ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

IR-BASED CHATBOT SYSTEM WITH CUSTOM DOCUMENT AND IMAGE DATASET

Lớp: CS336.O11

Môn: Truy vấn thông tin đa phương tiện

Giáo viên hướng dẫn: ThS. Đỗ Văn Tiến

Thành viên nhóm:

20521609 - Nguyễn Hoàng Minh
20521998 - Nguyễn Thiện Thuật

TP. HỒ CHÍ MINH – 1/2024

LỜI MỞ ĐẦU

Trong thời đại của trí tuệ nhân tạo, chatbot là một ứng dụng phổ biến và hữu ích, cho phép người dùng giao tiếp với các hệ thống thông tin bằng ngôn ngữ tự nhiên. Chatbot có thể được sử dụng cho nhiều mục đích khác nhau, như hỗ trợ khách hàng, giáo dục, giải trí, y tế, du lịch, v.v. Tuy nhiên, để xây dựng một chatbot hiệu quả và thân thiện, các nhà phát triển cần có một hệ thống truy xuất thông tin (IR) mạnh mẽ, có thể tìm kiếm và trả lời các câu hỏi của người dùng dựa trên các nguồn dữ liệu khác nhau, bao gồm văn bản và hình ảnh. Đây là một thách thức lớn, vì các nguồn dữ liệu thường có kích thước lớn, đa dạng, không cấu trúc và chứa nhiều nội dung không liên quan.

Mục tiêu của đề tài là xây dựng một hệ thống chatbot dựa trên truy vấn thông tin (IR), có thể truy xuất và trả lời các câu hỏi của người dùng từ một tập dữ liệu tùy chỉnh, bao gồm các tài liệu văn bản và hình ảnh. Để thực hiện đề tài này, chúng tôi đã xây dựng một hệ thống chatbot dựa trên truy vấn thông tin với 4 khả năng chính: chat với large language model, chat với tài liệu, chat tìm kiếm ảnh trong bộ dữ liệu và chat tìm kiếm ảnh tương đồng. Hệ thống sử dụng một số mô hình học sâu để biểu diễn và so sánh các câu hỏi và các đoạn văn bản hoặc hình ảnh, sau đó sử dụng các thuật toán tính toán sự tương đồng của những câu truy vấn với bộ dữ liệu để xếp hạng và lựa chọn đầu ra phù hợp nhất. Hơn nữa, hệ thống cũng có khả năng sinh ra các câu trả lời tự nhiên, dựa trên các đoạn văn bản được truy xuất.

Bên cạnh đó, đề tài cũng triển khai demo cho hệ thống đề xuất sử dụng một bộ dữ liệu được thu thập với 10 tập văn bản với 10 nhãn chứa các hình ảnh, kết hợp với thư viện streamlit nhằm xây dựng giao diện trực quan để người dùng tương tác với hệ thống.

Chúng tôi tin rằng với ý tưởng này sẽ giới thiệu một hệ thống chatbot có thể được sử dụng để giải quyết những hạn chế còn tồn tại và hơn thế nữa là có thể tối ưu và mở rộng khả năng của nó.

PHÂN CÔNG VIỆC VÀ ĐÁNH GIÁ THÀNH VIÊN

Nhóm công việc	**Công việc cụ thể**	**Phân công**	
		Nguyễn Hoàng Minh 20521609	**Nguyễn Thiện Thuật 20521998**
Xây dựng và triển khai ý tưởng	Tìm hiểu các bài toán trong truy xuất thông tin đa phương tiện *(truy vấn tài liệu, truy vấn ảnh, tìm ảnh tương đồng)*	x	x
	Xây dựng kiến trúc chung cho toàn bộ hệ thống	x	-
	Lập bảng phân công việc và đánh giá	x	-
Code thực nghiệm	Xây dựng hệ thống cho bài toán chat với LLMs	x	-
	Xây dựng hệ thống cho bài toán chat với documents dựa trên truy vấn bằng text và LLMs	x	-
	Xây dựng hệ thống cho bài toán truy xuất ảnh từ images dataset dựa trên truy vấn bằng text	-	x
	Xây dựng hệ thống cho bài toán tìm kiếm ảnh tương đồng từ images dataset dựa trên truy vấn bằng image	-	x
Bộ dữ liệu	Xác định nguồn dữ liệu demo	x	-
	Thu thập dữ liệu	-	x
Code triển khai hệ thống	Đóng gói các hệ thống thành phần	x	-
	Xử lý và xác định các truy vấn	-	x
	Xây dựng web demo bằng streamlit	x	-
	Kiểm tra và sửa lỗi hệ thống	x	x
Slide	Viết nội dung slide	40%	60%
	Trình bày slide	-	x
Báo cáo	Viết nội dung báo cáo	40%	60%
	Trình bày báo cáo	-	x
Đánh giá chung khả năng hoàn thành nhiệm vụ		**100%**	**100%**

*Ghi chú: ‘x’: tham gia, ‘-’: không tham gia

MỤC LỤC

1. GIỚI THIỆU

1.1. Đặt vấn đề

Trong những năm gần đây, việc xuất hiện của Trí tuệ nhân tạo - Artificial Intelligence (AI) đã tạo ra một cuộc cách mạng công nghệ 4.0. AI xuất hiện ở khắp mọi nơi, mọi lĩnh vực và đã giúp cuộc sống hiện đại trở nên dễ dàng hơn.

Cụ thể, Chatbot là một ứng dụng trí tuệ nhân tạo cho phép người dùng tương tác với máy móc bằng ngôn ngữ tự nhiên. Chatbot có thể được sử dụng cho nhiều mục đích khác nhau, như hỗ trợ khách hàng, giáo dục, giải trí, y tế, du lịch, v.v. Để xây dựng một chatbot hiệu quả, các nhà phát triển cần giải quyết một số thách thức, trong đó có việc truy xuất thông tin. Chatbot cần có khả năng tìm kiếm và trả lời các câu hỏi của người dùng dựa trên các nguồn dữ liệu khác nhau, bao gồm văn bản và hình ảnh.

Các IR-based chatbot hiện nay thường phải huấn luyện lại hoặc phải xây dựng hệ thống mới khi có dữ liệu mới. Điều này làm tăng chi phí, thời gian và công sức của các nhà phát triển, cũng như yêu cầu họ phải có chuyên môn cao về các mô hình và thuật toán học máy. Ngoài ra, các hệ thống chatbot hiện nay thường chỉ tích hợp đơn lẻ một tác vụ, như trợ lý ảo, tìm kiếm ảnh bằng văn bản hoặc tìm kiếm ảnh tương đồng. Điều này làm giảm khả năng đa dạng và đáp ứng hết nhu cầu của người dùng, cũng như làm hạn chế sự tương tác và hấp dẫn của chatbot.

1.2. Mục tiêu và thách thức của đề tài

1.2.1. Mục tiêu

Mục tiêu của đề tài này là xây dựng một hệ thống chatbot vừa tích hợp đa dạng các bài toán về truy xuất thông tin vừa có thể chat để tìm kiếm thông tin ở bất kỳ bộ dữ liệu nào mà người dùng mong muốn. Hệ thống chatbot này sẽ có khả năng truy xuất và trả lời các câu hỏi của người dùng từ một tập dữ liệu tùy chỉnh, bao gồm các tài liệu văn bản và hình ảnh, mà không cần phải huấn luyện lại hoặc xây dựng lại hệ thống. Hệ thống chatbot này cũng sẽ có khả năng sinh ra các câu trả lời tự nhiên, dựa trên các đoạn văn bản hoặc hình ảnh được truy xuất. Hệ thống chatbot này sẽ mang lại nhiều lợi ích cho người dùng, như tiết kiệm thời gian, nâng cao trải nghiệm, khám phá nhiều thông tin hơn và tăng cường sự tương tác và hài lòng.

1.2.2. Thách thức

Trong phần này, chúng tôi sẽ thảo luận về các kết quả, vấn đề và thách thức của đề tài về xây dựng một hệ thống chatbot có thể truy xuất thông tin từ một tập dữ liệu tùy chỉnh, bao gồm văn bản và hình ảnh.

Thứ nhất, thách thức về tài nguyên máy tính là một trong những thách thức lớn nhất trong việc xây dựng hệ thống chatbot dựa trên truy xuất thông tin. Các hệ thống chatbot cần xử lý một lượng lớn dữ liệu, bao gồm cả văn bản và hình ảnh. Điều này đòi hỏi hệ thống phải có tài nguyên máy tính mạnh mẽ, bao gồm bộ nhớ, CPU và GPU.

Thứ hai, thách thức về dữ liệu. Để xây dựng một hệ thống chatbot hiệu quả, cần có một bộ dữ liệu lớn và đa dạng. Bộ dữ liệu này bao gồm cả văn bản và hình ảnh. Tuy nhiên, việc thu thập bộ dữ liệu này là một thách thức lớn khi tài nguyên máy tính không cho phép để ứng dụng trên bộ dữ liệu đa dạng và kích thước lớn.

Thứ ba, thách thức về hiệu năng. Để có thể đáp ứng nhu cầu của người dùng, hệ thống chatbot cần có hiệu năng cao. Nhưng để có được điều này cần phải có bộ dữ liệu đủ lớn và đa dạng.

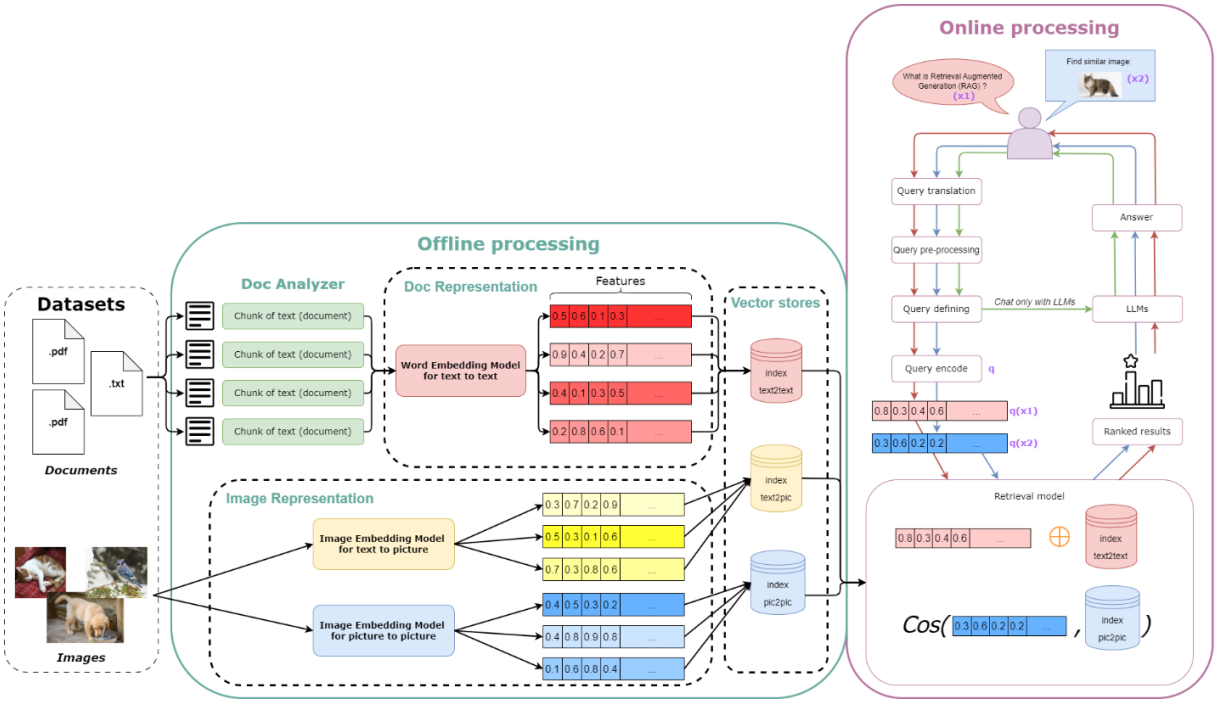
1.3. Tóm tắt các công việc thực hiện

Để thực hiện đề tài, chúng tôi đã lên kế hoạch và thực hiện những công việc sau:

- Tìm hiểu và lựa chọn những thư viện phù hợp cho việc xây dựng hệ thống chatbot dựa trên truy vấn thông tin. Chúng tôi đã sử dụng các thư viện như PyTorch, Transformers, Streamlit, v.v.
- Xây dựng hệ thống truy xuất thông tin cho 3 bài toán: truy vấn thông tin từ dữ liệu văn bản, truy xuất ảnh và tìm kiếm ảnh tương đồng từ dữ liệu hình ảnh.
- Đề xuất một hệ thống chatbot có thể chat với LLMs và tìm kiếm thông tin từ bất kỳ dữ liệu văn bản và hình ảnh nào của người dùng. Chúng tôi đã sử dụng một mô hình học sâu là GPT-4 để sinh ra các câu trả lời tự nhiên, dựa trên các đoạn văn bản hoặc hình ảnh được truy xuất.
- Thu thập dữ liệu và triển khai web chatbot demo bằng streamlit từ hệ thống đã đề xuất. Chúng tôi đã sử dụng một tập dữ liệu tùy chỉnh, bao gồm 10 tài liệu văn bản và 250 hình ảnh.

2. HỆ THỐNG IR-BASED CHATBOT

2.1. Hướng tiếp cận



Hình 2.1 Tổng quan kiến trúc hệ thống IR-based Chatbot

Trong báo cáo này, chúng tôi đề xuất xây dựng một hệ thống Chatbot dựa trên truy vấn thông tin có khả năng truy vấn thông tin từ bộ dữ liệu bất kỳ bao gồm cả hình ảnh và văn bản làm câu trả lời cho Chatbot. Bên cạnh đó, hệ thống cũng kết hợp với các mô hình ngôn ngữ lớn nhằm sinh ra các câu trả lời tự nhiên dựa trên các thông tin từ bộ dữ liệu. Hệ thống này đồng thời cũng cung cấp một trang web để người dùng có thể tùy chỉnh và chat với bộ dữ liệu mà họ mong muốn.

2.1.1. Tổng quan hệ thống IR-based Chatbot

Kiến trúc hệ thống có 2 thành phần chính Offline processing làm nhiệm vụ xử lý bộ dữ liệu và Online processing làm nhiệm vụ tương tác với người dùng để trả lời những thông tin mà họ mong muốn dựa trên kết quả thông tin truy vấn được từ bộ dữ liệu. Chúng tôi chia thành 2 thành phần nhằm mục đích tiết kiệm thời gian truy vấn của người dùng khi tương tác, vì dữ liệu đã được tiền xử lý và lưu trữ trước ở giai đoạn Offline processing nên trong quá trình truy vấn, hệ thống sẽ không phải xử lý lại. Hình 2.1 mô tả tổng quan hệ thống của chúng tôi, chia làm 4 tác vụ chính: hỏi đáp về bộ dữ liệu, tìm kiếm ảnh từ bộ dữ liệu, tìm kiếm ảnh tương đồng và hỏi đáp với mô hình ngôn ngữ lớn.

Đầu vào: Bộ dữ liệu (gồm hình ảnh và tài liệu) và một câu truy vấn (text hoặc hình ảnh).

Pre-processing model: Xử lý bộ dữ liệu và lưu đặc trưng dữ liệu vào kho lưu trữ.

Main model: Tiền xử lý và xác định câu truy vấn, truy vấn thông tin từ kho lưu trữ.

Đầu ra: Câu trả lời (hình ảnh hoặc văn bản) cho câu truy vấn.

2.1.2. Phương pháp xây dựng hệ thống

Giải pháp của chúng tôi là tiến hành thực nghiệm các tính năng mới ở bên ngoài hệ thống, sau khi các tính năng này hoạt động ổn với bộ dữ liệu thử nghiệm thì code sẽ được đóng gói thành các thư viện và tích hợp nó vào hệ thống chung. Phương pháp thực hiện được chia thành 2 phần chính:

Phần 1 – *Xây dựng tính năng cho bài toán truy xuất thông tin*

Phần này tập trung xây dựng riêng biệt các tính năng cho hệ thống chatbot, trong báo cáo này chúng tôi thực hiện xây dựng 3 bài toán chính: truy xuất tài liệu, truy xuất ảnh và tìm kiếm ảnh tương đồng. Với từng loại bài toán, chúng tôi thực hiện tìm hiểu sử dụng các mô hình pre-train và công nghệ lưu trữ open-source nhằm kế thừa độ hiệu quả của chúng mà không cần phải huấn luyện và đánh giá lại mô hình, giúp tiết kiệm thời gian và đồng thời độ hiệu quả của hệ thống sẽ tốt hơn khi huấn luyện từ đầu.

Phần 2 – *Tích hợp tính năng vào hệ thống chung*

Phần này sử dụng các tính năng thông qua các thư viện đã được xây dựng ở phần 1 làm cơ sở để xử lý và xác định các truy vấn của người dùng nhằm đưa truy vấn đó vào bài toán đã được xây dựng. Các kết quả tìm kiếm ảnh sẽ được sắp xếp theo mức độ tương đồng với câu truy vấn từ cao đến thấp. Đối với các kết quả truy xuất được ở dạng văn bản, chúng tôi thực hiện thử nghiệm thêm các giải pháp liên kết với mô hình ngôn ngữ nhằm đưa ra những câu trả lời cho người dùng một cách tự nhiên hơn.

2.2. Offline processing

Phần này làm nhiệm vụ xử lý bộ dữ liệu từ phía người dùng, chúng tôi sẽ giải thích cụ thể từng thành phần của hệ thống Offline processing cho 3 bài toán: truy xuất tài liệu, truy xuất ảnh và tìm kiếm ảnh tương đồng. Cách thức xử lý và lưu trữ các đặc trưng của dữ liệu vào kho lưu trữ cho từng thành phần.

2.2.1. Các framework nền tảng

Trong báo cáo này, chúng tôi sẽ sử dụng 4 framework để xây dựng hệ thống IR-based Chatbot, đó là: CLIP, Resnet50, Faiss và SQLite. Các framework nền tảng là các phương tiện hỗ trợ các lập trình viên trong quá trình triển khai, thiết lập các sản phẩm website, ứng dụng di động hay các ứng dụng trí tuệ nhân tạo. Các framework nền tảng thường cung cấp các tính năng, công cụ và thư viện sẵn có để giúp cho việc phát triển phần mềm trở nên nhanh chóng và hiệu quả hơn. Ngoài ra, chúng tôi sử dụng thêm thư viện **LangChain**[1] cho việc xây dựng chat với LLMs dựa trên Truy vấn thông tin (IR).

CLIP[2] [1] là viết tắt của Contrastive Language-Image Pre-training, là một framework nền tảng cho việc học sâu về ngôn ngữ và hình ảnh. CLIP được phát triển bởi OpenAI, một tổ chức nghiên cứu trí tuệ nhân tạo phi lợi nhuận. CLIP cho phép huấn luyện một mô hình học sâu có thể hiểu được cả ngôn ngữ và hình ảnh, và có thể thực hiện các nhiệm vụ như phân loại hình ảnh, tìm kiếm hình ảnh, sinh hình ảnh từ văn bản và ngược lại. CLIP có thể được sử dụng để tạo ra các ứng dụng trí tuệ nhân tạo sáng tạo và hữu ích, như DALL-E hay CLIPDraw.

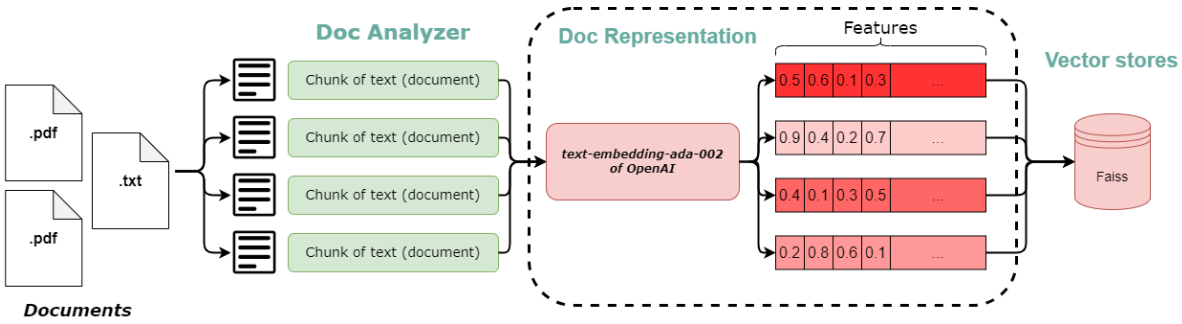
Resnet50 [2] là một framework nền tảng cho việc học sâu về hình ảnh, đặc biệt là cho việc phát hiện và nhận dạng đối tượng. resnet50 là một kiến trúc mạng nơ-ron tích chập sâu, có 50 tầng, được huấn luyện trên bộ dữ liệu ImageNet, một bộ dữ liệu lớn gồm hơn 14 triệu hình ảnh thuộc hơn 20 nghìn lớp. resnet50 có thể nhận biết được hơn 1000 loại đối tượng khác nhau, từ con người, động vật, đến các vật thể trong cuộc sống. resnet50 có thể được sử dụng để tạo ra các ứng dụng học sâu liên quan đến hình ảnh, như nhận diện khuôn mặt, phân đoạn ảnh, phát hiện biển báo giao thông và nhiều hơn nữa.

[1] https://www.langchain.com
[2] https://github.com/mlfoundations/open_clip

Faiss[3] [3] là một framework nền tảng cho việc tìm kiếm và phân cụm các vector đa chiều. faiss được phát triển bởi Facebook AI Research, một bộ phận nghiên cứu trí tuệ nhân tạo của Facebook. faiss cho phép xử lý nhanh chóng và hiệu quả các bài toán tìm kiếm gần nhất hàng hàng (nearest neighbor search) và phân cụm k-means trên các vector đa chiều lớn. faiss có thể được sử dụng để tạo ra các ứng dụng trí tuệ nhân tạo liên quan đến ngôn ngữ, hình ảnh, âm thanh, video và nhiều hơn nữa.

