAG framework được ứng dụng như thế nào trong một hệ thống hỏi đáp tự động. Câu hỏi sẽ được biến đổi thành dạng vector data, sau đó so sánh vector data này với tập vector trong vector database của file pdf import vào từ người dùng. Nếu có sự tương đồng cao trong hai phần này, hệ thống có thể hiểu nội dung câu hỏi có liên quan tới tài liệu PDF và phản hồi thông tin liên quan tới câu hỏi của người dùng. Nếu không tìm thấy sự tương đồng, hệ thống sẽ tìm kiếm câu trả lời trong tập dữ liệu được đào tạo sẵn. Từ đó phản hồi tới người dùng câu trả lời tương ứng.



Hình 3.2: Sơ đồ khối "Search for answers in the database"

Nếu không còn nhận được câu hỏi nào từ người dùng, phiên làm việc sẽ được kết thúc.

3.1.2. Mô hình ngôn ngữ

Trong phạm vi đề án "Cải thiện mô hình ngôn ngữ lớn bằng RAG framework và ứng dụng trong hệ thống hỏi đáp tự động", module được lựa chọn là Llama 3.2 3B. Được biết đến là mô hình cỡ nhỏ chỉ có văn bản, phù hợp với các thiết bị di động và biên giới phát hành bởi Meta, bao gồm cả các phiên bản được đào tạo trước và điều chỉnh theo hướng dẫn. Các mô hình "text-only" như

Llama 3.2 được tối ưu hóa cho các trường hợp sử dụng hội thoại đa ngôn ngữ, bao gồm các tác vụ truy xuất và tóm tắt tác nhân. Chúng hoạt động tốt hơn nhiều mô hình trò chuyện kín và mã nguồn mở có sẵn trên các tiêu chuẩn chung của ngành.

Llama 3.2 là mô hình ngôn ngữ tự động hồi quy sử dụng kiến trúc biến áp được tối ưu hóa. Các phiên bản đã điều chỉnh sử dụng tính năng tinh chỉnh có giám sát (SFT) và học tăng cường bằng phản hồi của con người (RLHF) để phù hợp với sở thích của con người về tính hữu ích và an toàn.

Bảng 3.1: Kiến trúc các phiên bản của mô hình Llama 3.2

	Training Data	Params	Input modalities	Output modalities	Context Length	GQA	Shared Embeddings	Token count	Knowledge cutoff
Llama 3.2 (text only)	A new mix of publicly available online data.	1B (1.23B)	Multilingual Text	Multilingual Text and code	128k	Yes	Yes	Up to 9T tokens	December 2023
		3B (3.21B)	Multilingual Text	Multilingual Text and code					
Llama 3.2 Quantized (text only)	A new mix of publicly available online data.	1B (1.23B)	Multilingual Text	Multilingual Text and code	8k	Yes	Yes	Up to 9T tokens	December 2023
		3B (3.21B)	Multilingual Text	Multilingual Text and code					

Yếu tố đào tạo: Meta đã sử dụng thư viện đào tạo tùy chỉnh, cụm GPU được xây dựng tùy chỉnh và cơ sở hạ tầng sản xuất để đào tạo trước. Tinh chỉnh, lượng tử hóa, chú thích và đánh giá cũng được thực hiện trên cơ sở hạ tầng sản xuất.

Sử dụng năng lượng trong đào tạo: Quá trình đào tạo đã sử dụng tổng cộng 916 nghìn giờ tính toán GPU trên phần cứng loại H100-80GB (TDP là 700W), theo bảng bên dưới. Thời gian đào tạo là tổng thời gian GPU cần thiết để đào tạo từng model và mức tiêu thụ điện năng là công suất điện cao nhất trên mỗi thiết bị GPU được sử dụng, được điều chỉnh để đạt hiệu quả sử dụng điện năng.

Đào tạo về phát thải khí nhà kính: Tổng lượng phát thải khí nhà kính theo địa điểm ước tính là 240 tấn khí CO2 cho đào tạo. Kể từ năm 2020, Meta đã duy trì lượng phát thải khí nhà kính bằng 0 trong các hoạt động toàn cầu của mình và

kết hợp 100% lượng điện sử dụng với năng lượng tái tạo; do đó, tổng lượng phát thải khí nhà kính dựa trên thị trường cho hoạt động đào tạo là 0 tấn khí CO2.

