AI Course

Capstone Project   
Final Report

For students (instructor review required)

ⓒ2023 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright of this document.

This document is a literary property protected by copyright law so reprint and reproduction without permission are prohibited.

To use this document other than the curriculum of Samsung Innovation Campus, you must receive written consent from copyright holder.

**CHATBOT HỖ TRỢ TRA CỨU THÔNG TIN TRƯỜNG ĐH SƯ PHẠM KỸ THUẬT**

**25/08/2024**

**Group 2**

**Đặng Kim Thành**

**Mục lục**

**[1. Introduction](#_Toc175405294)** [6](#_Toc175405294)

**[1.1.](#_Toc175405295)****[Lý do chọn đề tài](#_Toc175405295)** [6](#_Toc175405295)

**[1.2.](#_Toc175405296)****[Động lực và mục tiêu](#_Toc175405296)** [6](#_Toc175405296)

**[1.2.1.](#_Toc175405297)****[Động lực](#_Toc175405297)** [6](#_Toc175405297)

**[1.2.2.](#_Toc175405298)****[Mục tiêu dự án](#_Toc175405298)** [7](#_Toc175405298)

**[1.3.](#_Toc175405299)****[Phân công nhiệm vụ](#_Toc175405299)** [7](#_Toc175405299)

**[1.4.](#_Toc175405300)****[Lịch trình làm việc](#_Toc175405300)** [7](#_Toc175405300)

**[2.](#_Toc175405301)****[Project Execution](#_Toc175405301)** [9](#_Toc175405301)

**[2.1.](#_Toc175405302)****[Thu thập dữ liệu](#_Toc175405302)** [9](#_Toc175405302)

**[2.1.1.](#_Toc175405303)****[Nguồn dữ liệu sử dụng](#_Toc175405303)** [9](#_Toc175405303)

**[2.1.2.](#_Toc175405304)****[Quá trình thu thập và xử lý cơ bản dữ liệu](#_Toc175405304)** [9](#_Toc175405304)

**[2.1.3.](#_Toc175405305)****[Mục tiêu và kết quả](#_Toc175405305)** [10](#_Toc175405305)

**[2.2.](#_Toc175405306)****[Thiết kế hệ thống](#_Toc175405306)** [10](#_Toc175405306)

**[2.3.](#_Toc175405307)****[Phương pháp huấn luyện](#_Toc175405307)** [14](#_Toc175405307)

**[2.3.1.](#_Toc175405308)****[Quy trình RAG (Retrieval-Augmented Generation)](#_Toc175405308)** [15](#_Toc175405308)

**[2.3.2.](#_Toc175405309)****[Framework Langchain](#_Toc175405309)** [16](#_Toc175405309)

**[2.3.3.](#_Toc175405310)****[Fine-tuning bằng PEFT và LoRA](#_Toc175405310)** [21](#_Toc175405310)

**[2.3.4.](#_Toc175405311)****[Kiểm thử và triển khai](#_Toc175405311)** [21](#_Toc175405311)

**[2.4.](#_Toc175405312)****[Quy trình công việc](#_Toc175405312)** [22](#_Toc175405312)

**[2.4.1.](#_Toc175405313)****[Mô hình ngôn ngữ lớn (LLM)](#_Toc175405313)** [22](#_Toc175405313)

**[2.4.2.](#_Toc175405314)****[Mô hình embedding](#_Toc175405314)** [23](#_Toc175405314)

**[2.4.3.](#_Toc175405315)****[Cơ sở dữ liệu vector hóa Pinecone](#_Toc175405315)** [24](#_Toc175405315)

**[3.](#_Toc175405316)****[Results](#_Toc175405316)** [26](#_Toc175405316)

**[3.1.](#_Toc175405317)****[Tiền xử lý dữ liệu](#_Toc175405317)** [26](#_Toc175405317)

**[3.2.](#_Toc175405318)****[Thiết lập mô hình](#_Toc175405318)** [27](#_Toc175405318)

**[3.3.](#_Toc175405319)****[Kiểm thử và cải tiến](#_Toc175405319)** [29](#_Toc175405319)

**[3.3.1.](#_Toc175405320)****[Quy trình kiểm thử](#_Toc175405320)** [29](#_Toc175405320)

**[3.3.2.](#_Toc175405321)****[Các chỉ số đánh giá](#_Toc175405321)** [30](#_Toc175405321)

**[3.3.3.](#_Toc175405322)****[Cải tiến](#_Toc175405322)** [33](#_Toc175405322)

**[4.](#_Toc175405323)****[Projected Impact](#_Toc175405323)** [34](#_Toc175405323)

**[4.1.](#_Toc175405324)****[Thành tựu và lợi ích](#_Toc175405324)** [34](#_Toc175405324)

**[4.2.](#_Toc175405325)****[Hướng phát triển tương lai](#_Toc175405325)** [35](#_Toc175405325)

**[5.](#_Toc175405326)****[Team Member Review and Comment](#_Toc175405326)**  [36](#_Toc175405326)

**[6.](#_Toc175405327)****[Instructor Review and Comment](#_Toc175405327)** [37](#_Toc175405327)

# **1. Giới thiệu đề tài**

* 1. **Lý do chọn đề tài**

Trong bối cảnh hiện nay, khi công nghệ thông tin và truyền thông phát triển mạnh mẽ, việc cung cấp thông tin chính xác và kịp thời cho sinh viên và các bên quan tâm đến trường đại học là vô cùng quan trọng. Với sự gia tăng của các kênh thông tin trực tuyến, việc tìm kiếm và tra cứu thông tin một cách nhanh chóng và chính xác trở nên ngày càng khó khăn hơn.

Đối với Trường Đại học Sư phạm Kỹ thuật TP.HCM, việc đảm bảo rằng sinh viên và người dùng có thể dễ dàng tiếp cận thông tin về các khóa học, học phí, quy trình tuyển sinh và các dịch vụ khác của trường là ưu tiên hàng đầu. Tuy nhiên, sự đa dạng và phong phú của nguồn thông tin có thể gây khó khăn cho người dùng trong việc tìm kiếm thông tin cần thiết. Hơn nữa, việc phản hồi chậm hoặc thông tin không rõ ràng có thể dẫn đến sự bối rối và giảm trải nghiệm người dùng.

Nhằm giải quyết những thách thức trên, dự án xây dựng một chatbot thông minh hỗ trợ tra cứu thông tin đã được thực hiện. Chatbot này được kỳ vọng sẽ không chỉ cung cấp thông tin một cách nhanh chóng và chính xác mà còn giúp giảm tải công việc cho bộ phận tư vấn và hỗ trợ sinh viên của nhà trường. Việc áp dụng các công nghệ mới như trí tuệ nhân tạo (AI) vào quá trình này hứa hẹn sẽ mang lại hiệu quả cao và cải thiện đáng kể chất lượng dịch vụ.

* 1. **Động lực và mục tiêu**
     1. **Động lực**

Là một sinh viên của Trường Đại học Sư phạm Kỹ thuật TP.HCM, tôi nhận thấy rằng việc truy cập và tra cứu thông tin một cách nhanh chóng và hiệu quả là vô cùng quan trọng. Trong quá trình học tập, tôi thường gặp khó khăn khi tìm kiếm các thông tin cần thiết từ nhiều nguồn khác nhau, từ các quy chế học tập đến thông tin về tuyển sinh, học phí, và các dịch vụ hỗ trợ sinh viên. Điều này không chỉ làm mất thời gian mà đôi khi còn dẫn đến hiểu lầm hoặc bối rối do thông tin không được cung cấp rõ ràng.

Động lực của tôi khi thực hiện dự án này là mong muốn tạo ra một giải pháp có thể cải thiện trải nghiệm của sinh viên và những người quan tâm đến trường. Tôi tin rằng việc áp dụng công nghệ trí tuệ nhân tạo, đặc biệt là chatbot, sẽ là một cách tiếp cận hiệu quả để giải quyết vấn đề này. Chatbot không chỉ giúp tự động hóa việc cung cấp thông tin mà còn đảm bảo tính nhất quán và chính xác của thông tin được cung cấp.

Thông qua dự án này, tôi hy vọng sẽ góp phần tạo ra một môi trường học tập và sinh hoạt thuận tiện hơn cho sinh viên, đồng thời hỗ trợ nhà trường trong việc quản lý và cung cấp thông tin một cách hiệu quả. Giảm thiểu thời gian tìm kiếm thông tin và cung cấp phản hồi nhanh chóng, chính xác.

* + 1. **Mục tiêu dự án**

Mục tiêu của dự án này là xây dựng một chatbot hỗ trợ tra cứu thông tin cho Trường Đại học Sư phạm Kỹ thuật TP.HCM. Cụ thể, chatbot này sẽ có khả năng:

• Cung cấp thông tin về trường: Bao gồm thông tin về các khóa học, chương trình đào tạo, cơ sở vật chất, và các hoạt động của trường.