SQLite[4] là một framework nền tảng cho việc quản lý cơ sở dữ liệu quan hệ. SQLite là một thư viện phần mềm nhỏ gọn, độc lập, hiệu năng cao và đáng tin cậy, cho phép tạo ra và truy vấn cơ sở dữ liệu quan hệ mà không cần đến một máy chủ cơ sở dữ liệu riêng biệt. SQLite có thể được sử dụng để tạo ra các ứng dụng cần lưu trữ và xử lý dữ liệu cục bộ, như các ứng dụng di động, máy tính để bàn, thiết bị nhúng và nhiều hơn nữa.

2.2.2. Hệ thống truy xuất tài liệu



Hình 2.2 Quy trình xử lý offline hệ thống truy xuất tài liệu

Từ kết quả của nghiên cứu trong bài báo Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks [4], đã chứng minh rằng phương pháp Retrieval-Augmented Generation (RAG) có thể giúp cho các mô hình ngôn ngữ lớn (LLMs) trả lời chính xác hơn dựa vào truy vấn được trích xuất từ dataset làm ngữ cảnh cho LLMs. Chúng tôi sử dụng RAG cho bài toán chat với tài liệu như một phương pháp giúp cho LLM có thể hiểu được dữ liệu mà người dùng đã có, giúp các câu trả lời gần với thông tin được tìm kiếm.

[3] https://github.com/facebookresearch/faiss
[4] https://www.sqlite.org/index.html

Hệ thống truy xuất tài liệu của chúng tôi có 2 phần:

- ***(1) Indexing***: pipeline để nhận dữ liệu từ một nguồn và lập chỉ mục (indexing) cho nó.
- ***(2) Retrieval and generation****:* là chuỗi RAG có nhiệm vụ nhận truy vấn của người dùng trực tuyến và truy xuất dữ liệu liên quan từ indexing, sau đó chuyển dữ liệu đó đến mô hình ngôn ngữ để tạo ra câu trả lời tự nhiên cho người dùng.

Các thành phần của offline processing (Indexing):

Doc Analyzer Làm nhiệm vụ load tài liệu (document) *d* của người dùng và sau đó tách văn bản tài liệu lớn thành các phần nhỏ hơn (chunks). Điều này giúp ích cho cả hai việc lập chỉ mục cho bộ dữ liệu (Indexing) và đưa nó vào mô hình ngôn ngữ, vì các tập tài liệu quá lớn sẽ gây khó khăn trong việc tìm kiếm và sẽ không vừa input context của các mô hình. Trong hệ thống này, chúng tôi xây dựng xử lý các tài liệu có định dạng file là .pdf hoặc .txt. Các tham số được cài đặt trong hệ thống của chúng tôi như sau: Phương pháp phân chia chunks dựa trên ký tự (CharacterTextSplitter); separator="\n": mỗi phần sẽ được phân tách bằng dòng mới; chunk_size=2000: độ dài của mỗi phần (chunk) là 2000 ký tự; chunk_overlap=200: các chunk liền kề sẽ chia sẻ 200 ký tự cuối cùng.

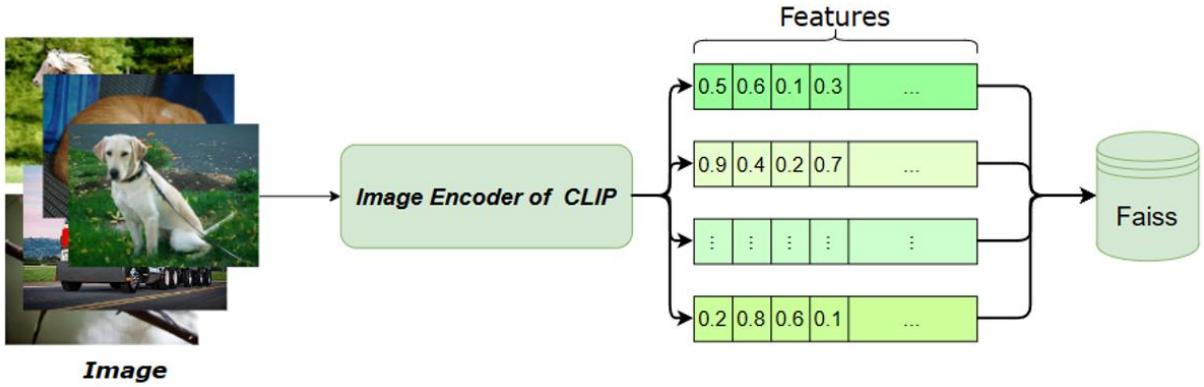
Doc representation Dùng để biểu diễn các chunk dưới dạng các vector đặc trưng nhằm phục vụ cho việc truy vấn [5], về nguyên tắc, Doc encoder *d* và Query encoder *q* có thể được triển khai bởi bất kỳ mạng thần kinh nào, nhưng trong hệ thống của chúng tôi, chúng tôi sử dụng "text-embedding-ada-002" *(API của OpenAI)* như encoder chính thức của hệ thống. Chúng tôi đưa các chunk vào mô hình embedding và sử dụng kết quả đầu ra như là vector embedding của chunk h_c. Do đó, *M* chunks sẽ được encode lần lượt là $H_c = [h_{c1}, h_{c2}, ..., h_{cM}]$.

Ngoài ra, chúng tôi cũng thử nghiệm các pre-train khác như Sentence-BERT (SBERT) [6], INSTRUCTOR [7], nhưng các phương pháp này hoạt động kém hiệu quả hơn phương pháp được sử dụng trong hệ thống.

Vectorstore Dùng để trữ, truy xuất và xử lý các vector biểu diễn trong không gian vector. Chúng tôi sử dụng kho lưu trữ này nhằm mục đích lưu trữ các chunk embedding H_c vừa được trích xuất ở thành phần trước. Giải pháp này giúp cho việc truy vấn trở nên

nhanh chóng *(vì Vectorstore được thiết kế riêng cho việc lưu trữ và truy xuất các vector)* và hệ thống sẽ không phải xử lý lại dữ liệu khi thực hiện truy vấn. Trong hệ thống của chúng tôi, chúng tôi lập chỉ mục bằng cách sử dụng FAISS open-source library [3]. FAISS rất hiệu quả để tìm kiếm sự tương đồng và phân cụm các vectơ, nó có thể dễ dàng áp dụng cho hàng tỷ vectơ mà không làm ảnh hưởng đến tốc độ truy xuất.

2.2.3. Hệ thống truy xuất hình ảnh



Hình 2.3 Quy trình xử lý offline hệ thống truy xuất hình ảnh

Để hỗ trợ truy xuất hình ảnh, chúng tôi đã tích hợp phương pháp CLIP (Contrastive Language-Image Pre-Training), chúng tôi đã mở rộng hệ thống truy xuất của mình để hỗ trợ truy xuất hình ảnh. Phương pháp CLIP [1] sử dụng vision encoder để xử lý hình ảnh và mô hình đã được pretrained trên dữ liệu lớn để hiểu các đặc trưng từ hình ảnh một cách hiệu quả.

Trong hệ thống đã được xây dựng và phát triển, chúng tôi tiến hành sử dụng mô hình VIT-L14-Datacomp1B trong CLIP làm vision encoder.

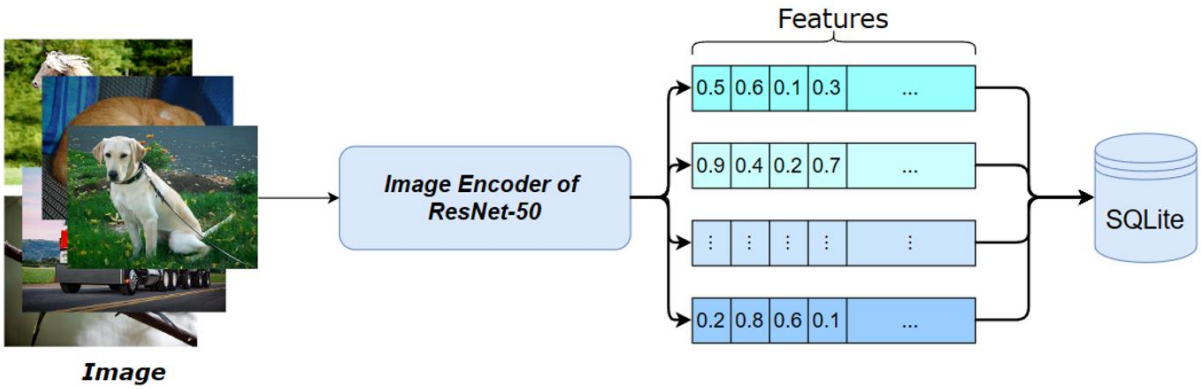
- VIT-L14 [8] là viết tắt của "Vision Transformer - Large model with 14 layers". Đây là một kiến trúc mô hình Transformer được sử dụng trong việc xử lý hình ảnh. Mô hình này được huấn luyện trên dữ liệu lớn để hiểu các đặc trưng từ hình ảnh một cách hiệu quả.
- Datacomp1B [9] là một tập dữ liệu lớn được sử dụng để huấn luyện mô hình VIT-L14. Tập dữ liệu này bao gồm hàng triệu hình ảnh thuộc nhiều phân loại khác nhau. Việc huấn luyện trên tập dữ liệu lớn giúp mô hình VIT-L14 học được các đặc trưng phổ quát từ hình ảnh và có khả năng biểu diễn chúng một cách chính xác.

- Với mô hình VIT-L14, các hình ảnh được xử lý bằng cách chia nhỏ thành các khối nhỏ hơn gọi là patch. Mỗi patch được biểu diễn bằng một vectơ đặc trưng. Sau đó, các vectơ đặc trưng này được đưa vào mạng Transformer để tạo ra biểu diễn tổng thể cho hình ảnh. Mô hình VIT-L14-Datacomp1B được huấn luyện với tham số nhỏ và độ chính xác cao. Điều này đảm bảo rằng mô hình có khả năng biểu diễn các đặc trưng từ hình ảnh một cách chi tiết và chính xác. Việc sử dụng mô hình này trong CLIP giúp hệ thống truy xuất hình ảnh của chúng tôi hiểu và tìm kiếm thông tin từ các hình ảnh một cách hiệu quả.

Hệ thống của chúng tôi hỗ trợ các định dạng file dữ liệu như .png và .jpg để xử lý hình ảnh. Bên cạnh đó, chúng tôi sử dụng Faiss làm vectorstore để lưu trữ, truy xuất và xử lý các vectơ biểu diễn trong không gian vector. Faiss là một thư viện mã nguồn mở hiệu quả cho việc tìm kiếm sự tương đồng và phân cụm các vectơ, đồng thời cho phép chúng tôi xử lý hàng tỷ vectơ một cách nhanh chóng.