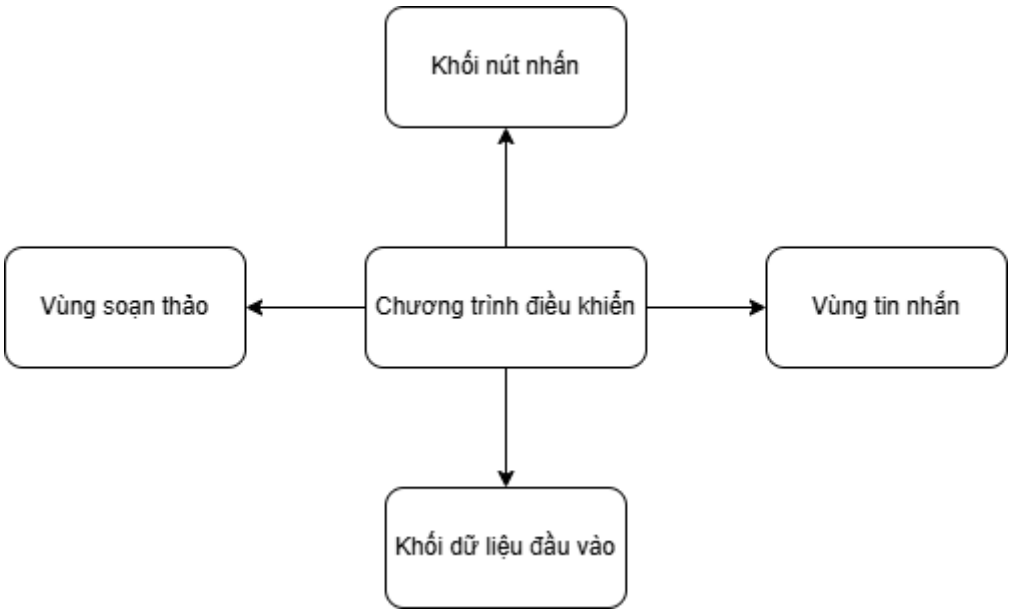
Bảng 3.2: Tài nguyên tiêu tốn khi đào tạo mô hình ngôn ngữ Llama 3.2

	Training Time (GPU hours)	Logit Generation Time (GPU Hours)	Training Power Consumption (W)	Training Location-Based Greenhouse Gas Emissions (tons CO2eq)	Training Market-Based Greenhouse Gas Emissions (tons CO2eq)
Llama 3.2 1B	370k	-	700	107	0
Llama 3.2 3B	460k	-	700	133	0
Llama 3.2 1B SpinQuant	1.7	0	700	*Negligible***	0
Llama 3.2 3B SpinQuant	2.4	0	700	*Negligible***	0
Llama 3.2 1B QLoRA	1.3k	0	700	0.381	0
Llama 3.2 3B QLoRA	1.6k	0	700	0.461	0
Total	833k	86k		240	0

Có thể thấy để đào tạo và cập nhật một mô hình ngôn ngữ lớn tiêu tốn rất nhiều thời gian và lượng tài nguyên khổng lồ. Vì vậy mặc dù mỗi ngày có rất nhiều kiến thức mới được sinh ra nhưng không thể cập nhật liên tục các mô hình này. Vì thế yêu cầu một kỹ thuật có thể giúp các mô hình ngôn ngữ lớn mở rộng phạm vi kiến thức và tìm kiếm các thông tin mới được đặt lên trên hết, và RAG chính là một trong những giải pháp tối ưu cho nhu cầu này.

3.1.3. Các thành phần của hệ thống hỏi đáp tự động

Các thành phần chính của hệ thống hỏi đáp tự động bao gồm:



Hình 3.3: Các thành phần hệ thống

- Khối nút nhấn: Khối nút nhấn có vai trò chính là liên kết các chức năng của hệ thống với chương trình điều khiển, ghi nhận yêu cầu của người dùng và gọi chức năng đó từ chương trình để thực hiện lệnh.
- Vùng soạn thảo: Là nơi ghi nhận các câu hỏi và nội dung người dùng nhập vào từ bàn phím.
- Vùng tin nhắn: Là nơi hiển thị nội dung trao đổi giữa người dùng với hệ thống hỏi đáp tự động.
- Khối dữ liệu đầu vào: Là nơi người dùng sử dụng để thực hiện nhập các tài liệu PDF vào hệ thống hỏi đáp tự động, mở rộng giới hạn, phạm vi tìm kiếm cho hệ thống.
- Chương trình điều khiển: Chương trình điều khiển được tạo thành từ một tập hợp các câu lệnh. Một chương trình điều khiển tốt là một chương trình có logic rõ ràng, tối ưu được hiệu suất và liên kết tốt với các thành phần của một chương trình. Các khối được liên kết với nhau và tạo nên một thể thống nhất, hoạt động theo logic được thiết kế và lập trình một cách chặt chẽ.