• Tư vấn tuyển sinh: Giúp người dùng tìm hiểu về các yêu cầu tuyển sinh, quy trình đăng ký, và các chính sách học phí.

• Giải đáp những thắc mắc phổ biến liên quan đến học tập, sinh hoạt tại trường, và các vấn đề khác mà sinh viên thường gặp phải.

* 1. **Phân công nhiệm vụ**

|  |  |
| --- | --- |
| **Công việc** | **Người thực hiện** |
| Lập kế hoạch và chuẩn bị | Đặng Kim Thành |
| Thu thập dữ liệu | Đặng Kim Thành |
| Xử lý dữ liệu | Đặng Kim Thành |
| Tìm hiểu các loại mô hình và nghiên cứu | Đặng Kim Thành |
| Phát triển chatbot | Đặng Kim Thành |
| Kiểm thử và triển khai | Đặng Kim Thành |
| Đánh giá và hoàn thiện | Đặng Kim Thành |

* 1. **Lịch trình làm việc**

|  |  |  |
| --- | --- | --- |
| **Ngày** | **Công việc** | **Người thực hiện** |
| 07/07/2024 - 18/07/2024 | Lập kế hoạch và chuẩn bị | Đặng Kim Thành |
| Xác định mục tiêu, yêu cầu dự án, và phân chia công việc. | Đặng Kim Thành |
| 19/07/2024 - 22/07/2024 | Thu thập dữ liệu | Đặng Kim Thành |
| Kiểm soát tiến độ, đốc thúc thành viên, nghiên cứu trước kế hoạch kiểm thử | Đặng Kim Thành |
| Thu thập tài liệu về trường Sư Phạm Kỹ Thuật (pdf, video, website). | Đặng Kim Thành |
| Nghiên cứu mô hình RAG và chuẩn bị môi trường phát triển. | Đặng Kim Thành |
| 23/07/2024 -27/07/2024 | Xử lý dữ liệu | Đặng Kim Thành |
| Tiếp tục kiểm soát tiến độ, phát triển kế hoạch kiểm thử và chuẩn bị các công cụ kiểm thử | Đặng Kim Thành |
| Xây dựng bộ xử lý dữ liệu (chuyển đổi video thành văn bản, làm sạch, chuẩn hóa dữ liệu, tách đoạn). | Đặng Kim Thành |
| Tiếp tục nghiên cứu và bắt đầu xây dựng dự án, chuẩn bị các bộ công cụ cần thiết. | Đặng Kim Thành |
| 28/07/2024 - 02/08/2024 | Tìm hiểu các loại mô hình và nghiên cứu | Đặng Kim Thành |
| Xem xét và đảm bảo chất lượng dữ liệu đã thu thập. | Đặng Kim Thành |
| Xây dựng bộ lưu trữ dữ liệu và cơ sở dữ liệu | Đặng Kim Thành |
| Tìm hiểu về kỹ thuật prompt engineering và thử nghiệm với các prompt khác nhau để tối ưu hóa đầu ra | Đặng Kim Thành |
| 03/08/2024 - 11/08/2024 | Phát triển chatbot | Đặng Kim Thành |
| Quản lý, kiểm thử quá trình tích hợp và chất lượng mô hình. | Đặng Kim Thành |
| Tích hợp cơ sở dữ liệu vào quy trình thu thập ngữ cảnh. | Đặng Kim Thành |
| Tối ưu hóa quá trình thu thập ngữ cảnh và quá trình xử lý của chatbot | Đặng Kim Thành |
| 12/08/2024 - 15/08/2024 | Kiểm thử và triển khai | Đặng Kim Thành |
| Kiểm thử chức năng và hiệu năng của chatbot | Đặng Kim Thành |
| Đánh giá tổng kết chất lượng dự án. | Đặng Kim Thành |
| Cải tiến chatbot theo kết quả kiểm thử. | Đặng Kim Thành |
| Chuẩn bị tài liệu hướng dẫn sử dụng, viết báo cáo. | Đặng Kim Thành |
| Triển khai chatbot, làm thuyết trình cuối dự án. | Đặng Kim Thành |
| 16/08/2024 - 17/08/2024 | Đánh giá và hoàn thiện | Đặng Kim Thành |
| Báo cáo tổng kết dự án. | Đặng Kim Thành |

1. **Quy trình thực hiện dự án**
   1. **Thu thập dữ liệu**

Việc thu thập dữ liệu là một bước quan trọng trong quá trình xây dựng chatbot hỗ trợ cung cấp thông tin cho Trường Đại học Sư phạm Kỹ thuật TP.HCM. Để đảm bảo chatbot có thể trả lời các câu hỏi một cách chính xác và đầy đủ, tôi đã sử dụng các nguồn dữ liệu chính thống và có tính tin cậy cao.

* + 1. **Nguồn dữ liệu sử dụng**

Thư viện số của trường: Đây là nguồn dữ liệu chính cung cấp các tài liệu học thuật, giáo trình, và các tài liệu liên quan đến chương trình học. Thư viện số chứa đựng một lượng lớn thông tin về các môn học, yêu cầu tốt nghiệp, và các tài nguyên học tập khác mà sinh viên có thể cần đến. [[11](https://thuvienso.hcmute.edu.vn/)]

Văn bản quy chế của trường: Các văn bản quy chế bao gồm các quy định về tuyển sinh, học phí, kỷ luật, và các quy trình học vụ khác. Những văn bản này cung cấp cơ sở pháp lý và hướng dẫn cụ thể, đảm bảo rằng chatbot có thể cung cấp các thông tin chính xác về quy định của trường.[[10](https://hcmute.edu.vn/)]

Câu hỏi thường gặp (FAQ): FAQ là một nguồn thông tin quan trọng để nắm bắt các vấn đề và thắc mắc phổ biến của sinh viên. Việc thu thập các câu hỏi này giúp tôi xác định những câu hỏi nào thường gặp nhất và chuẩn bị sẵn các câu trả lời tự động cho chatbot.

* + 1. **Quá trình thu thập và xử lý cơ bản dữ liệu**

Xác định và lựa chọn nguồn dữ liệu: Tôi đã lựa chọn các nguồn dữ liệu từ thư viện số, văn bản quy chế của trường, và phần FAQ để xây dựng bộ dữ liệu đầu vào. Những nguồn này đảm bảo tính chính thống và sự đa dạng của thông tin.

Trích xuất và chuẩn bị dữ liệu: Các tài liệu từ thư viện số và văn bản quy chế được tải về và chuyển đổi sang định dạng văn bản để dễ dàng xử lý. Tương tự, các câu hỏi từ phần FAQ cũng được tổng hợp lại và phân loại theo các chủ đề khác nhau như học vụ, tài chính, và đời sống sinh viên.

Làm sạch dữ liệu: Các tài liệu sau khi được định dạng cơ bản theo cấu trúc cơ bản mà các mô hình ngôn ngữ lớn thường sử dùng sẽ được tập trung xử lý sâu vào cốt lõi: loại bỏ thông tin dư thừa, kết nối các câu bị xuống dòng, ghép các thông tin mang tính liên kết lại, trích xuất các loại dữ liệu khác nhau về cùng một kiểu.

* + 1. **Mục tiêu và kết quả**

Mục tiêu của quá trình thu thập dữ liệu là xây dựng một bộ dữ liệu toàn diện và chính xác, đáp ứng được nhu cầu thông tin của sinh viên và giảng viên. Dữ liệu này sẽ được sử dụng để huấn luyện chatbot, giúp chatbot có khả năng trả lời các câu hỏi về thông tin học tập, quy định của trường, và các vấn đề thường gặp một cách hiệu quả và chính xác.