Với việc mở rộng hệ thống truy xuất của chúng tôi sang truy xuất hình ảnh, chúng tôi mong muốn mang lại trải nghiệm tốt hơn cho người dùng và giúp họ tìm kiếm thông tin liên quan đến hình ảnh một cách hiệu quả.

2.2.4. Hệ thống tìm kiếm hình ảnh tương đồng



Hình 2.4 Quy trình xử lý offline hệ thống tìm kiếm hình ảnh tương đồng

Dựa vào nghiên cứu về ResNet-50 được giới thiệu bởi Microsoft Research, hệ thống của chúng tôi sử dụng mạng nơ-ron này để tìm các đặc trưng của ảnh.

- ResNet-50 [2] là một kiến trúc mạng nơ-ron sâu được giới thiệu bởi Microsoft Research. Được xây dựng dựa trên ý tưởng của các mạng Residual Network

(ResNet), ResNet-50 là một phiên bản cụ thể của ResNet với tổng cộng 50 lớp (bao gồm các lớp tích chập và lớp kết nối đầy đủ).

- Mục tiêu chính của ResNet-50 là giải quyết vấn đề mất mát độ sâu (vanishing gradient) trong việc huấn luyện mạng nơ-ron sâu. Thông qua việc sử dụng các khối residual, ResNet-50 cho phép việc xây dựng các mạng sâu hơn mà vẫn giữ được hiệu suất tốt.
- Kiến trúc của ResNet-50 được chia thành các khối cơ bản gọi là residual blocks. Mỗi khối này bao gồm một chuỗi các lớp tích chập và kết nối đầy đủ. Đặc điểm của ResNet-50 là sự hiện diện của các khối bottleneck, trong đó kích thước đầu vào được giảm xuống trước khi tăng lên trở lại. Điều này giúp giảm độ phức tạp tính toán của mạng và cải thiện hiệu suất.
- ResNet-50 đã được huấn luyện trước trên một bộ dữ liệu lớn gọi là ImageNet. ImageNet là một tập dữ liệu ảnh rất lớn với hàng triệu hình ảnh thuộc vào hàng nghìn lớp khác nhau. Quá trình huấn luyện trước này giúp ResNet-50 học được các đặc trưng cơ bản của ảnh từ bộ dữ liệu này.

Trong hệ thống, chúng tôi hỗ trợ xử lý các định dạng file dữ liệu ảnh như .png và .jpg. Điều này cho phép người dùng tải lên các file ảnh có định dạng phổ biến và sử dụng chúng để tìm kiếm hình ảnh tương đồng.

Để lưu trữ và truy xuất các vector biểu diễn của ảnh, chúng tôi sử dụng Vector store dựa trên SQLite. Vector store này giúp chúng tôi lưu trữ các vector biểu diễn ảnh một cách hiệu quả và tiếp cận chúng một cách nhanh chóng.

2.3. Online processing

Phần Online processing làm nhiệm vụ tương tác với người dùng thời gian thực để trả lời những thông tin mà họ mong muốn, chúng tôi sẽ giải thích cách xử lý câu truy vấn và chức năng xếp hạng truy xuất cho 3 bài toán: truy xuất tài liệu, truy xuất ảnh và tìm kiếm ảnh tương đồng. Đồng thời cũng giải thích cách chúng tôi thực hiện liên kết với mô hình ngôn ngữ để tạo ra câu trả lời với ngữ cảnh là kết quả truy xuất của hệ thống.

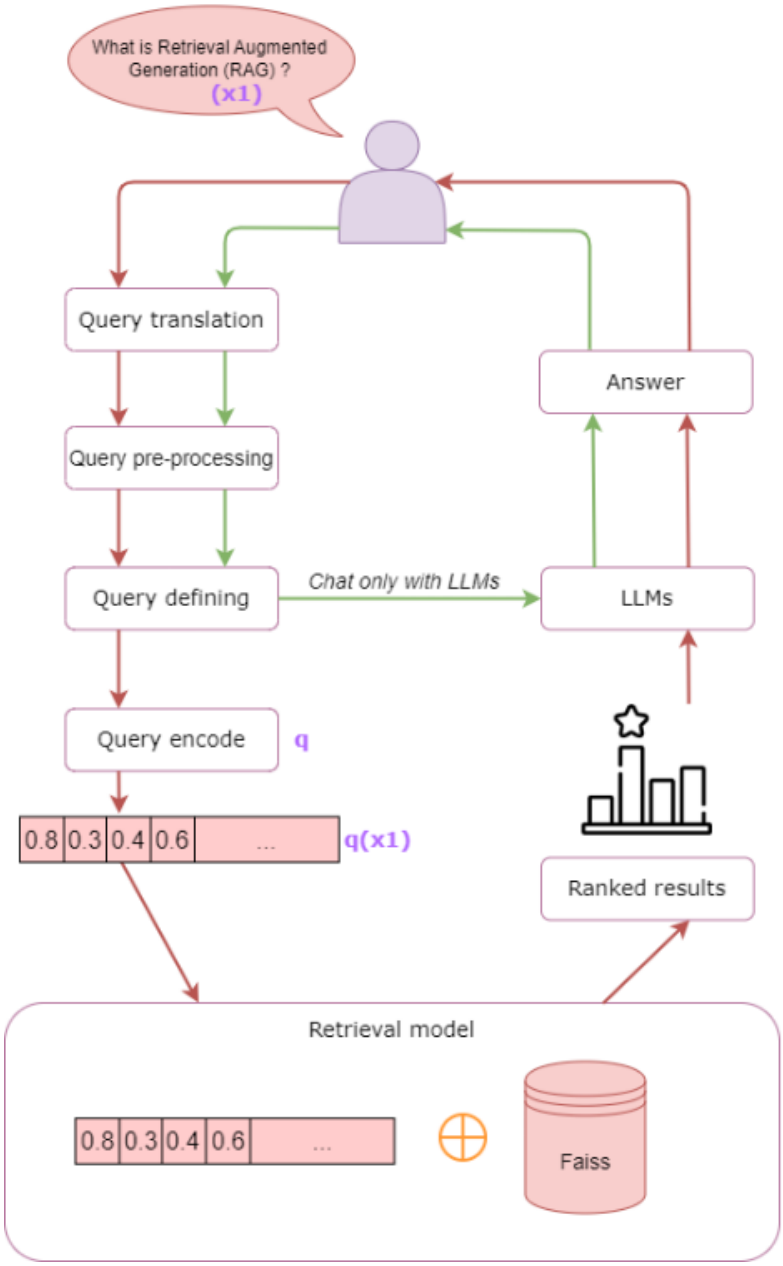
2.3.1. Xử lý truy vấn

Xử lý truy vấn là quá trình chuyển đổi một truy vấn của người dùng thành một câu truy vấn có thể được hiểu và xử lý bởi hệ thống máy tính. Trong đề tài lần này, chúng tôi tiến hành thực hiện những việc xử lý truy vấn như sau:

- **Dịch tiếng Việt sang tiếng Anh:** Công việc xử lý truy vấn này được thực hiện nhằm mục đích dễ dàng xác định mục tiêu của câu lệnh truy vấn là truy xuất văn bản hay truy xuất hình ảnh. Khi đó truy vấn được viết bằng tiếng Việt, cần phải dịch sang tiếng Anh để hệ thống máy tính có thể tìm thấy các key word để hiểu được nhu cầu của người dùng. Để thực hiện được công việc này, chúng tôi sử dụng thư viện translator của google.
- **Tiền xử lý truy vấn:** Quá trình này bao gồm các bước chuyển đổi chữ in hoa thành chữ thường, loại bỏ các ký tự đặc biệt hay các dấu câu, loại bỏ các từ không liên quan hay không mang nhiều ngữ nghĩa (stopword). Những việc này được thực hiện nhằm cải thiện hiệu suất của hệ thống chatbot.
- **Xác định câu truy vấn dùng cho tác vụ:** Quá trình này bao gồm việc xác định loại truy vấn mà người dùng đang thực hiện. Có ba loại truy vấn chính:
    - Truy xuất tài liệu: Truy vấn này yêu cầu hệ thống máy tính trả về các tài liệu có liên quan đến truy vấn. Với câu lệnh truy vấn có chứa từ “document” sẽ tiến hành truy vấn thông tin của những tài liệu được tải lên hệ thống. Còn nếu như không có sẽ tiến hành truy vấn thông qua ChatGPT.
    - Truy xuất hình ảnh: Truy vấn này yêu cầu hệ thống máy tính trả về các hình ảnh có liên quan đến truy vấn. Với câu lệnh truy vấn có chứa từ mang ý nghĩa tìm kiếm (find, look, …) cùng với từ mang ý nghĩa về hình ảnh (image, picture, photo, …) sẽ tiến hành truy vấn những hình ảnh phù hợp với nhu cầu của người dùng.
    - Tìm kiếm ảnh tương tự: Truy vấn này yêu cầu hệ thống máy tính trả về các hình ảnh có tính tương tự với hình ảnh được cung cấp. Tại tác vụ này chỉ cần upload ảnh vào mục tìm kiếm ảnh tương đồng, hệ thống sẽ tự hiểu và trả về những hình ảnh tương tự.

2.3.2. Chức năng xếp hạng truy xuất

2.3.2.1. Hệ thống truy xuất tài liệu



Hình 2.5 Quy trình xử lý online hệ thống truy xuất tài liệu

Các thành phần của online processing (Retrieve and Generate):