3.2. TRIỂN KHAI HỆ THỐNG HỎI ĐÁP TỰ ĐỘNG

3.2.1. Yêu cầu bài toán

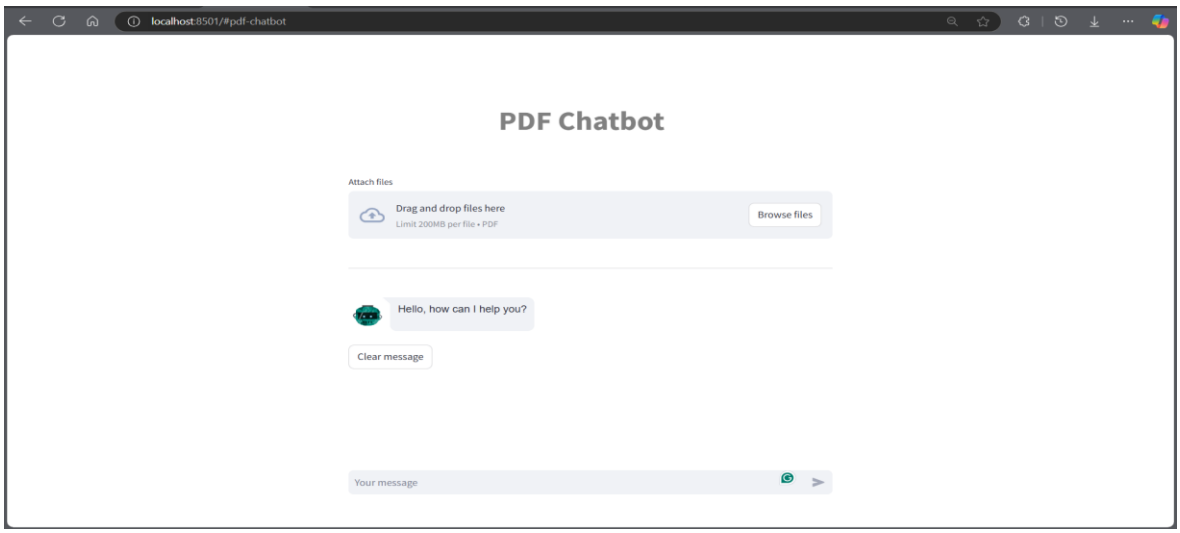
Trước khi bước vào quá trình phát triển, cần xác định rõ những yêu cầu cần thiết để đảm bảo rằng sản phẩm cuối cùng đáp ứng được nhu cầu và mong muốn của người dùng. Một hệ thống hỏi đáp tự động cần đảm bảo:

- Giao diện người dùng (UI): Dễ sử dụng và thân thiện.
- Tính tự động và độ chính xác: Khả năng tự động xử lý câu hỏi và trả lời một cách chính xác. Điều này đòi hỏi một hệ thống xử lý ngôn ngữ tự nhiên (NLP) mạnh mẽ và cập nhật.
- Tính linh hoạt và mở rộng: Có khả năng linh hoạt để mở rộng chức năng và tính năng mới theo thời gian. Hệ thống nên được thiết kế để dễ dàng mở rộng và cập nhật.

3.2.2. Giao diện hệ thống hỏi đáp tự động

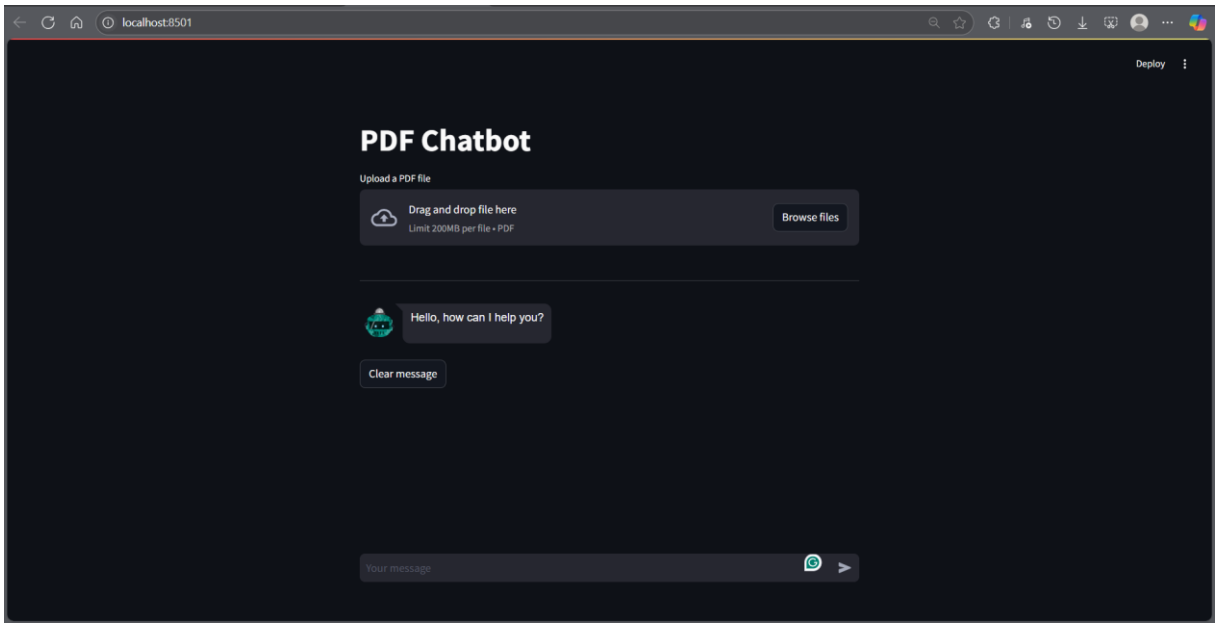
Sau thời gian tìm hiểu, nghiên cứu và thực nghiệm, hệ thống hỏi đáp tự động đã được hoàn thành theo yêu cầu.