**Kết quả đạt được:**

* Thu thập và xử lý dữ liệu từ nhiều nguồn khác nhau: Đã thu thập dữ liệu từ rất nhiều tài liệu PDF, trang web, và các tài liệu chữ của Trường Đại học Sư phạm Kỹ thuật TP.HCM.
* Kiểm tra và đánh giá được sơ bộ nội dung dữ liệu thu thập để đảm bảo chất lượng dữ liệu đáng tin cậy
* Phân tích và lựa chọn cơ sở dữ liệu để mã hóa và lưu trữ dữ liệu một cách hợp lý và dễ dàng kiểm soát, nắm bắt.
  1. **Thiết kế hệ thống**

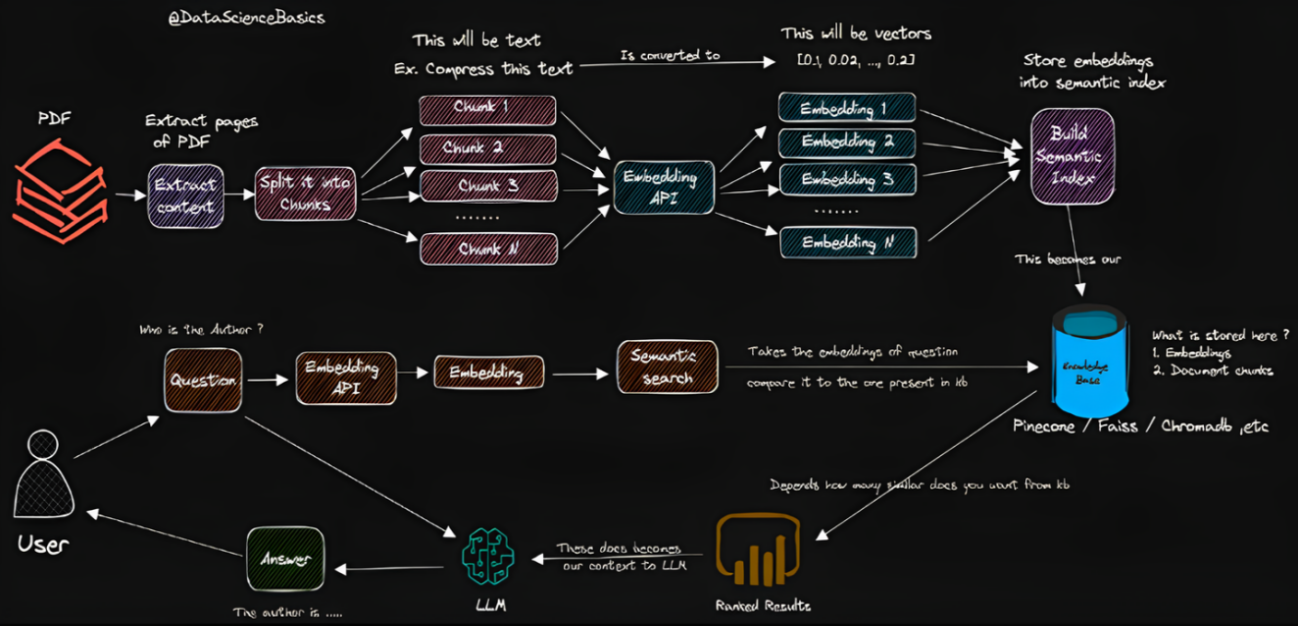
Trong quá trình xây dựng chatbot hỗ trợ tra cứu thông tin cho Trường Đại học Sư phạm Kỹ thuật TP.HCM, tôi đã lựa chọn các phương pháp và công cụ tiên tiến nhằm đảm bảo hiệu quả và độ chính xác cao cho hệ thống. Để tối ưu hóa khả năng xử lý ngôn ngữ tự nhiên và tăng cường trải nghiệm người dùng, hệ thống chatbot được thiết kế dựa trên quy trình Retrieval-Augmented Generation (RAG), sử dụng mô hình ngôn ngữ lớn Llama 2 - 7b và tích hợp với framework Langchain:

Quy trình RAG được áp dụng nhằm nâng cao khả năng tìm kiếm và truy xuất thông tin từ cơ sở dữ liệu lớn. Cụ thể, chatbot sẽ tiếp nhận câu hỏi từ người dùng, sau đó sử dụng bộ truy xuất thông tin (retriever) để tìm kiếm tài liệu liên quan trong cơ sở dữ liệu được thu thập trước đó. Cuối cùng, mô hình ngôn ngữ lớn (LLM) sẽ kết hợp câu hỏi và tài liệu tìm được để sinh ra câu trả lời phù hợp nhất với yêu cầu của người dùng.

Framework Langchain đóng vai trò quan trọng trong việc tích hợp các công cụ và phương pháp khác nhau, hỗ trợ xây dựng và triển khai mô hình ngôn ngữ lớn một cách hiệu quả. Langchain cho phép việc quản lý và xử lý các tài liệu, prompts, và các agent bên ngoài, giúp hệ thống hoạt động một cách trơn tru và hiệu quả hơn.

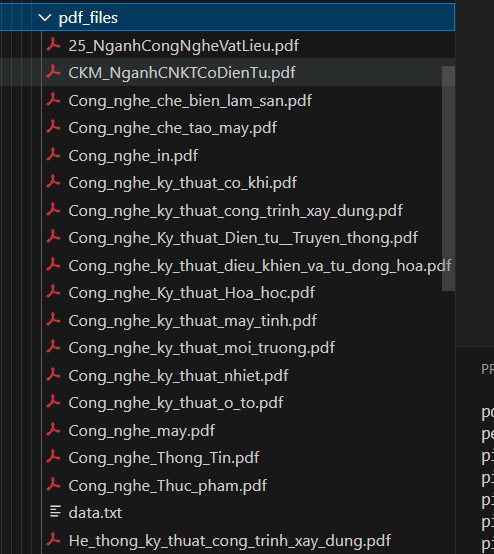
Mô hình Llama 2 - 7b được lựa chọn do đây là mô hình mã nguồn mở có khả năng mở rộng và linh hoạt cao, với dữ liệu huấn luyện lớn gần 70 nghìn tỷ token. Điều này giúp mô hình bao quát được nhiều thông tin từ các lĩnh vực khác nhau, cung cấp các phản hồi nhanh chóng và chính xác cho người dùng.

Với sự kết hợp của các thành phần trên, hệ thống chatbot được kỳ vọng sẽ mang lại trải nghiệm tra cứu thông tin nhanh chóng, chính xác, và dễ dàng cho sinh viên và nhân viên của trường:

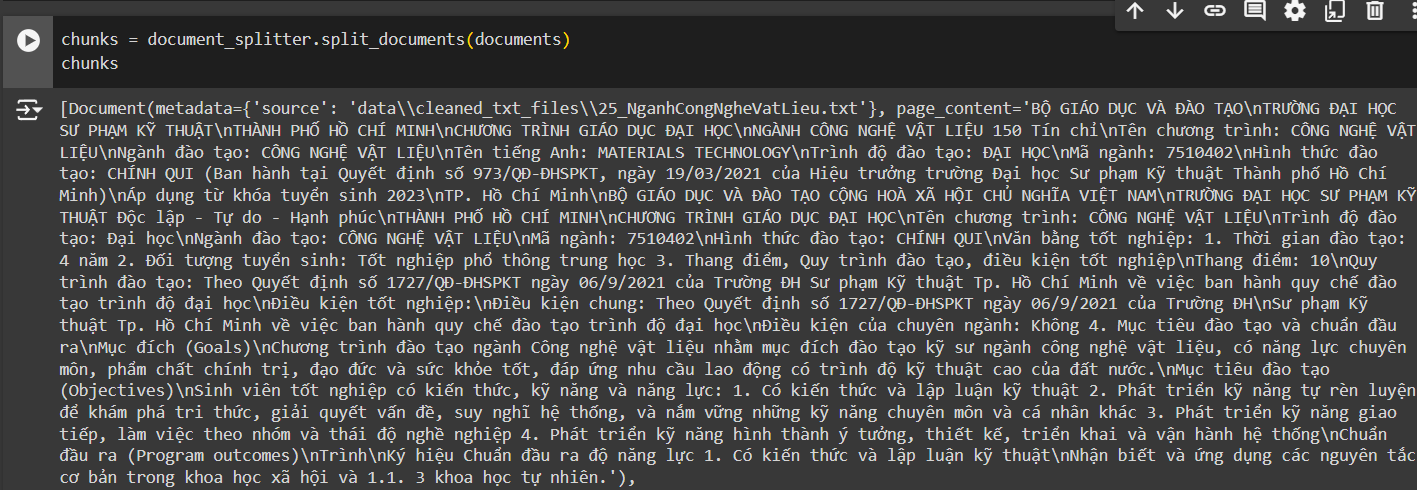


* Quy trình sẽ được thực hiện như sau:

Bước 1: Chuẩn bị các tài liệu tệp tin (PDF, Words,…).