Query Encoders Biểu diễn query dưới dạng vector. Trong hệ thống chúng tôi sử dụng "text-embedding-ada-002" *(API của OpenAI)* để embedding. Với *x* là câu truy vấn bằng text do người dùng nhập vào:

$$h_x = q(x) = Embedding(x)$$

Retrieval model Với mỗi truy vấn *x* của người dùng, các chunk có liên quan sẽ được truy xuất từ kho lưu trữ *(trong hệ thống này là FAISS).* Đối với mỗi câu truy vấn *x*, chúng tôi sử dụng **tìm kiếm dựa trên tích vô hướng (dot product)** để tìm top-K các tài liệu z_i phù hợp từ H_c [5]. Độ tương đồng giữa vector truy vấn h_x và các chunk $H_c = [h_{c1}, h_{c2}, ..., h_{cM}]$ được biểu diễn như sau:

$$sim(h_x, h_{ci}) = {h_x}^T * h_{ci}$$

Với h_x là vector embedding của câu truy vấn *x* và h_{ci} là vector embedding *i-th* trong H_c. $sim(h_x, h_{ci})$ càng cao, thể hiện mức độ phù hợp càng cao giữa câu truy vấn *x* và tài liệu z_i. Tương tự, ta sắp xếp giảm dần theo mức độ tương đồng với câu truy vấn và lấy được top-K các tài liệu z_i:

$$TopK\big(sort(sim(h_x, H_c), reverse = True)\big)$$

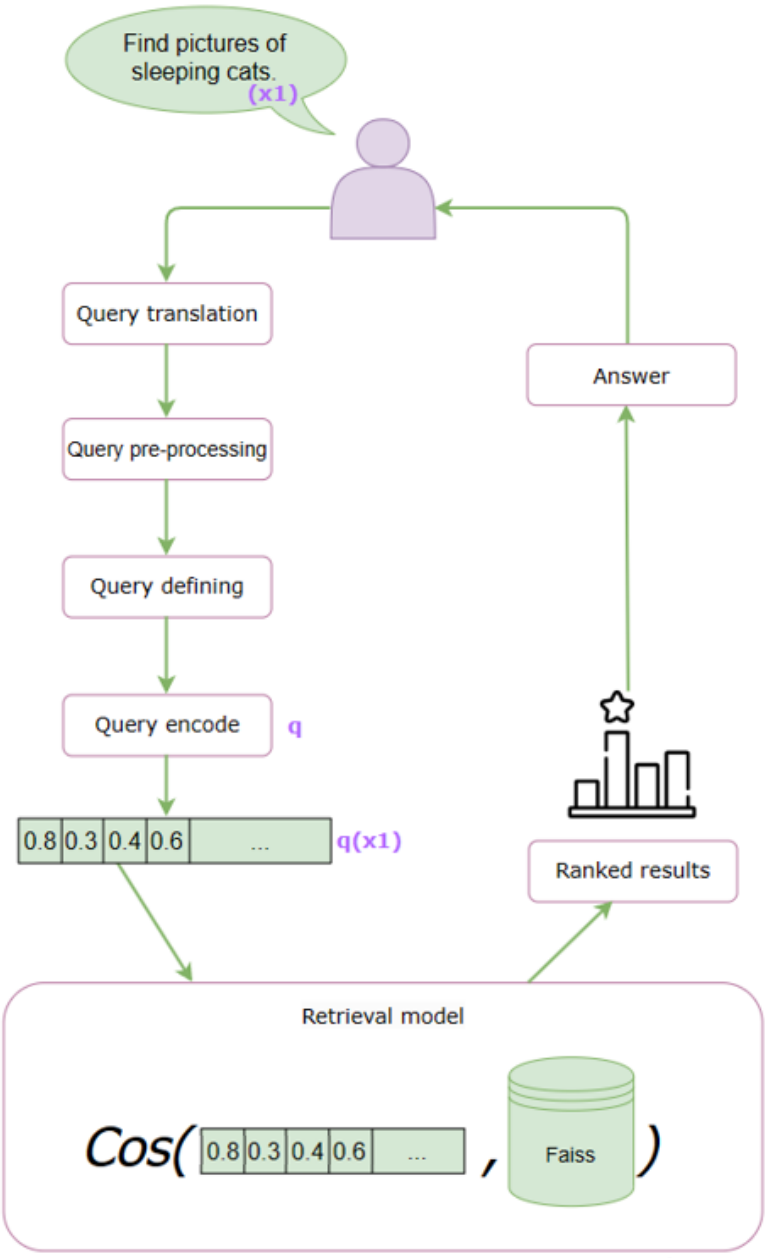
Argument Câu truy vấn của người dùng và ngữ cảnh bổ sung từ việc truy vấn được *(top-K các tài liệu z_i)* sẽ được đưa vào a prompt template**.**

Generation ChatModel/LLMs sinh ra câu trả lời dựa trên prompt được cung cấp. Xây dựng giống Generator trong nghiên cứu Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks [4]. Tuy nhiên, hệ thống của chúng tôi sử dụng ChatGPT *(API của OpenAI)* thay vì BART. Ngoài ra, hệ thống của chúng tôi cũng liên kết được với FLAN-T5 [10] thông qua API của Hugging Face để sinh ra câu trả lời.

Chat với mô hình ngôn ngữ lớn: trường hợp query của người dùng không đề cập tới bộ dữ liệu, query sẽ gọi thẳng cho LLMs để tìm kết quả mà không có ngữ cảnh về bộ dữ liệu *(bỏ qua Query Encoders và Retrieval model)*.

Đối với hệ thống truy xuất tài liệu này, chúng tôi chỉ quan tâm Prompt bao gồm câu hỏi và ngữ cảnh từ kết quả truy vấn được. Không quan tâm tới ngữ cảnh lịch sử cuộc trò chuyện, chính vì thế nên mọi query của người dùng là độc lập với nhau trong cuộc trò chuyện. Giải pháp này giúp hệ thống tối ưu (không làm dài thêm độ dài của prompt) và trả lời đúng context của tài liệu truy vấn được, tránh tình trạng hệ thống bị phụ thuộc vào ngữ cảnh của cuộc trò chuyện.

2.3.2.2. Hệ thống truy xuất hình ảnh



Hình 2.6 Quy trình xử lý online hệ thống truy xuất hình ảnh

Các thành phần của online processing:

Query Translation với mục tiêu vượt qua rào cản ngôn ngữ và tạo ra sự giao tiếp và hiểu biết hiệu quả giữa người dùng nói các ngôn ngữ khác nhau. Giúp mô hình dễ dàng hiểu được như cầu tìm kiếm hình ảnh từ người dùng. Tại đây chúng tôi sử dụng thư viện translation của google để giải quyết vấn đề.

Query Defining Để phân biệt được những dòng text của người dùng có mục đích là tìm kiếm văn bản hay tìm kiếm hình ảnh thì công việc này sẽ giúp hệ thống làm điều đó. Ở giai đoạn này, hệ thống sẽ tiến hành kiểm tra câu truy vấn đã được translation và tiền xử lý. Nếu trong câu có các từ mang nghĩa "tìm kiếm" cùng với các từ mang nghĩa "hình ảnh" thì hệ thống sẽ xác định đây là câu truy vấn hình ảnh. Còn không có sẽ là câu truy vấn thông tin.

Query Encoders Biểu diễn query dưới dạng vector. Trong hệ thống chúng tôi sử dụng mô hình Vision Transformer (ViT) để embedding. Với *x* là câu truy vấn bằng text do người dùng nhập vào:

$$\text{ViT}(x)=\text{Transformer}(\text{PatchEmbedding}(x))$$

Trong đó, PatchEmbedding là một lớp chuyển đổi hình ảnh thành các patch, nghĩa là các miền nhỏ hơn có kích thước cố định. Transformer là một mạng Transformer được áp dụng lên các patch đã nhúng để xử lý thông tin và tạo ra vectơ biểu diễn cuối cùng cho hình ảnh.

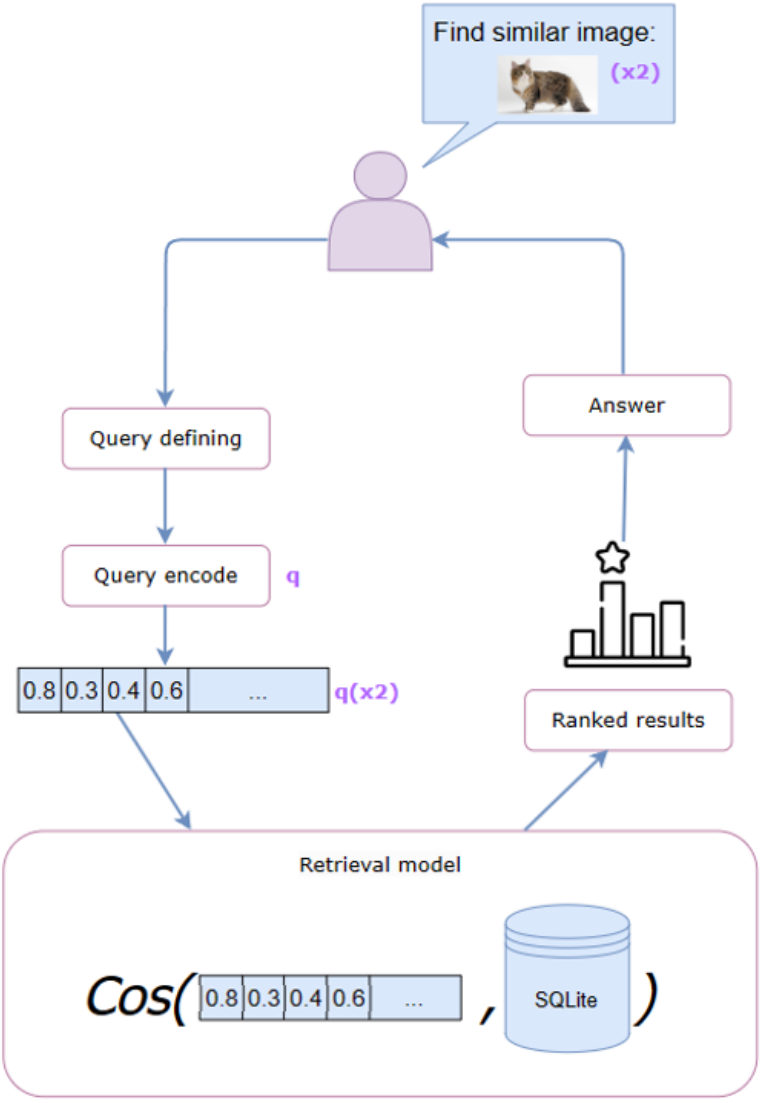
Retrieval model Với mỗi truy vấn *x* của người dùng, các hình ảnh có liên quan sẽ được truy xuất từ kho lưu trữ *(trong hệ thống này là FAISS).* Đối với mỗi câu truy vấn *x*, chúng tôi sử dụng **tìm kiếm dựa trên hàm tương đồng Cosine** để tìm top-K các hình ảnh z_i phù hợp từ H_c. Độ tương đồng giữa vector truy vấn h_x và các vector hình ảnh có trong kho lưu trữ $H_c = [h_{c1}, h_{c2}, ..., h_{cM}]$ được biểu diễn như sau:

$$cosine(h_x, h_{ci}) = \frac{h_x . h_{ci}}{\|h_x\| . \|h_{ci}\|}$$

Với h_x là vector embedding của câu truy vấn *x* và h_{ci} là vector embedding *i-th* trong H_c. $cosine(h_x, h_{ci})$ càng cao, thể hiện mức độ phù hợp càng cao giữa câu truy vấn *x* và hình ảnh z_i. Tương tự, ta sắp xếp giảm dần theo mức độ tương đồng với câu truy vấn và lấy được top-K các hình ảnh z_i:

$$TopK\big(sort(sim(h_x, H_c), reverse = True)\big)$$

2.3.2.2. Hệ thống tìm kiếm hình ảnh tương đồng



Hình 2.7 Quy trình xử lý online hệ thống tìm kiếm hình ảnh tương đồng

Các thành phần của online processing:

Query Defining Để phân biệt được tác vụ này so với các tác vụ khác, hệ thống tiến hành xác định bằng ở mục loading ảnh. Nếu có ảnh được tải lên thì hệ thống sẽ tiến hành xử lý công việc tìm kiếm hình ảnh tương đồng. Còn không có ảnh tải lên thì hệ thống xác định đây là tác vụ khác.