Giao diện thực tế của hệ thống:



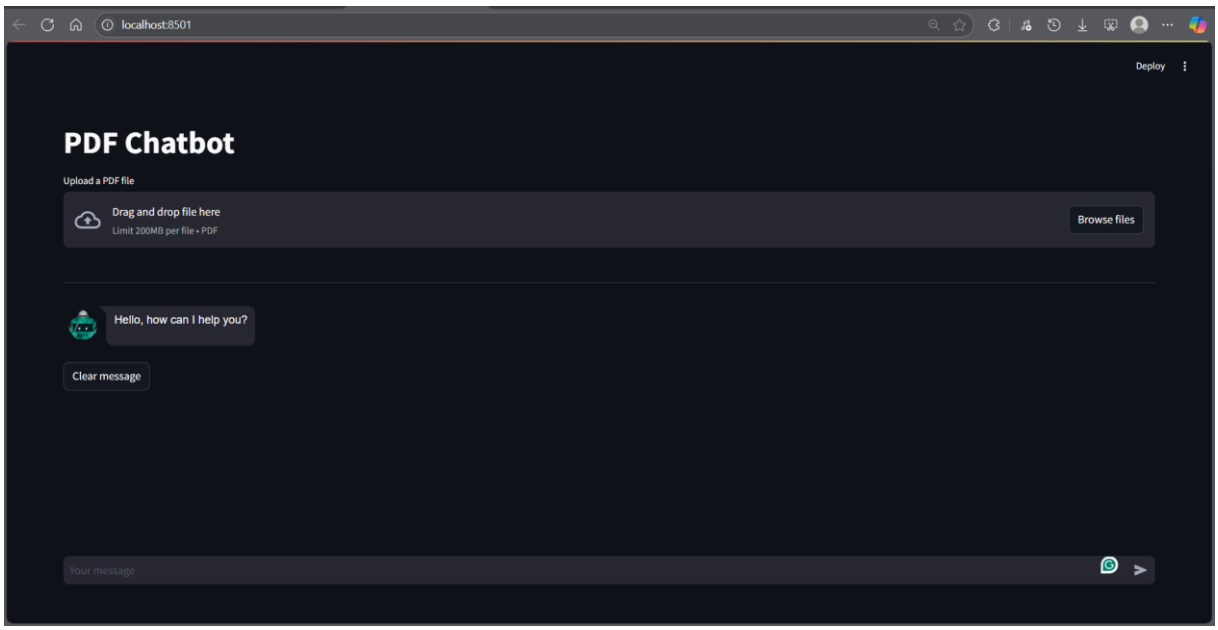
Hình 3.4: Giao diện mặc định hệ thống

Chatbot hỗ trợ chế độ tối cho người dùng để phù hợp với các hệ thống khác nhau:



Hình 3.5: Giao diện hệ thống ở chế độ tối

Ngoài ra để phù hợp với từng loại màn hình mà người dùng đang sử dụng, hệ thống cho phép người dùng co giãn tỷ lệ vùng hiển thị cho phù hợp với kích thước màn hình.