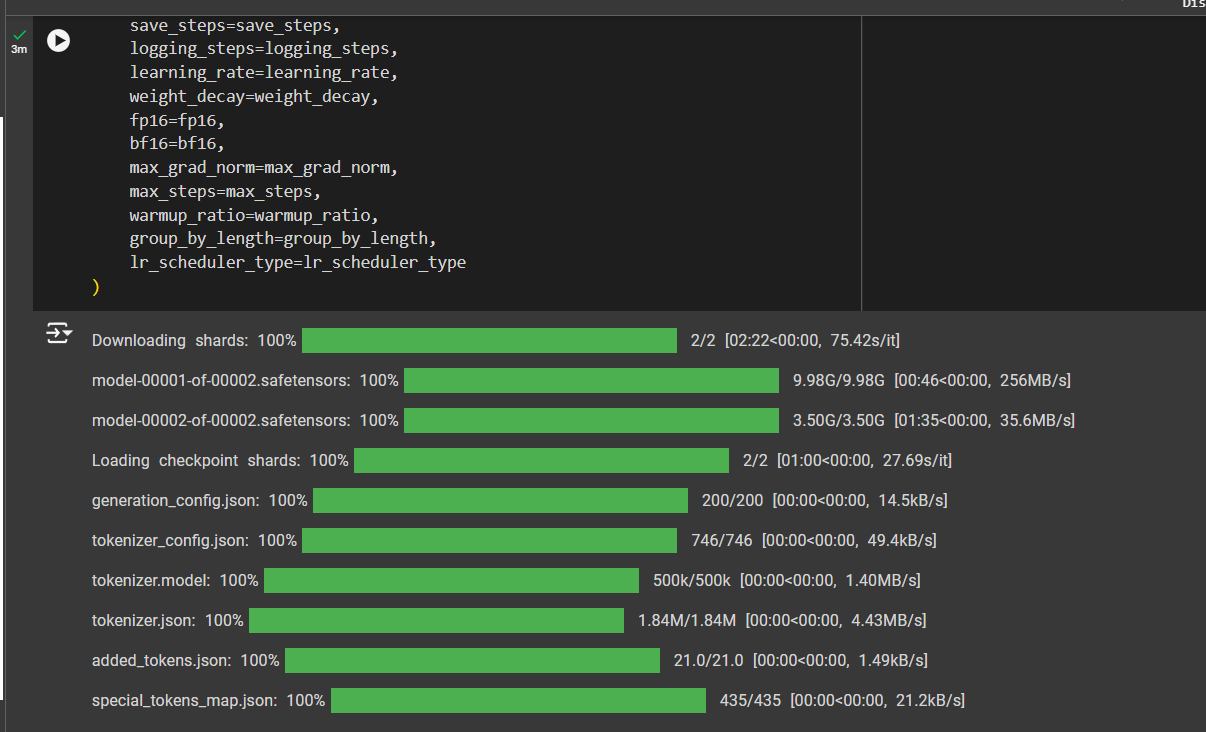
Bước 2: Thực hiện xử lý tài liệu, phân tách nội dung tài liệu và chia thành các khối nhỏ hơn.



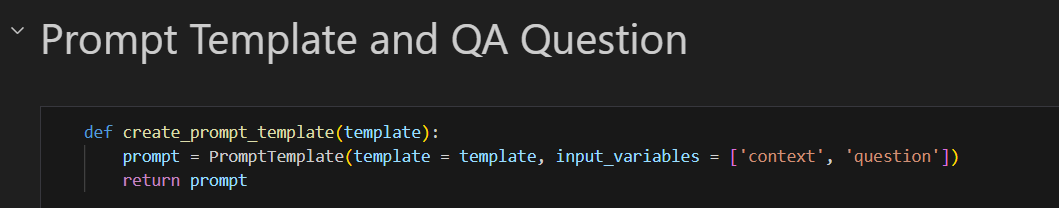
Bước 3: Thực hiện nhúng các khối nhỏ hơn vào trong cơ sở dữ liệu vector.



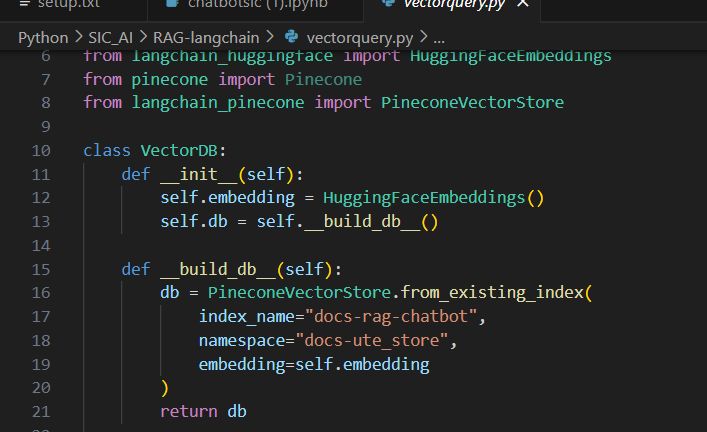
Bước 4: Thực hiện finetune mô hình llama2 với dữ liệu đã chuẩn bị.



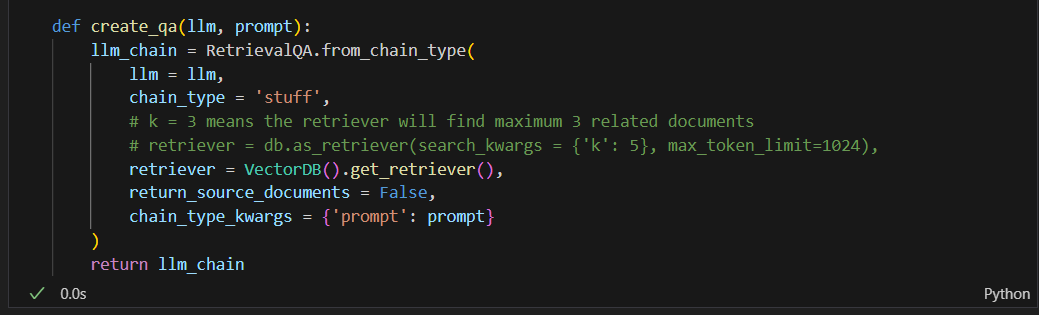
Bước 5: Khi người dùng đặt câu hỏi, nhúng câu hỏi thành các vector.



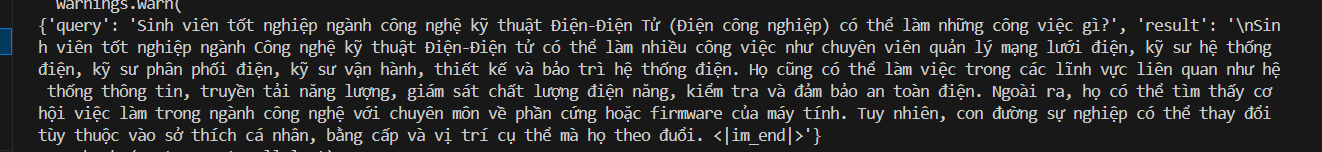
Bước 6: Thực hiện tìm kiếm so sánh với cơ sở dữ liệu đã lưu ở trên bước 3.



Bước 7: Thực hiện xếp hạng các kết quả tìm được chẳng hạn số lượng xếp hạng yêu cầu là 10 thì sẽ lấy số lượng là 10 kết quả gần nhất.



Bước 8: Sử dụng mô hình ngôn ngữ lớn (Large Language Model - LLM) đã được finetune để tạo ra câu trả lời và phản hồi lại người dùng (nếu không có câu trả lời trong tài liệu sẽ phản hồi là không tìm thấy).



* 1. **Phương pháp huấn luyện**

Quá trình huấn luyện mô hình chatbot hỗ trợ cung cấp thông tin cho Trường Đại học Sư phạm Kỹ thuật TP.HCM được thực hiện bằng cách kết hợp nhiều phương pháp tiên tiến trong xử lý ngôn ngữ tự nhiên (NLP). Mục tiêu là tạo ra một chatbot hiệu quả, chính xác và dễ dàng mở rộng.

* + 1. **Quy trình RAG (Retrieval-Augmented Generation)**

Để đảm bảo chatbot có khả năng trả lời các câu hỏi một cách chính xác và linh hoạt, tôi đã sử dụng phương pháp Retrieval-Augmented Generation (RAG). Phương pháp này kết hợp giữa việc truy xuất thông tin từ cơ sở dữ liệu và tạo ra câu trả lời tự động, giúp chatbot có thể cung cấp thông tin chi tiết từ các nguồn dữ liệu mà không cần phải ghi nhớ tất cả. Quy trình này kết hợp hai giai đoạn chính:

* Truy xuất thông tin (Retrieval): Chatbot đầu tiên sẽ tìm kiếm và truy xuất các tài liệu phù hợp với câu hỏi của người dùng từ cơ sở dữ liệu. Việc này được thực hiện thông qua các công cụ tìm kiếm nội dung được huấn luyện riêng trên các văn bản quy chế, tài liệu từ thư viện số, và FAQ.
* Tạo văn bản (Generation): Sau khi đã tìm thấy các tài liệu phù hợp, mô hình sẽ sử dụng các kỹ thuật tạo văn bản để xây dựng câu trả lời dựa trên nội dung được truy xuất. Phương pháp này đảm bảo rằng câu trả lời của chatbot vừa đúng đắn, vừa phù hợp với ngữ cảnh.

A diagram of a book

Description automatically generated

Đầu tiên, chương trình sẽ tiếp nhận câu hỏi từ prompt người dùng. Sau đó, phần retriever sẽ tìm trong cơ sở dữ liệu được thu thập từ trước để tìm ra tài liệu liên quan nhất đối với câu hỏi, sau đó mô hình LLM sẽ kết hợp câu hỏi và tài liệu tìm được để sinh ra câu trả lời lời phù hợp nhất với yêu cầu của người dùng.