Query Encoders mô hình ResNet-50 có thể trích xuất các đặc trưng (features) từ hình ảnh, và sau đó sử dụng các phương pháp khác để thực hiện nhúng dữ liệu. Thông thường, chúng ta có thể sử dụng một lớp mạng nơ-ron tiếp theo, chẳng hạn như một lớp kết nối đầy đủ (fully connected layer) hoặc một lớp biểu diễn (representation layer), để chuyển đổi các đặc trưng từ ResNet-50 thành các vectơ

nhúng có số chiều cố định. Các vectơ nhúng này có thể được sử dụng để so sánh và tìm kiếm hình ảnh tương đồng, hoặc có thể đưa vào các mô hình học máy khác để thực hiện các tác vụ như phân loại, phát hiện vật thể, hay nhận dạng.

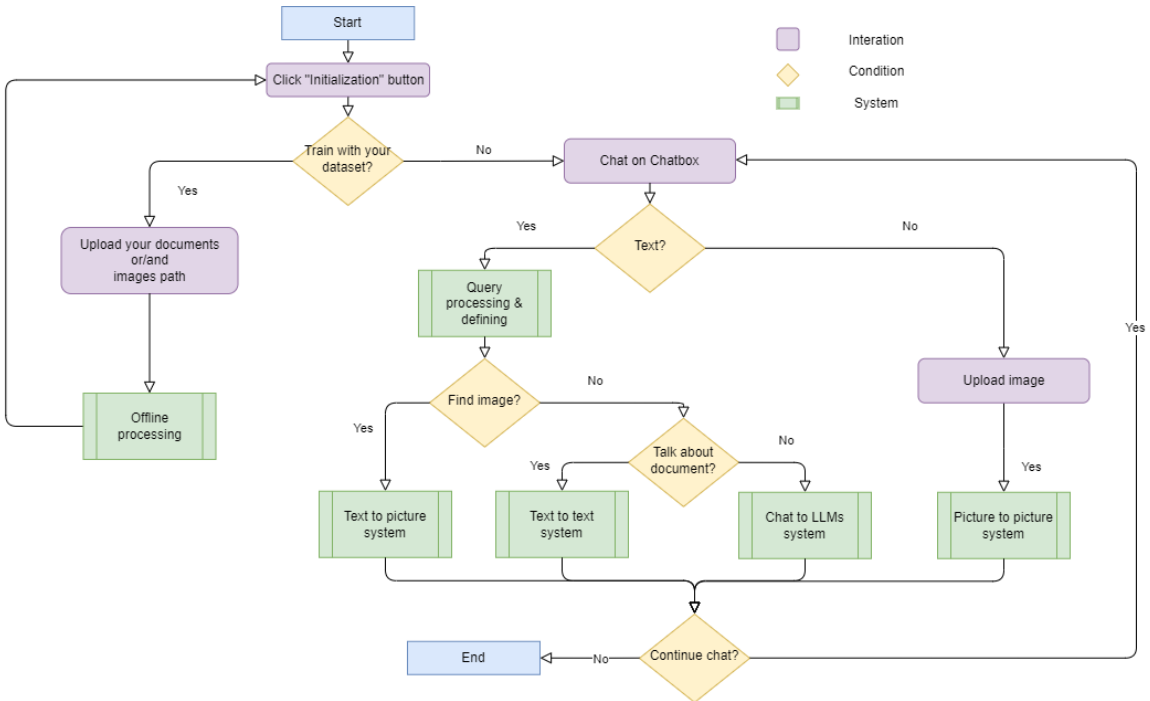
Retrieval model Với mỗi hình ảnh *x* của người dùng, các hình ảnh có liên quan sẽ được truy xuất từ kho lưu trữ *(trong hệ thống này là SQLite).* Đối với mỗi hình *x*, chúng tôi sử dụng **tìm kiếm dựa trên hàm tương đồng Cosine** để tìm top-K các hình ảnh z_i phù hợp từ H_c. Độ tương đồng giữa vector hình ảnh h_x và các vector hình ảnh có trong kho lưu trữ $H_c = [h_{c1}, h_{c2}, ..., h_{cM}]$ được biểu diễn như sau:

$$cosine(h_x, h_{ci}) = \frac{h_x . h_{ci}}{\|h_x\| . \|h_{ci}\|}$$

Với h_x là vector embedding của câu truy vấn *x* và h_{ci} là vector embedding *i-th* trong H_c. $cosine(h_x, h_{ci})$ càng cao, thể hiện mức độ phù hợp càng cao giữa hình ảnh *x* và hình ảnh z_i. Tương tự, ta sắp xếp giảm dần theo mức độ tương đồng với hình ảnh và lấy được top-K các hình ảnh z_i:

$$TopK\big(sort(sim(h_x, H_c), reverse = True)\big)$$

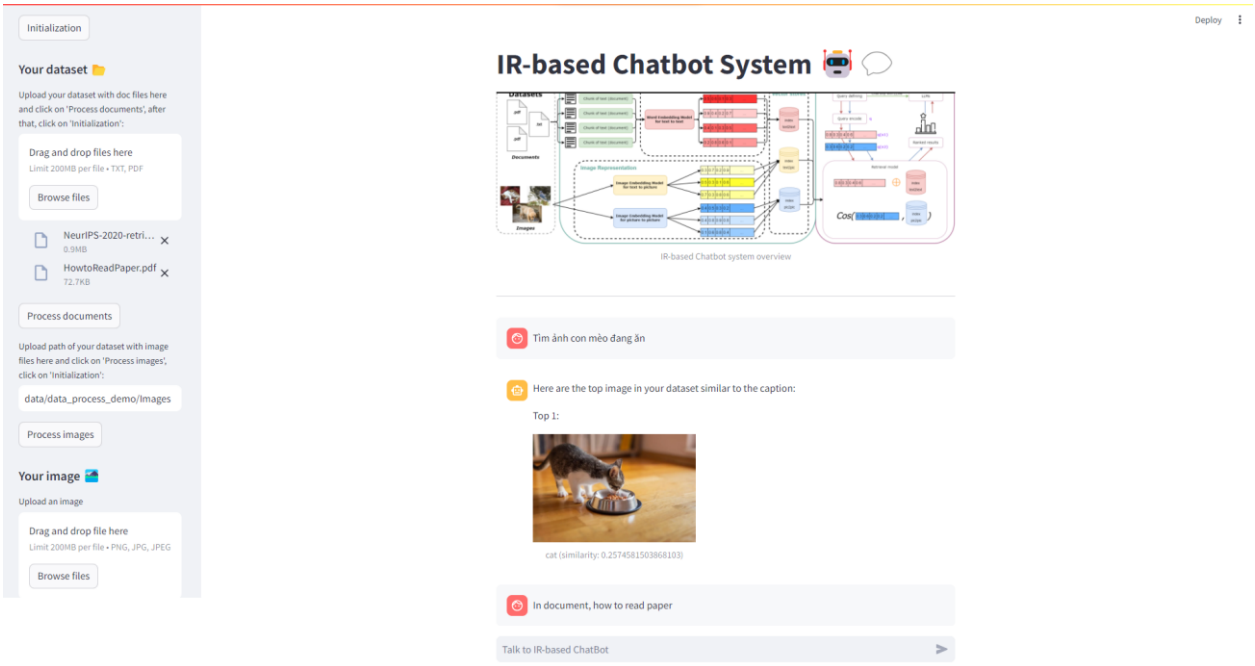
3. TRIỂN KHAI HỆ THỐNG



Hình 3.1 Userflow tương tác với IR-based Chatbot system

Người dùng có thể tương tác bằng cách tùy chỉnh bộ dataset hoặc chat với hệ thống. Chúng tôi thiết đặt nút 'Initialization' nhằm tách luồng khởi tạo ra khỏi luồng chat của

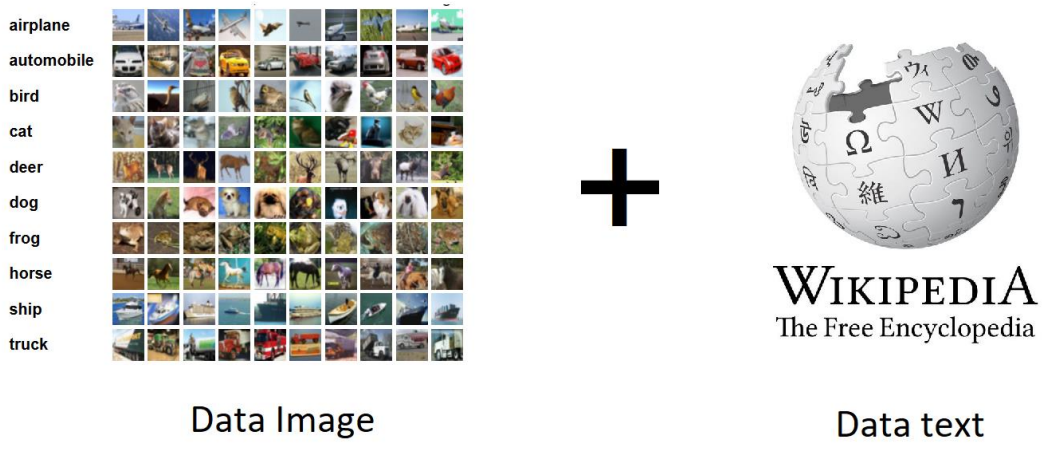
chatbot (thao tác này giúp hệ thống không khởi tạo lại hệ thống trong quá trình chat, làm tăng hiệu năng chat)