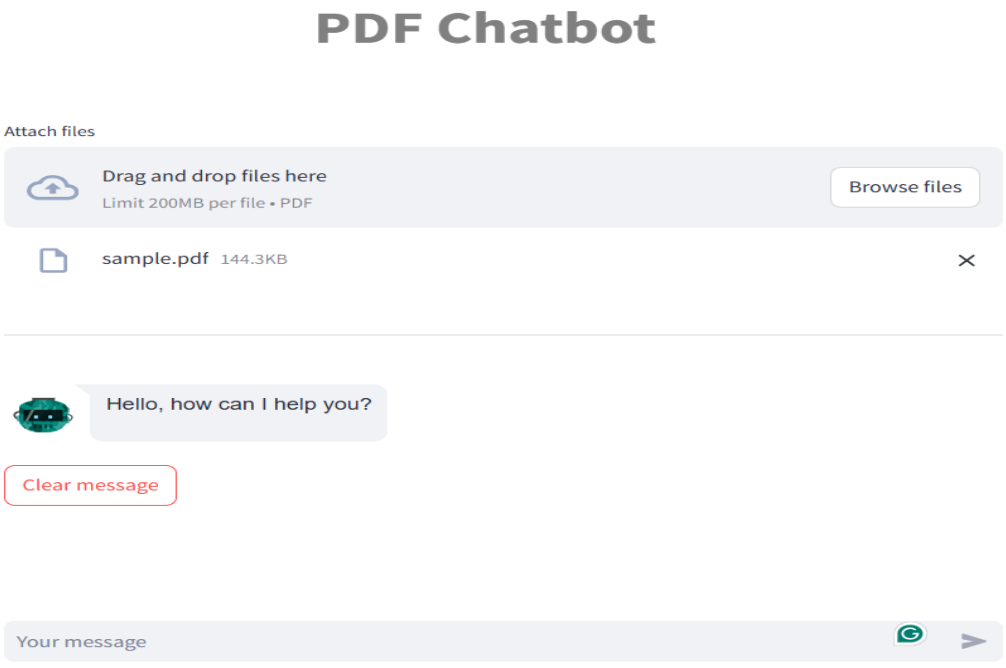


Hình 3.6: Giao diện khi mở rộng vùng hiển thị

Hệ thống sẽ yêu cầu thêm file PDF nếu người dùng chưa thực hiện bước này trước khi giao tiếp với hệ thống.

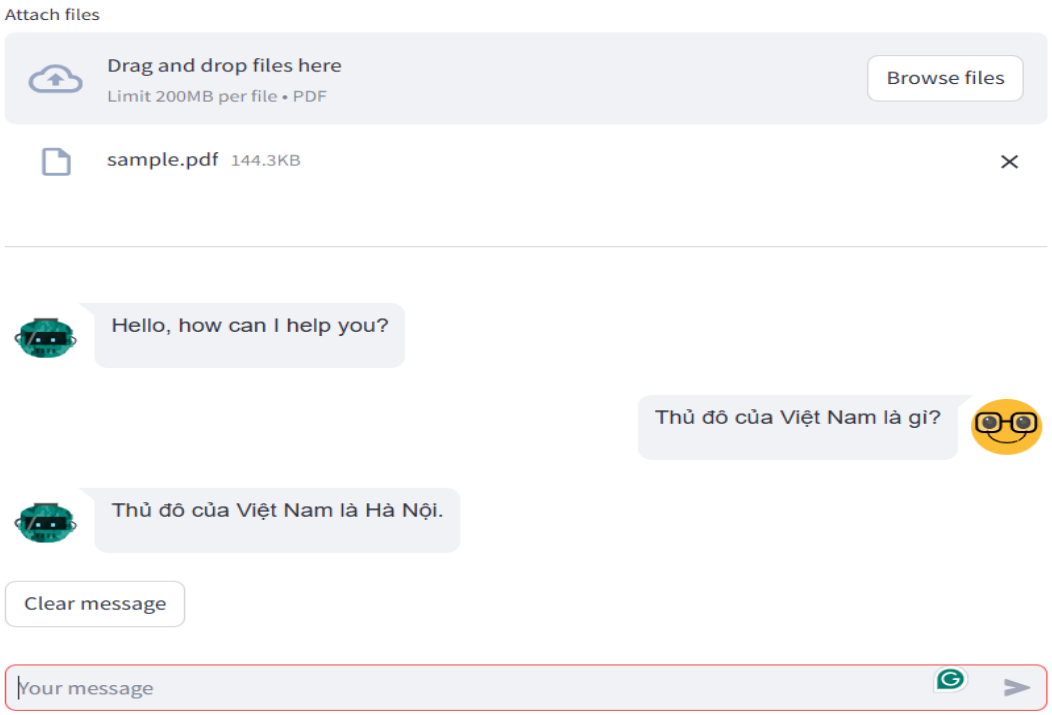
Hình 3.7: Hệ thống yêu cầu nhập file PDF