* + 1. **Framework Langchain**

Để xây dựng quy trình trên, tôi sử dụng Langchain, một framework mạnh mẽ giúp kết hợp các mô hình ngôn ngữ và công cụ xử lý văn bản. Langchain cung cấp khả năng xây dựng và quản lý các quy trình xử lý ngôn ngữ phức tạp, cho phép tôi dễ dàng tích hợp các công cụ tìm kiếm, truy xuất dữ liệu, và các mô hình tạo văn bản vào trong hệ thống chatbot.

A diagram of a diagram of a system

Description automatically generated with medium confidence

* + - 1. **Các thành phần trong Langchain:**
  + **Các loại mô hình trong LangChain:**

Mô hình ngôn ngữ lớn: Những mô hình ngôn ngữ lớn (Large Language Model - LLM) như GPT-3.5 (nền tảng của ChatGPT), GPT-2, GPT-3,... Chức năng chính là dữ liệu đầu vào của LLM là một chuỗi văn bản (câu hỏi, yêu cầu) và dữ liệu đầu ra cũng là một chuỗi văn bản tương ứng với yêu cầu đầu vào. Thực hiện nhiều tác vụ xử lý ngôn ngữ tự nhiên như tạo văn bản, dịch ngôn ngữ, viết nội dung sáng tạo,...

Mô hình trò chuyện: Mô hình trò chuyện (chat model) dựa trên trên các LLM tuy nhiên nó sẽ gắn với các ngữ cảnh trong Chatbot (như ChatGPT). Chat model thì dữ liệu đầu vào là văn bản hoặc tin nhắn trò chuyện (chat message) từ người dùng, AI hoặc hệ thống. Các chat message được tập hợp thành một đoạn hội thoại thì sẽ được đưa vào Chat Model.

Mô hình nhúng văn bản: Mô hình nhúng văn bản (text embedding model) đại diện cho một đoạn văn bản, được biểu diễn bằng một véc tơ có số chiều cố định. Hai véc tơ càng gần nhau thì mức độ tương đồng ngữ nghĩa/nội dung giữa các đoạn văn bản càng cao. Langchain cho phép kết nối đến các loại mô hình sinh embedding khác nhau (OpenAI, Hugging Face, Cohere,…).

* + **Lời nhắc (Prompts)**

Trong hệ thống chatbot sử dụng LangChain, Lời nhắc (Prompts) đóng vai trò quan trọng trong việc định hướng câu trả lời của mô hình ngôn ngữ (LLM). Lời nhắc là văn bản nhập vào được cung cấp cho mô hình AI để tạo ra phản hồi. Chúng bao gồm một câu hỏi, câu lệnh, hoặc bất kỳ thông tin nào mà mô hình cần để tạo ra câu trả lời phù hợp. Việc thiết kế lời nhắc hiệu quả có thể cải thiện chất lượng và độ chính xác của câu trả lời từ mô hình, đảm bảo rằng hệ thống chatbot cung cấp thông tin hữu ích và đúng trọng tâm.

**Mục đích của lời nhắc trong LangChain:**

Hướng dẫn mô hình: Lời nhắc được sử dụng để cung cấp ngữ cảnh hoặc hướng dẫn cụ thể mà mô hình cần tuân theo khi tạo ra phản hồi. Điều này rất quan trọng trong việc đảm bảo rằng câu trả lời không chỉ đúng về mặt thông tin mà còn phù hợp với ngữ cảnh sử dụng.

Tối ưu hóa khả năng phản hồi: Bằng cách thiết kế các lời nhắc tốt, chúng ta có thể tận dụng tối đa khả năng của mô hình ngôn ngữ, giúp nó hiểu rõ hơn về những gì cần trả lời và làm cách nào để trả lời một cách hiệu quả nhất.

Tùy chỉnh phản hồi cho các ngữ cảnh cụ thể: Lời nhắc có thể được điều chỉnh để phù hợp với ngữ cảnh cụ thể, ví dụ như khi trả lời các câu hỏi về chính sách trường học, thông tin khóa học, hoặc các quy định của trường. Điều này giúp chatbot có thể cung cấp các câu trả lời chính xác và phù hợp hơn cho người dùng.

**Cách thiết kế lời nhắc hiệu quả trong LangChain:**

Rõ ràng và ngắn gọn: Lời nhắc cần phải rõ ràng và ngắn gọn để tránh gây nhầm lẫn cho mô hình. Tránh sử dụng ngôn ngữ phức tạp hoặc câu hỏi mơ hồ.

Cung cấp đủ ngữ cảnh: Đôi khi, chỉ một câu hỏi đơn giản không đủ để mô hình tạo ra một phản hồi chính xác. Do đó, cung cấp thêm ngữ cảnh có thể giúp cải thiện chất lượng câu trả lời.

Sử dụng ví dụ cụ thể: Đưa ra các ví dụ cụ thể trong lời nhắc có thể giúp mô hình hiểu rõ hơn về cách thức mà nó nên trả lời.

Điều chỉnh theo phản hồi: Trong quá trình thử nghiệm và triển khai, các lời nhắc có thể được điều chỉnh dựa trên phản hồi của người dùng hoặc kết quả đầu ra của mô hình để liên tục cải thiện chất lượng phản hồi.

* + **Chuỗi (Chains)**

Chuỗi (Chains) là một thành phần quan trọng trong LangChain, cho phép kết hợp nhiều mô hình và công cụ lại với nhau để thực hiện các tác vụ phức tạp hơn. Trong hệ thống chatbot, chuỗi được sử dụng để tạo ra một luồng xử lý liên tục, từ việc nhận đầu vào của người dùng đến việc xử lý thông tin và cung cấp phản hồi.

**Mục đích của Chuỗi trong LangChain**

Tạo quy trình xử lý nhiều bước: Chuỗi cho phép hệ thống xử lý các yêu cầu từ người dùng qua nhiều bước logic. Ví dụ, một yêu cầu của người dùng có thể cần truy vấn thông tin từ cơ sở dữ liệu, sau đó xử lý và định dạng lại dữ liệu trước khi trả về kết quả.

Kết hợp các mô hình khác nhau: Chuỗi có thể kết hợp nhiều mô hình khác nhau trong LangChain, chẳng hạn như mô hình ngôn ngữ để tạo phản hồi văn bản và mô hình phân tích cảm xúc để hiểu được tâm trạng của người dùng.

Tối ưu hóa việc sử dụng tài nguyên: Sử dụng chuỗi giúp tận dụng tối đa các khả năng của nhiều mô hình khác nhau trong một quy trình duy nhất, từ đó tối ưu hóa hiệu quả và tài nguyên của hệ thống.

**Các loại Chuỗi phổ biến trong LangChain**

Simple Chains (Chuỗi đơn giản): Chuỗi này gồm một hoặc hai bước đơn giản, như nhận đầu vào từ người dùng và trả lời bằng phản hồi từ mô hình ngôn ngữ. Ví dụ, một câu hỏi đơn giản như "Thời gian mở cửa của thư viện trường là khi nào?" có thể được xử lý bởi một chuỗi đơn giản, nơi mà mô hình ngôn ngữ sẽ truy vấn cơ sở dữ liệu và trả lời.

Sequential Chains (Chuỗi tuần tự): Đây là loại chuỗi mà các bước được thực hiện theo thứ tự nhất định. Một ví dụ là chuỗi xử lý đầu vào người dùng, sau đó thực hiện phân tích, tiếp theo là tìm kiếm thông tin trong cơ sở dữ liệu, và cuối cùng là tổng hợp kết quả để trả lời người dùng.

Conditional Chains (Chuỗi điều kiện): Loại chuỗi này cho phép thực hiện các bước xử lý dựa trên điều kiện cụ thể. Ví dụ, nếu người dùng hỏi một câu hỏi kỹ thuật, chuỗi có thể điều hướng đến một mô hình ngôn ngữ chuyên về thông tin kỹ thuật; nếu câu hỏi liên quan đến quy định trường học, chuỗi sẽ chuyển hướng đến một mô hình chuyên về các quy định đó.

Parallel Chains (Chuỗi song song): Cho phép thực hiện nhiều bước xử lý đồng thời. Ví dụ, một chuỗi có thể vừa xử lý đầu vào người dùng vừa thực hiện một tác vụ khác như phân tích cảm xúc hoặc dự đoán ý định của người dùng.

* **Tác nhân (Agents)**

Agents (Tác nhân) trong LangChain là các thành phần đóng vai trò như những tác nhân thông minh có khả năng quyết định hành động dựa trên các đầu vào khác nhau. Tác nhân không chỉ dừng lại ở việc xử lý các yêu cầu của người dùng một cách trực tiếp mà còn có khả năng xác định các bước cần thiết và lựa chọn công cụ phù hợp để hoàn thành nhiệm vụ đó. Điều này giúp tạo ra một hệ thống chatbot linh hoạt và mạnh mẽ, có thể tự động hóa và xử lý các tác vụ phức tạp hơn.