Hình 3.2 Giao diện web hệ thống IR-based Chatbot

3.1. Bộ dữ liệu thực nghiệm

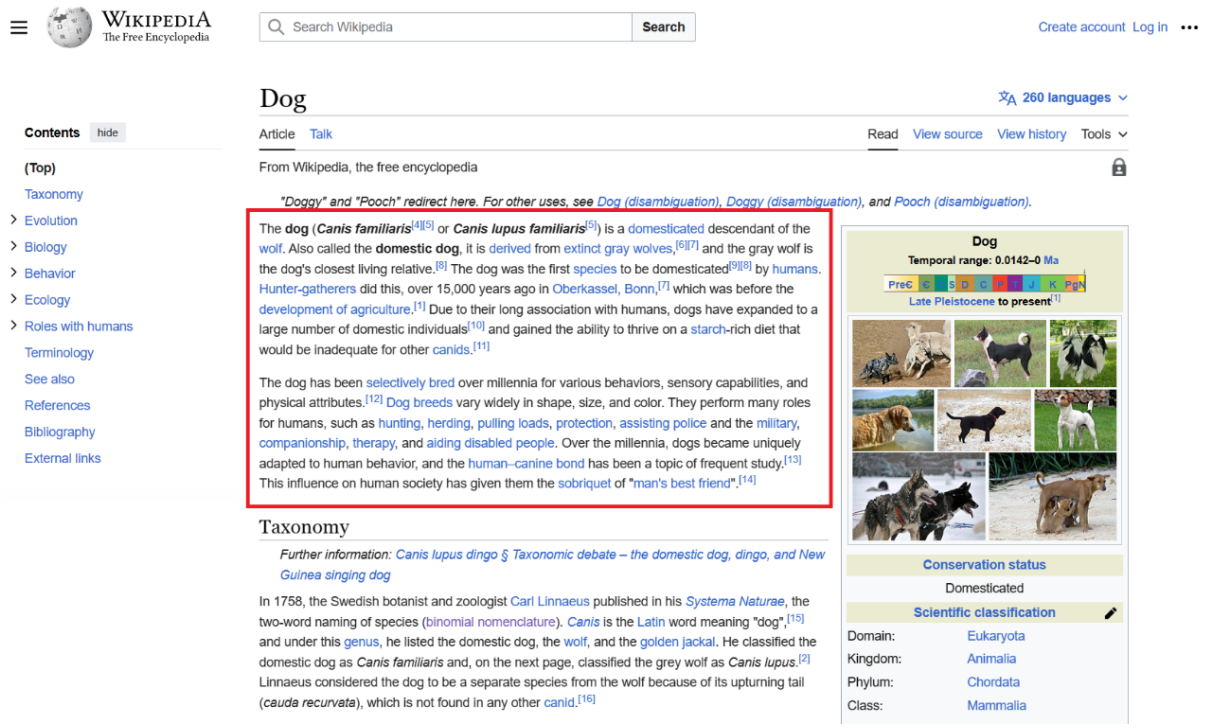
Để đáp ứng được mục tiêu của đề tài này, chúng tôi đã xây dựng một bộ dữ liệu phù hợp với tài nguyên máy tính mà chúng tôi sử dụng. Sau nhiều lần thử nghiệm và nghiên cứu bộ dữ liệu đảm bảo chất lượng nhất, chúng tôi đã quyết định thu thập một bộ dữ liệu gồm 10 nhãn ảnh với 25 tấm mỗi nhãn, cùng với đó là 10 file text tương ứng 10 nhãn. Bộ dữ liệu này được lấy ý tưởng từ CIFAR-10, một bộ dữ liệu tiêu chuẩn được sử dụng phổ biến để huấn luyện và đánh giá các mô hình học máy thị giác.



Hình 3.3 Xây dựng bộ dữ liệu thực nghiệm

Để xây dựng bộ dữ liệu, chúng tôi sử dụng các phương pháp sau:

- Thu thập ảnh: Chúng tôi tiến hành thu thập ảnh từ các nguồn như Google Images, Flickr,... Các ảnh được thu thập phải có chất lượng trung bình-cao (kích thước tối thiểu 500x500), đa dạng nội dung, không bị mờ, nhiễu, hoặc nhãn mục tiêu không là mục tiêu chính của ảnh.
- Thu thập văn bản: Chúng tôi thu thập văn bản mô tả các nhãn mục tiêu từ nguồn thông tin Wikipedia. Văn bản được thu thập phải đầy đủ, chính xác, và phù hợp với nhãn mục tiêu.



Hình 3.4 Ví dụ phần thông tin được thu thập cho dữ liệu văn bản.

Sau khi thu thập dữ liệu chúng tôi đã xây dựng thành công bộ dữ liệu với 10 nhãn ảnh và 10 file text tương ứng. Bộ dữ liệu có các đặc điểm sau:

- Số lượng ảnh: 250 tấm ảnh, chia đều cho 10 nhãn mục tiêu.
- Kích thước ảnh: Giao động từ 500x500 pixel đến 800x800 pixel.
- Định dạng ảnh: PNG, JPG.
- Số lượng file text: 10 file, mỗi file mô tả một nhãn mục tiêu.
- Định dạng file text: TXT.

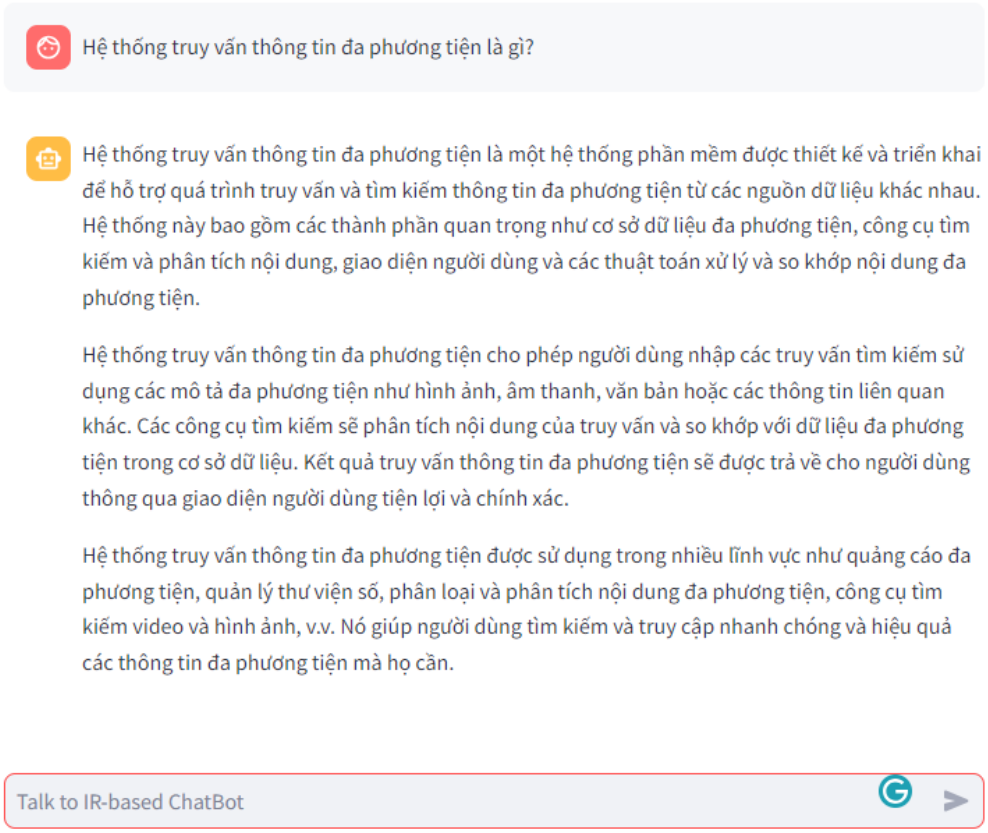
Bộ dữ liệu được xây dựng trong báo cáo này có thể được sử dụng để huấn luyện và đánh giá các mô hình học máy thị giác. Bộ dữ liệu này có chất lượng tốt đối với các nguồn tài nguyên hạn hẹp, đáp ứng được các yêu cầu của các mô hình học máy hiện đại.

3.2. Tính năng tùy chỉnh bộ dữ liệu

Đối với tính năng này, người dùng có thể tùy chỉnh bộ dữ liệu văn bản bằng cách upload các tài liệu có định dạng .txt, .pdf vào phần upload files hoặc tùy chỉnh bộ dữ liệu hình ảnh bằng cách sao chép đường dẫn tới thư mục hình ảnh chứa các folder là class các ảnh. Sau đó nhấn nút ‘Process documents’ hoặc ‘Process images’. Sau khi xử lý thành công, hệ thống sẽ thông báo “Completed!” và người dùng chỉ cần nhấn nút 'Initialization' để khởi tạo lại hệ thống.

3.3. Tính năng hỏi đáp với mô hình ngôn ngữ

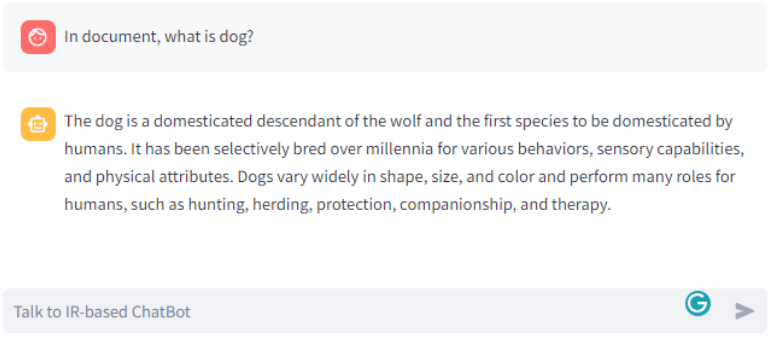
Ở tính năng này, người dùng có thể chat để hỏi bất kỳ thông tin gì cho hệ thống, câu truy vấn có thể bằng tiếng Anh hoặc tiếng Việt. Tuy nhiên câu truy vấn phải không đề cập tới nội dung dữ liệu của người dùng.



Hình 3.5 Kết quả một câu truy vấn của tính năng hỏi đáp với mô hình ngôn ngữ

3.4. Tính năng hỏi đáp về tài liệu

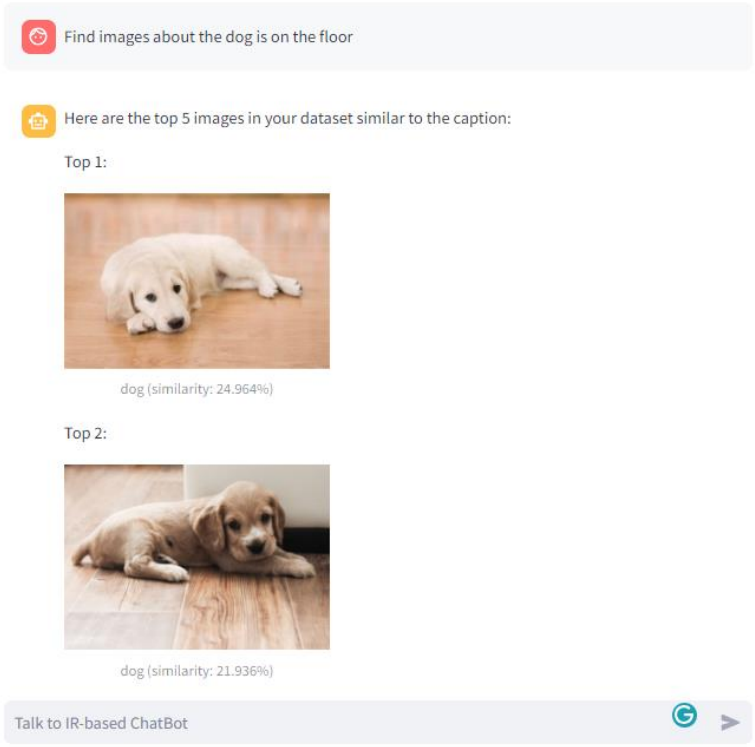
Ở tính năng này, người dùng có thể chat để hỏi bất kỳ thông tin gì liên quan tới dữ liệu của người dùng, câu truy vấn có thể bằng tiếng Anh hoặc tiếng Việt. Tuy nhiên chữ "document" hoặc "tài liệu" phải xuất hiện trong câu truy vấn.