Giao diện hệ thống sau khi thực hiện thêm file pdf



Hình 3.8: Hệ thống sẵn sàng để sử dụng

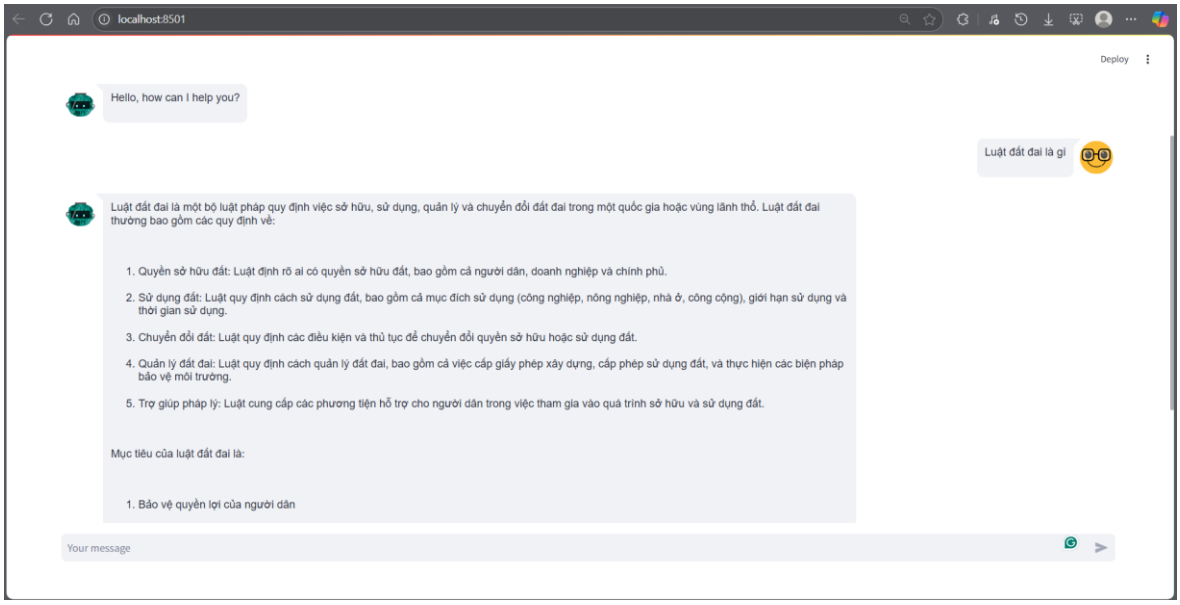
Hệ thống giao tiếp với người dùng



Hình 3.9: Tương tác với hệ thống hỏi đáp tự động

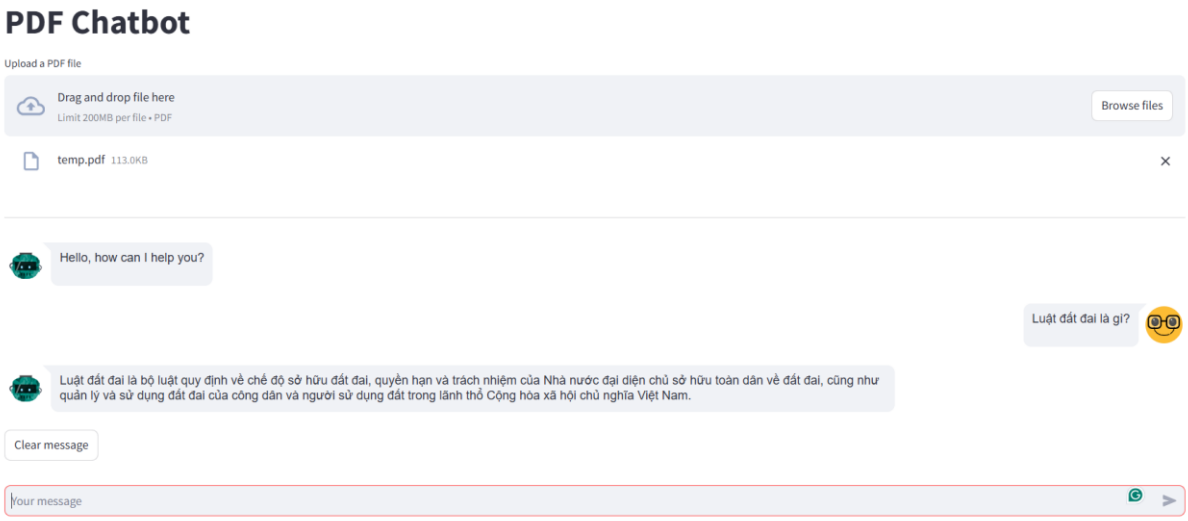
Để kiểm tra nội dung dữ liệu được bổ sung vào chatbot có thực sự được kiểm tra hay không, hệ thống sẽ được hỏi cùng một câu hỏi trước và sau khi bổ sung file dữ liệu.

Tại thời điểm thực nghiệm, mô hình được sử dụng cập nhật đến năm 2021. Điều này dẫn đến khi hỏi một câu hỏi, phản hồi mà người dùng nhận về sẽ chỉ là những thông tin chỉ cập nhật đến năm 2021. Những thông tin từ năm 2022 có thể chatbot sẽ phản hồi không chính xác. Điều này là một trong những nguyên nhân chính để sự phát triển của RAG framework là cần thiết



Hình 3.10: Chatbot phản hồi về luật đất đai khi chưa bổ sung kiến thức

Những thông tin mà chatbot phản hồi không hẳn sẽ bị đánh giá sai, nhưng có thể chỉ là những thông tin chung, hoặc nếu mô hình sử dụng không phù hợp có thể dẫn đến những trường hợp thiếu sót thông tin hoặc là những thông tin cũ mà chưa được cập nhật kịp thời, dẫn đến sai sót không đáng có. Điều này sẽ được cải thiện khi bổ sung những tài liệu, kiến thức của người dùng.



Hình 3.11: Chatbot phản hồi về luật đất đai khi đã bổ sung kiến thức

Có thể thấy sau khi bổ sung file kiến thức, khi hỏi cùng một câu hỏi nhưng chatbot sẽ phản hồi ngắn gọn, cụ thể và chính xác hơn so với trước.