**Mục đích và vai trò của Tác nhân trong LangChain**

Ra quyết định động (Dynamic Decision-Making): Tác nhân có khả năng phân tích đầu vào của người dùng và quyết định hành động tiếp theo một cách tự động. Điều này có thể bao gồm việc truy xuất thông tin từ cơ sở dữ liệu, thực hiện tính toán, hoặc thậm chí thực hiện một chuỗi các tác vụ để cung cấp câu trả lời phù hợp.

Tích hợp với các công cụ khác nhau: Một trong những lợi ích lớn nhất của tác nhân trong LangChain là khả năng tích hợp với nhiều công cụ khác nhau. Tác nhân có thể chọn sử dụng một mô hình ngôn ngữ để xử lý văn bản, một công cụ tìm kiếm để truy vấn thông tin, hoặc một dịch vụ web API để lấy dữ liệu.

Tối ưu hóa quy trình tương tác với người dùng: Thay vì chỉ trả lời trực tiếp câu hỏi của người dùng, tác nhân có thể thực hiện các hành động phức tạp hơn như chuyển hướng câu hỏi, xác định các bước tiếp theo hoặc thậm chí đặt câu hỏi lại cho người dùng để làm rõ yêu cầu.

**Các loại Tác nhân phổ biến trong LangChain**

Action-Based Agents (Tác nhân dựa trên hành động): Loại tác nhân này dựa trên một danh sách các hành động có thể thực hiện, và chọn hành động phù hợp nhất dựa trên đầu vào của người dùng. Ví dụ, một tác nhân có thể quyết định "truy vấn cơ sở dữ liệu" nếu người dùng yêu cầu thông tin về các khóa học, hoặc "sử dụng mô hình ngôn ngữ" nếu cần tạo một phản hồi tự nhiên hơn.

Tool-Using Agents (Tác nhân sử dụng công cụ): Tác nhân này có khả năng gọi và sử dụng các công cụ bên ngoài để thực hiện các nhiệm vụ cụ thể. Chẳng hạn, nó có thể sử dụng một API để lấy thông tin thời tiết hiện tại hoặc một công cụ phân tích để đánh giá dữ liệu. Trong hệ thống chatbot của trường, tác nhân có thể sử dụng công cụ để truy xuất thông tin từ hệ thống quản lý học tập (LMS) hoặc cơ sở dữ liệu sinh viên.

Reactive Agents (Tác nhân phản ứng): Đây là loại tác nhân hoạt động dựa trên một loạt các quy tắc xác định trước để phản ứng lại với các đầu vào của người dùng. Tác nhân này không có khả năng lập kế hoạch dài hạn mà chỉ thực hiện các phản hồi tức thì dựa trên trạng thái hiện tại. Trong bối cảnh chatbot, tác nhân này có thể được sử dụng cho các yêu cầu đơn giản và thường xuyên gặp như “Giờ làm việc của thư viện” hoặc “Hướng dẫn đăng ký môn học”.

Proactive Agents (Tác nhân chủ động): Không giống như tác nhân phản ứng, tác nhân chủ động có khả năng lập kế hoạch và dự đoán các nhu cầu của người dùng. Nó có thể chủ động cung cấp thông tin liên quan hoặc đề xuất các hành động tiếp theo dựa trên lịch sử tương tác của người dùng. Ví dụ, nếu một sinh viên thường xuyên hỏi về các lớp học cụ thể, tác nhân chủ động có thể cung cấp lịch trình hoặc nhắc nhở về hạn chót đăng ký.

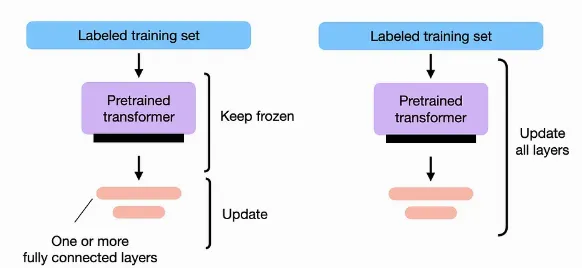
Tổng kết lại thì trong LangChain, quy trình hoạt động cơ bản bắt đầu bằng việc nhận đầu vào từ người dùng qua giao diện tương tác. Hệ thống sẽ sử dụng Prompts để xây dựng câu hỏi và xác định ngữ cảnh phù hợp cho mô hình ngôn ngữ, đảm bảo phản hồi chính xác và liên quan đến chủ đề được yêu cầu. Tiếp theo, các Chains sẽ được sử dụng để xử lý và phân tích đầu vào theo từng bước hoặc chuỗi hành động được định sẵn, giúp phân tích thông tin một cách chi tiết và có hệ thống. Sau đó, các Agents (tác nhân) sẽ được kích hoạt để lựa chọn công cụ hoặc hành động phù hợp, tự động hóa các quy trình phức tạp và đưa ra quyết định động dựa trên dữ liệu đã được xử lý. Cuối cùng, kết quả sẽ được trả về cho người dùng dưới dạng phản hồi hoặc thông tin, đảm bảo tính chính xác và hiệu quả của thông tin cung cấp. Quy trình này giúp hệ thống chatbot trở nên thông minh, linh hoạt, và có khả năng tương tác hiệu quả hơn với người dùng.

* + 1. **Fine-tuning bằng PEFT và LoRA**

Để tối ưu hóa mô hình ngôn ngữ cho các nhiệm vụ cụ thể trong chatbot, tôi đã thực hiện fine-tuning mô hình bằng các kỹ thuật Parameter-Efficient Fine-Tuning (PEFT) và Low-Rank Adaptation (LoRA). Các kỹ thuật này giúp điều chỉnh mô hình đã được huấn luyện trước đó trên các tập dữ liệu lớn, giúp mô hình thích nghi với dữ liệu cụ thể của trường mà không cần phải huấn luyện lại toàn bộ.

PEFT (Parameter-Efficient Fine-Tuning): Đây là phương pháp giúp giảm số lượng tham số cần phải huấn luyện lại, nhờ đó tiết kiệm tài nguyên và thời gian trong quá trình fine-tuning.

LoRA (Low-Rank Adaptation): LoRA là một phương pháp tối ưu hóa giúp mô hình học được các đặc trưng mới với một số lượng nhỏ tham số, giúp cải thiện hiệu suất của mô hình trong các nhiệm vụ cụ thể mà không làm giảm hiệu quả tổng thể.



Một số lớp đầu vẫn sẽ được đóng băng nhằm lưu giữ trọng số mô hình gốc vì ở đây chỉ đảm nhiệm các nhiệm vụ lấy đặc trưng cơ bản của việc xử lý ngôn ngữ. Còn các lớp sau sẽ được huấn luyện lại nhằm mục đích phục vụ cho dữ liệu đưa vào mô hình có kết quả tốt nhất.

* + 1. **Kiểm thử và triển khai**

Sau khi huấn luyện và tối ưu hóa, mô hình được kiểm thử kỹ lưỡng trên tập dữ liệu kiểm tra để đánh giá hiệu suất. Quá trình này bao gồm việc đánh giá các chỉ số như độ chính xác, độ nhạy, và độ đặc hiệu của chatbot.

Cuối cùng, mô hình chatbot được triển khai trên nền tảng trực tuyến của trường, và sẵn sàng hỗ trợ sinh viên và giảng viên. Chatbot cũng được theo dõi và cập nhật định kỳ để đảm bảo nó luôn cung cấp thông tin chính xác và kịp thời.

* 1. **Quy trình công việc**

Tiếp theo dự án sẽ tiếp tục đi theo từng bước xây dựng dự án từ việc chọn mô hình, mã hóa dữ liệu, đưa dữ liệu lên đám mây và truy xuất dữ liệu phục vụ cho việc tạo ngẫu nhiên kết quả.

* + 1. **Mô hình ngôn ngữ lớn (LLM)**

Trong lĩnh vực mô hình ngôn ngữ lớn (LLM), có nhiều mô hình nổi bật với các đặc điểm và ứng dụng khác nhau. Dưới đây là một số mô hình nổi tiếng và sự so sánh giữa chúng:

A white sheet with black text

Description automatically generated

A table with numbers and stars

Description automatically generated

Trong các mô hình trên, ta nhận thấy rằng llama2 là mô hình tốt được khai thác miễn phí và không cần cam kết điều khoản:

Mã nguồn mở và dễ tiếp cận: LLaMA2 là một mô hình mã nguồn mở được Meta phát triển và công khai, cho phép các nhà nghiên cứu và nhà phát triển có thể truy cập và điều chỉnh mô hình theo nhu cầu của mình. Điều này giúp giảm chi phí và thời gian phát triển so với các mô hình mã nguồn đóng như GPT-4.

Hiệu suất vượt trội trên nhiều ngôn ngữ: LLaMA2 được huấn luyện trên một tập dữ liệu đa ngôn ngữ lớn, bao gồm cả tiếng Việt. Các nghiên cứu đã chỉ ra rằng mô hình này thể hiện hiệu suất tốt trong các nhiệm vụ xử lý ngôn ngữ tự nhiên (NLP) cho nhiều ngôn ngữ, đặc biệt là trong các tác vụ như tạo văn bản, phân tích cảm xúc, và dịch ngôn ngữ.