Hình 3.6 Kết quả một câu truy vấn của tính năng hỏi đáp về tài liệu

3.5. Tính năng tìm kiếm ảnh bằng ngôn ngữ

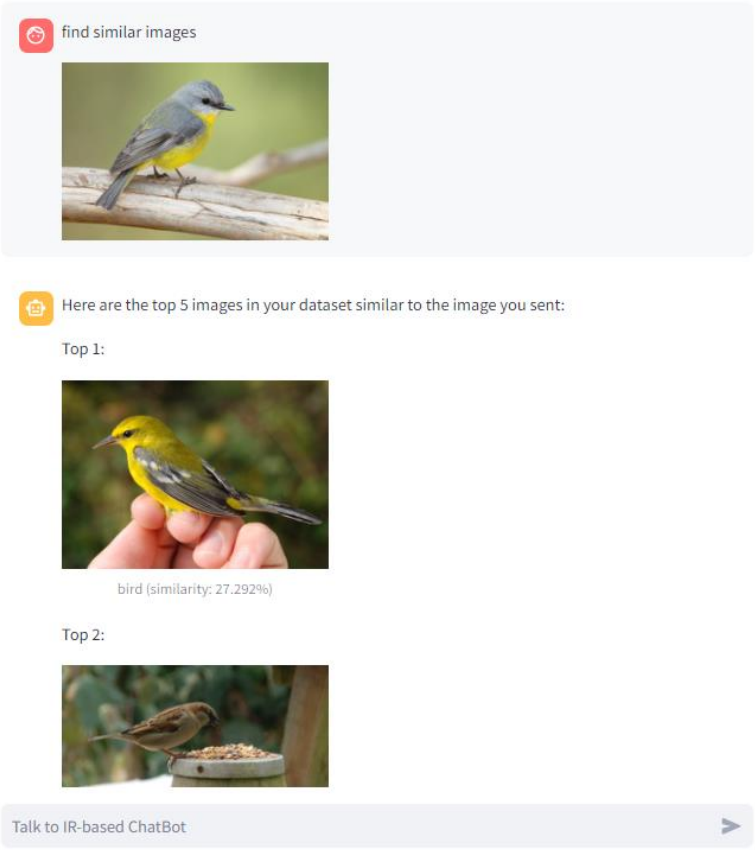
Để dùng được tính năng này, điều kiện là trong câu truy vấn phải xuất hiện các từ liên quan đến tìm kiếm ảnh *(ví dụ: "find picture", 'photo', 'ảnh',...),* câu truy vấn có thể bằng tiếng Anh hoặc tiếng Việt. Khi đủ điều kiện, hệ thống sẽ hiển thị kết quả các ảnh có trong bộ dữ liệu của người dùng có mức độ phù hợp với câu truy vấn giảm dần.



Hình 3.7 Kết quả một câu truy vấn của tính năng tìm kiếm ảnh bằng ngôn ngữ

3.6. Tính năng tìm kiếm ảnh tương đồng

Để dùng tính năng này, điều kiện là trong câu truy vấn phải có ảnh muốn tìm kiếm được upload lên. Hệ thống sẽ hiển thị kết quả các ảnh có trong bộ dữ liệu có mức độ tương đồng giảm dần.



Hình 3.8 Kết quả một câu truy vấn của tính năng tìm kiếm ảnh tương đồng

4. KẾT LUẬN

4.1. Kết quả

Thông qua quá trình nghiên cứu và thực hiện, nhóm đã đạt được các kết quả sau:

- Xây dựng được hệ thống truy xuất thông tin trên cả ảnh và văn bản: Hệ thống truy xuất thông tin được xây dựng trên cơ sở các phương pháp truy xuất thông tin truyền thống, kết hợp với các phương pháp học máy. Hệ thống có khả năng xử lý các truy vấn tìm kiếm cả trên văn bản và hình ảnh.
- Đề xuất được một hệ thống chatbot framework: Hệ thống chatbot framework được thiết kế linh hoạt, cho phép người dùng có thể dễ dàng tùy chỉnh để phù hợp với nhu cầu của mình. Hệ thống bao gồm các tác vụ như tìm thông tin cừ

một văn bản, tìm hình ảnh từ một văn bản và tìm hình ảnh tương tự với hình ảnh được tải lên.

- Có hệ thống demo hoàn chỉnh: Hệ thống demo hoàn chỉnh được xây dựng trên cơ sở hệ thống chatbot framework đã đề xuất với thư viện Streamlit. Hệ thống demo cho phép người dùng có thể chat với bất kỳ bộ dữ liệu bao gồm tài liệu và ảnh mà người dùng mong muốn.

Các kết quả đạt được trong đề tài này đã đáp ứng được các yêu cầu của đề tài. Hệ thống truy xuất thông tin và hệ thống chatbot framework được xây dựng có thể được ứng dụng trong nhiều lĩnh vực khác nhau.

4.2. Hạn chế

Bên cạnh những kết quả đạt được, nghiên cứu cũng còn tồn tại một số hạn chế sau:

- Chưa có benchmark để đánh giá độ hiệu quả của hệ thống: Để đánh giá độ hiệu quả của hệ thống, cần có một bộ dữ liệu đánh giá (benchmark) phù hợp. Bộ dữ liệu này cần bao gồm các truy vấn có độ khó khác nhau, từ đó có thể đánh giá được khả năng xử lý các truy vấn của hệ thống.
- Chưa thực nghiệm so sánh các độ đo khoảng cách và đánh giá mức độ hiệu quả của các thư viện vectorstore: Hiện tại, hệ thống sử dụng độ đo dot product và cosine để tính toán độ tương tự giữa các văn bản và hình ảnh. Để lựa chọn được độ đo khoảng cách phù hợp, cần thực nghiệm so sánh các độ đo khoảng cách khác nhau và đánh giá mức độ hiệu quả của chúng. Các thư viện vectorstore cũng cần được so sánh về hiệu năng tính toán và tính chính xác.
- Chưa tối ưu code, hiệu năng tính toán và kiến trúc hệ thống: Cần tối ưu code để hệ thống chạy hiệu quả hơn. Hiệu năng tính toán của hệ thống cũng cần được cải thiện bằng cách sử dụng các kỹ thuật tối ưu hóa. Kiến trúc hệ thống cũng cần được tối ưu để đáp ứng được các yêu cầu về hiệu năng và khả năng mở rộng.

Việc khắc phục các hạn chế trên sẽ giúp hệ thống chatbot trở nên hoàn thiện hơn và có thể ứng dụng được trong nhiều lĩnh vực khác nhau.

4.3. Hướng phát triển

Để hệ thống chatbot trở nên hoàn thiện hơn, nhóm nghiên cứu đề xuất một số hướng phát triển sau:

- Tiến hành đánh giá theo các benchmark và tinh chỉnh hệ thống: Bộ dữ liệu đánh giá sẽ được xây dựng để đánh giá độ hiệu quả của hệ thống. Trên cơ sở kết quả đánh giá, hệ thống sẽ được tinh chỉnh để cải thiện hiệu năng và tính chính xác.
- Tích hợp nhiều bài toán hơn cho hệ thống chatbot đã được đề xuất: Hiện tại, hệ thống chatbot chỉ có thể xử lý các truy vấn đơn giản. Để hệ thống có thể đáp ứng được nhu cầu của người dùng một cách tốt hơn, cần tích hợp nhiều bài toán hơn cho hệ thống, bao gồm các bài toán như: Trả lời các câu hỏi mở, tạo ra các câu trả lời sáng tạo và thú vị, học hỏi từ tương tác với người dùng, v.v.
- Tối ưu kiến trúc hệ thống bằng cách giảm bớt các thành phần không cần thiết hoặc thay thế các thư viện có khả năng tính toán tốt hơn: Hiện tại, hệ thống vẫn còn một số điểm chưa tối ưu về kiến trúc. Để cải thiện hiệu năng của hệ thống, cần tối ưu kiến trúc hệ thống bằng cách giảm bớt các thành phần không cần thiết hoặc thay thế các thư viện có khả năng tính toán tốt hơn.

Việc tối ưu kiến trúc hệ thống sẽ giúp hệ thống chạy hiệu quả hơn và đáp ứng được các yêu cầu về hiệu năng và khả năng mở rộng.

TÀI LIỆU THAM KHẢO

[1] Radford, A. a. Kim, J. W. a. Hallacy, C. a. Ramesh, A. a. Goh, G. a. Agarwal, S. a. Sastry, G. a. Askell, A. a. Mishkin, P. a. Clark và J. a. others, "Learning transferable visual models from natural language supervision," trong *International conference on machine learning*, PMLR, 2021, pp. 8748-8763.

[2] He, K. a. Zhang, X. a. Ren, S. a. Sun và Jian, "Deep residual learning for image recognition," trong *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770--778.

[3] Douze, M. a. Guzhva, A. a. Deng, C. a. Johnson, J. a. Szilvasy, G. a. Mazare, P.-E. a. Lomeli, M. a. Hosseini, L. a. Jegou và Herve, "The Faiss library," *arXiv preprint arXiv:2401.08281,* 2024.

[4] Lewis, P. a. Perez, E. a. Piktus, A. a. Petroni, F. a. Karpukhin, V. a. Goyal, N. a. Kuttler, H. a. Lewis, M. a. Yih, W.-t. a. Rocktaschel và T. a. others, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, tập 33, pp. 9459-9474, 2020.

[5] Karpukhin, V. a. Oguz, B. a. Min, S. a. Lewis, P. a. Wu, L. a. Edunov, S. a. Chen, D. a. Yih and Wen-tau, "Dense passage retrieval for open-domain question answering," *arXiv preprint arXiv:2004.04906*, 2020.

[6] Reimers, Gurevych và N. a. Iryna, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[7] Su, H. a. Shi, W. a. Kasai, J. a. Wang, Y. a. Hu, Y. a. Ostendorf, M. a. Yih, W.-t. a. Smith, N. A. a. Zettlemoyer, L. a. Yu và Tao, "One embedder, any task: Instruction-finetuned text embeddings," *arXiv preprint arXiv:2212.09741,* 2022.

[8] Dosovitskiy, A. a. Beyer, L. a. Kolesnikov, A. a. Weissenborn, D. a. Zhai, X. a. Unterthiner, T. a. Dehghani, M. a. Minderer, M. a. Heigold, G. a. Gelly và S. a. others, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929,* 2020.

[9] Gadre, S. Y. a. Ilharco, G. a. Fang, A. a. Hayase, J. a. Smyrnis, G. a. Nguyen, T. a. Marten, R. a. Wortsman, M. a. Ghosh, D. a. Zhang và J. a. others, "DataComp: In search of the next generation of multimodal datasets," *arXiv preprint arXiv:2304.14108,* 2023.

[10] Chung, H. W. a. Hou, L. a. Longpre, S. a. Zoph, B. a. Tay, Y. a. Fedus, W. a. Li, Y. a. Wang, X. a. Dehghani, M. a. Brahma và S. a. others, "Scaling instruction-finetuned language models," *arXiv preprint arXiv:2210.11416,* 2022.