3.2.3. Đánh giá hệ thống hỏi đáp tự động

Để đảm bảo chatbot vận hành hiệu quả và đưa ra thông tin đúng đắn, việc đánh giá mức độ chính xác cần được triển khai một cách toàn diện. Sau đây là các bước và phương pháp đánh giá đã được áp dụng trong đề án:

Tạo bộ dữ liệu kiểm thử: Đề án sử dụng Luật Đất đai số 31/2024/QH15 được Quốc hội nước Cộng hòa xã hội chủ nghĩa Việt Nam khóa XV, kỳ họp bất thường lần thứ năm thông qua ngày 18/01/2024.

Mục tiêu chính của quá trình kiểm thử là để đánh giá độ chính xác, tính nhất quán, và hiệu quả của chatbot trong việc trả lời các câu hỏi liên quan đến nội dung được bổ sung thêm và chatbot thông qua RAG framework.

Thực hiện đánh giá thủ công:

Bước 1: Sử dụng bộ câu hỏi 50 câu đã được chuẩn bị (50 câu) đưa vào chatbot và ghi nhận phản hồi từ hệ thống.

Bước 2: Đối chiếu kết quả của chatbot với câu trả lời mẫu trong tài liệu thu thập được và ghi nhận tỷ lệ kết quả phản hồi được chấp nhận của chatbot với từng câu hỏi.

Kết quả kiểm thử được tổng hợp tại bảng 3.3:

Bảng 3.3: Kết quả kiểm thử thủ công

Tiêu chí	Hệ thống hỏi đáp tự động
RAG framework hoạt động	Có
Giao diện thuận tiện, dễ sử dụng	Có
Khả năng xử lý file	Kích thước lớn nhất: 200MB
Thời gian phản hồi	3-4s
Tỷ lệ phản hồi được chấp nhận	~93%
Hỗ trợ ra quyết định thay người dùng	Không

Tính nhất quán trong các câu trả lời	>97%

Qua kiểm thử cho thấy, hệ thống trả lời tự động đã được xây dựng đúng theo yêu cầu đặt ra và hoàn thành tốt các công việc được yêu cầu. Tuy nhiên, có thể thấy việc đánh giá độ chính xác của một hệ thống hỏi đáp tự động là tương đối. RAG của hệ thống hoạt động tốt hơn khi bộ kiểm thử là các tài liệu quy định rõ ràng, chi tiết và cụ thể; độ chính xác sẽ tăng lên khi các câu hỏi liên quan tới định nghĩa.

Thực hiện đánh giá tự động:

Bước 1: Sử dụng các thư viện như nltk, scikit-learn, và transformers để kiểm thử chatbot với bộ 50 câu hỏi đã chuẩn bị.

Bước 2: Ghi nhận các chỉ số đầu ra tiêu chuẩn bao gồm: Precision, Recall, F1-score, và BLEU score.

Kết quả đánh giá tự động như sau:

Bảng 3.4: Kết quả kiểm thử tự động

Tiêu chí	**Kết quả**
Điểm Precision	0.89
Điểm Recall	0.92
Điểm F1-score	0.91
Điểm BLEU score	0.92

Các chỉ số này phản ánh rằng chatbot có khả năng đưa ra câu trả lời chính xác và phù hợp với ngữ cảnh trong phần lớn các tình huống. Tuy vậy, để triển khai chatbot trong thực tế, cần tiến hành kiểm thử với người dùng cuối nhằm thu thập ý kiến về mức độ hài lòng đối với các phản hồi của hệ thống. Từ đó, có thể tinh chỉnh các tham số của mô hình nhằm đạt được hiệu suất hoạt động tối ưu.

Ngoài ra, hệ thống phụ thuộc nhiều vào cấu hình phần cứng của thiết bị. Điều này có thể gây hiện tượng chậm khi phải xử lý một tài liệu kích thước lớn, tài liệu đưa ra càng nhỏ, tốc độ xử lý và phản hồi câu hỏi càng nhanh.

3.3. KẾT LUẬN CHƯƠNG

Chương 3 đã trình bày về việc triển khai hệ thống hỏi đáp tự động bao gồm giao diện và ứng dụng của hệ thống hỏi đáp tự động. Hệ thống bước đầu cho thấy khả năng hiểu câu hỏi theo ngôn ngữ tự nhiên của người dùng và đưa ra câu trả lời phù hợp. Ngoài ra đánh giá được khả năng của hệ thống hỏi đáp tự động, tạo tiền đề cho các cải tiến trong tương lai.

KẾT LUẬN VÀ KHUYẾN NGHỊ

Đề án này đã trình bày toàn diện về lý thuyết và ứng dụng của hệ thống hỏi đáp tự động và các vấn đề liên quan. Từ các khái niệm cơ bản của một hệ thống hỏi đáp tự động, có thể đưa ra một số vai trò và lý thuyết ứng dụng của lĩnh vực vào bài toán xây dựng, phục vụ cho người dùng.