Được huấn luyện với dữ liệu tiếng Việt: LLaMA2 đã được huấn luyện trên tập dữ liệu bao gồm các văn bản bằng tiếng Việt, điều này làm cho mô hình có khả năng xử lý và hiểu ngôn ngữ Việt tốt hơn. Ví dụ, các bài báo và nghiên cứu về LLaMA2 cho thấy mô hình có khả năng tạo ra văn bản chất lượng cao và chính xác trong tiếng Việt, cũng như hiệu quả trong các tác vụ NLP liên quan đến tiếng Việt.

* + 1. **Mô hình embedding**

Trong các mô hình embedding được tôi nghiên cứu sơ qua thì tôi quyết định sử dụng model embedding của hunggingface thay vì model gpt4allembedding do:

Khả năng tinh chỉnh và tùy chỉnh cao: Hugging Face cung cấp khả năng tùy chỉnh cao với các mô hình pre-trained, giúp dễ dàng điều chỉnh cho các nhiệm vụ và ngữ cảnh cụ thể. Điều này đặc biệt hữu ích cho các ứng dụng cần tùy chỉnh mô hình để phù hợp với dữ liệu và yêu cầu riêng.

Cộng đồng và hỗ trợ: Hugging Face có một cộng đồng phát triển mạnh mẽ và hỗ trợ tốt từ các nhà nghiên cứu và nhà phát triển. Điều này giúp dễ dàng tìm kiếm tài liệu, hướng dẫn, và giải pháp cho các vấn đề liên quan.

Tính đa dạng và hiệu suất: Các mô hình của Hugging Face bao gồm nhiều lựa chọn và biến thể của các mô hình tiên tiến, đảm bảo hiệu suất cao và tính linh hoạt trong các ứng dụng NLP. Các mô hình này đã được huấn luyện trên nhiều dữ liệu và có khả năng hoạt động tốt trong nhiều tình huống khác nhau.

Tài nguyên và công cụ: Thư viện transformers của Hugging Face cung cấp các công cụ và tài nguyên hỗ trợ việc triển khai và tích hợp các mô hình embedding vào các ứng dụng dễ dàng và hiệu quả.



* + 1. **Cơ sở dữ liệu vector hóa Pinecone**

Trong quá trình phát triển dự án, hai cơ sở dữ liệu vector hóa đã được sử dụng là faiss và pinecone. Tuy nhiên cở sở dữ liệu faiss của facebook lại gặp các khó khăn sau:

Khó cấu hình và quản lý: Faiss yêu cầu cấu hình phức tạp và quản lý tài nguyên cho các hoạt động như lập chỉ mục và tìm kiếm. Điều này có thể tạo ra một đường cong học tập cao và yêu cầu hiểu biết sâu về các tham số và cấu hình.

Khả năng mở rộng và quản lý: Faiss chủ yếu được thiết kế để hoạt động trên máy tính cá nhân hoặc cụm máy chủ riêng. Việc mở rộng quy mô cho các ứng dụng lớn hơn có thể đụng phải những thách thức liên quan đến phân tán dữ liệu và hiệu suất.

Thiếu tính năng tích hợp: Faiss không cung cấp các tính năng tích hợp sẵn cho các dịch vụ như quản lý dữ liệu, phân tích hoặc bảo mật. Các tính năng này cần được phát triển và triển khai riêng biệt.

Yêu cầu tài nguyên tính toán cao: Đối với các tập dữ liệu lớn hoặc các yêu cầu tìm kiếm phức tạp, có thể yêu cầu tài nguyên tính toán đáng kể, điều này có thể ảnh hưởng đến hiệu suất và chi phí.

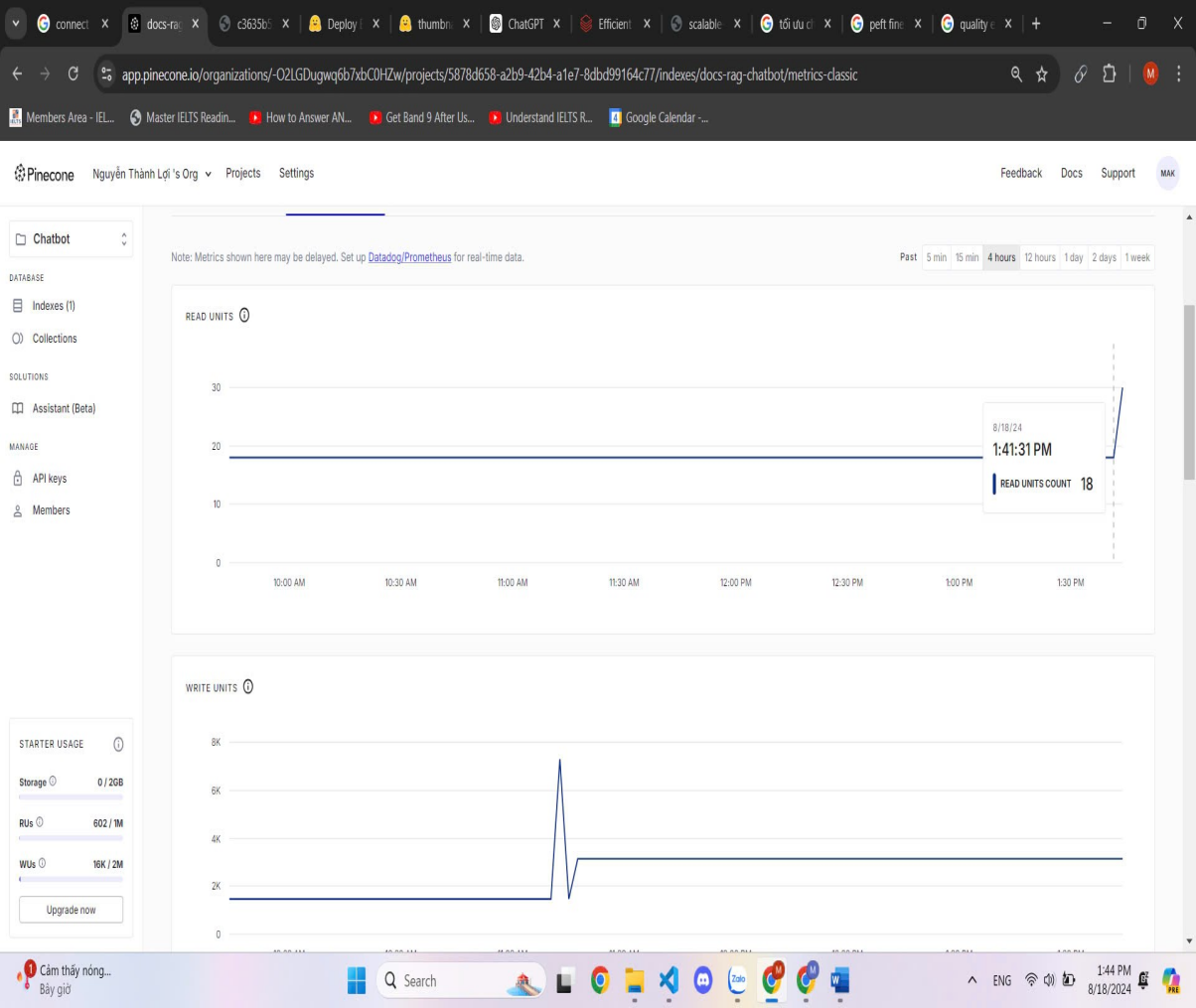
* Khiến cho cơ sở dữ liệu Pinecone là mô hình hợp nhất với dự án:

Dễ dàng triển khai và quản lý: Pinecone cung cấp dịch vụ hoàn chỉnh với khả năng mở rộng tự động, giúp giảm bớt công việc quản lý và cấu hình.

Tính năng tích hợp: Cung cấp các tính năng tích hợp sẵn như lập chỉ mục, tìm kiếm và phân tích, giúp tiết kiệm thời gian và công sức phát triển.

Khả năng mở rộng linh hoạt: Pinecone có khả năng mở rộng linh hoạt để đáp ứng nhu cầu của các ứng dụng lớn hoặc các yêu cầu tìm kiếm phức tạp.

Hỗ trợ và bảo mật: Được cung cấp bởi một nhà cung cấp dịch vụ, Pinecone thường có hỗ trợ và bảo mật chuyên nghiệp, giúp đảm bảo rằng hệ thống của bạn hoạt động ổn định và an toàn.



1. **Kết quả**
   1. **Tiền xử lý dữ liệu**

Trong quá trình phát triển chatbot hỗ trợ cung cấp thông tin cho Trường Đại học Sư phạm Kỹ thuật TP.HCM, việc thu thập và xử lý dữ liệu là một bước quan trọng để đảm bảo chất lượng và tính chính xác của thông tin mà chatbot cung cấp. Dữ liệu sử dụng trong dự án này chủ yếu được lấy từ các tệp PDF chứa thông tin về các khóa học, quy định, và các dịch vụ khác của trường.