Chương 1 đã cung cấp cái nhìn tổng quan về các cơ sở lý thuyết bao gồm trí tuệ nhân tạo, xử lý ngôn ngữ tự nhiên và các mô hình ngôn ngữ lớn. Nhờ vào khả năng học từ dữ liệu lớn, các mô hình này có thể tự động học và cải thiện từ kinh nghiệm, giúp nâng cao độ chính xác và hiệu quả của các hệ thống hỏi đáp. Mỗi nội dung đề là tiền để để góp phần quan trọng trong việc nâng cao độ chính xác và hiệu quả của các hệ thống hỏi đáp tự động.

Chương 2 đã đi sâu vào nghiên cứu cơ sở lý thuyết về bài toán cải thiện những thiếu sót của các hệ thống hỏi đáp tự động, từ việc giới thiệu về RAG framework và ứng dụng LLM vào hệ thống hỏi đáp. Nghiên cứu cách cải thiện khả năng học liên tục của hệ thống để nhanh chóng cập nhật kiến thức mới và cải thiện kết quả trả lời. Ta cũng đã thảo luận về một số phương pháp ứng dụng RAG và hệ thống hỏi đáp tự động một các hiệu quả. Các phương pháp này đã được chứng minh là có hiệu quả cao trong việc cải thiện hiệu năng của các hệ thống hỏi đáp, từ đó tối ưu hóa việc sử dụng nguồn lực và giảm thiểu chi phí.

Chương 3 đã đi sâu vào phần thực nghiệm và triển khai xây dựng một hệ thống hỏi đáp tự động. Dữ liệu kiểm thử được trích từ Luật Đất đai số 31/2024/QH15. Từ đó có cái nhìn tổng quan về một hệ thống hỏi đáp tự động, các tiêu chí đánh giá hệ thống và yếu tố cần lưu ý khi xây dựng hệ thống này.

Nhìn chung, việc xây dựng và triển khai một hệ thống hỏi đáp tự động trong thời kì bùng nổ công nghệ số như hiện nay có tầm quan trọng và giá trị trong việc hỗ trợ tìm kiếm câu trả lời. Các kỹ thuật tiên tiến trong hệ thống không chỉ giúp cải thiện hiệu quả kinh tế mà còn đóng góp quan trọng trong việc cải thiện các hạn chế của hệ thống. Nổi bật trong đó là RAG Framework, việc mở rộng ý tưởng

và nghiên cứu về các khía cạnh như tối ưu hóa trải nghiệm người dùng, tích hợp học sâu và học máy tăng cường, phát triển hệ thống đa ngôn ngữ và đa lĩnh vực, tích hợp tri thức và nguồn dữ liệu đa dạng, cũng như nâng cao khả năng hiểu ngữ cảnh và đối thoại tự nhiên, sẽ đóng vai trò quan trọng trong việc nâng cao khả năng ứng dụng và giá trị của hệ thống hỏi đáp tự động. Những phát triển này hứa hẹn mang lại lợi ích lớn cho cộng đồng người dùng, giúp họ dễ dàng tiếp cận thông tin và tìm kiếm câu trả lời một cách nhanh chóng và chính xác.

TÀI LIỆU THAM KHẢO

[1]. Russell, Stuart & Norvig, Peter. (2009) "Artificial Intelligence: A Modern Approach."

[2]. Alexander Choi (2024) "The LLM Effect: Are Humans Truly Using LLMs, or Are They Being Influenced By Them Instead?"

[3]. Ayman Asad Khan (2024) "Developing Retrieval Augmented Generation (RAG) based LLM Systems from PDFs: An Experience Report"

[4]. Nelson F. Liu (2023) "Lost in the Middle: How Language Models Use Long Contexts."

[5]. Yunfan Gao (2023) "Retrieval-Augmented Generation for Large Language Models: A Survey."

[6]. Khan, Rashid & Das, Anik (2017) "Build Better Chatbots"

[7]. Sarah Saad Alanaz (2021) "Question Answering Systems: A Systematic Literature Review."

[8]. Luật Đất đai số 31/2024/QH15

NgoPhucLuong - CH_HTTT_K13.1 - ĐATN.pdf

BÁO CÁO ĐỘC SÁNG

21%	25%	15%	7%
CHỈ SỐ TƯƠNG ĐỒNG	NGUỒN INTERNET	ẤN PHẨM XUẤT BẢN	BÀI CỦA HỌC SINH

NGUỒN CHÍNH

1	viblo.asia Nguồn Internet	4%
2	www.ctu.edu.vn Nguồn Internet	3%
3	funix.edu.vn Nguồn Internet	2%
4	Submitted to Hanoi Metropolitan University Bài của Học sinh	1%
5	vtitech.vn Nguồn Internet	1%
6	aws.amazon.com Nguồn Internet	1%
7	Submitted to Da Nang University of Economics Bài của Học sinh	1%
8	tailieu.vn Nguồn Internet	1%
9	quanly.hpu2.edu.vn Nguồn Internet	1%
10	www.slideshare.net Nguồn Internet	1%
11	Submitted to The Olympia School Bài của Học sinh	1%
12	anchor.fm Nguồn Internet	1%