* + 1. **Quy trình thu thập và xử lý dữ liệu**

Để chuẩn bị dữ liệu đầu vào cho chatbot, tôi đã thực hiện các bước sau:

* **Chuyển đổi tệp PDF sang văn bản (Text):**

Đầu tiên, các tệp PDF được chuyển đổi sang tệp văn bản (.txt) sử dụng thư viện pdfplumber. Mỗi trang trong tệp PDF được trích xuất văn bản, bao gồm cả dữ liệu bảng Các bảng trong tài liệu PDF được xử lý riêng biệt để đảm bảo dữ liệu trong các bảng này không bị mất hoặc sai lệch. Từng hàng trong bảng được nối lại thành một dòng văn bản và được thêm vào văn bản trang.

Sau khi trích xuất văn bản từ PDF, các số trang không cần thiết được loại bỏ để làm sạch dữ liệu.

* **Xử lý các tệp văn bản đã chuyển đổi (làm sạch và tổ chức lại dữ liệu):**

Văn bản được chuyển đổi từ PDF thường có các khoảng trắng không cần thiết, hoặc các dòng văn bản bị ngắt đoạn không đúng vị trí. Tôi đã phát triển các hàm để làm sạch và hợp nhất các dòng văn bản, loại bỏ các ký tự không mong muốn và tổ chức lại dữ liệu bảng trong văn bản.

Dữ liệu sau khi làm sạch sẽ được lưu lại vào các tệp văn bản mới, sẵn sàng cho quá trình xử lý tiếp theo.

* **Xử lý dữ liệu hàng loạt:**

Để tiện cho quá trình thu thập dữ liệu từ nhiều tệp PDF, tôi đã tạo một hàm xử lý hàng loạt. Hàm này tự động chuyển đổi tất cả các tệp PDF trong một thư mục sang văn bản, sau đó làm sạch dữ liệu và lưu lại kết quả cuối cùng.

* **Xử lý và chia nhỏ văn bản:**

Sử dụng RecursiveCharacterTextSplitter để chia nhỏ văn bản thành các đoạn nhỏ hơn. Các đoạn văn bản được chia nhỏ dựa trên các ký tự phân tách và kích thước đoạn, giúp đảm bảo rằng dữ liệu có thể được xử lý hiệu quả hơn trong các mô hình học máy.

Các tham số cấu hình bao gồm separators, chunk\_size, và chunk\_overlap để tối ưu hóa quá trình chia nhỏ văn bản.

Sử dụng mô hình embedding từ Hugging Face để tạo embedding cho các đoạn văn bản đã chia nhỏ.

Tạo cơ sở dữ liệu vector sử dụng Pinecone để lưu trữ các embedding và cho phép truy vấn hiệu quả. Thư viện Pinecone được sử dụng để kết nối và quản lý dữ liệu trong vector store.

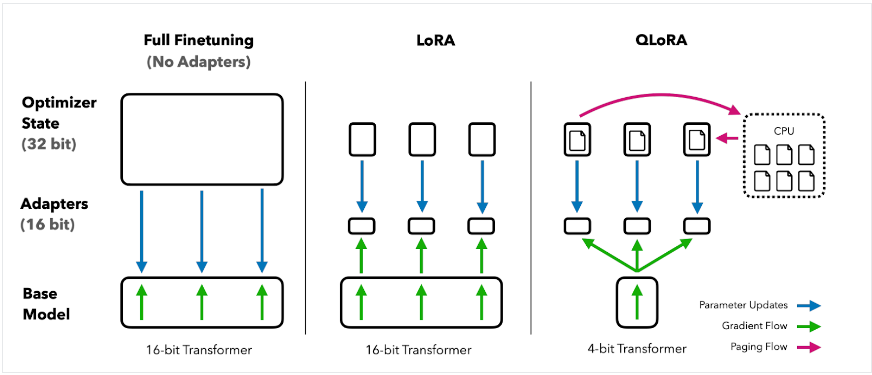
* Mục tiêu và kết quả: Mục tiêu chính của quá trình thu thập dữ liệu này là tạo ra một bộ dữ liệu văn bản sạch, có cấu trúc và sẵn sàng cho việc huấn luyện chatbot. Các tệp văn bản sau khi được xử lý sẽ là nguồn dữ liệu chính để xây dựng các tập huấn luyện cho mô hình ngôn ngữ mà chatbot sẽ sử dụng. Quá trình này đảm bảo rằng dữ liệu đầu vào không chỉ đầy đủ và chính xác mà còn được tối ưu hóa để mô hình có thể học tập và đưa ra các phản hồi chính xác và hiệu quả nhất.
  1. **Thiết lập mô hình**

Trong giai đoạn này, mục tiêu là fine-tune mô hình ngôn ngữ LLaMA-2 với dữ liệu đã được chuyển đổi và xử lý để chatbot có thể cung cấp thông tin chính xác và phù hợp cho người dùng. Quy trình này bao gồm các bước cài đặt thư viện cần thiết, tải và chuẩn bị dữ liệu, cấu hình mô hình, huấn luyện mô hình, và lưu trữ mô hình đã fine-tune.

Đầu tiên, tôi cài đặt các thư viện cần thiết bao gồm accelerate, peft, bitsandbytes, transformers, và trl. Các thư viện này cung cấp công cụ và cấu hình cần thiết để thực hiện quá trình fine-tuning với hiệu suất tối ưu.

Tiếp theo, dữ liệu đầu vào được tải từ tệp văn bản và được chuyển đổi thành định dạng phù hợp với mô hình. Dữ liệu được định dạng lại thành các đoạn hội thoại giữa người dùng và trợ lý, với các thẻ đặc biệt được thêm vào để mô hình hiểu đúng cấu trúc hội thoại. Việc chuyển đổi này giúp mô hình học cách phản hồi một cách tự nhiên và chính xác hơn.

Sau khi dữ liệu đã được chuyển đổi, tôi tải mô hình ngôn ngữ LLaMA-2 và tokenizer với cấu hình BitsAndBytes để hỗ trợ quantization, cho phép mô hình hoạt động hiệu quả trên các tài nguyên phần cứng hạn chế.



Lượng tử hóa 4-bit thông qua QLoRA cho phép tinh chỉnh hiệu quả các mô hình LLM khổng lồ trên phần cứng tiêu dùng trong khi vẫn giữ được hiệu suất cao. QLoRA định lượng một mô hình ngôn ngữ được đào tạo trước thành 4 bit và đóng băng các tham số. Trong quá trình tinh chỉnh, gradient được truyền ngược thông qua mô hình lượng tử hóa 4 bit đông lạnh thành chỉ các lớp Low-Rank Adapter. Vì vậy, toàn bộ mô hình được đào tạo trước vẫn cố định ở 4 bit trong khi chỉ có các bộ điều hợp được cập nhật và không ảnh hưởng đến hiệu suất mô hình.

Việc tinh chỉnh truyền thống các mô hình ngôn ngữ được đào tạo trước (PLM) yêu cầu cập nhật tất cả các tham số của mô hình, tốn kém về mặt tính toán và đòi hỏi lượng dữ liệu lớn thì Parameter-Efficient Fine-Tuning (PEFT) sẽ hoạt động bằng cách chỉ cập nhật một tập hợp con nhỏ các tham số có ảnh hưởng nhất của mô hình, làm cho nó hiệu quả hơn nhiều. Sau đó mô hình tiếp tục được cấu hình với các tham số LoRA để fine-tune hiệu quả, bao gồm các tham số như lora\_alpha, lora\_dropout, và lora\_r. Tham số huấn luyện được thiết lập để tối ưu hóa quá trình fine-tuning. Các tham số này bao gồm số epoch, kích thước batch, tốc độ học, và các tham số khác. Việc cấu hình đúng các tham số giúp tăng cường khả năng của mô hình mà không gặp phải các vấn đề về hiệu suất.

Danh sách các siêu tham số có thể được sử dụng để tối ưu hóa quá trình đào tạo:

* output\_dir: Thư mục đầu ra là nơi lưu trữ các dự đoán mô hình và điểm kiểm tra.
* num\_train\_epochs: Số lần đào tạo.
* FP16/BF16: Vô hiệu hóa đào tạo fp16 / bf16.
* per\_device\_train\_batch\_size: Kích thước tập huấn luyện trên GPU mỗi lần chạy.
* per\_device\_eval\_batch\_size: Kích thước tập đánh giá trên GPU mỗi lần chạy.
* gradient\_accumulation\_steps: Điều này đề cập đến số bước cần thiết để tích lũy gradient trong quá trình cập nhật.
* gradient\_checkpointing: Bật điểm kiểm tra gradient.
* max\_grad\_norm: Cắt gradient.
* learning\_rate: Tỷ lệ học tập ban đầu.
* weight\_decay: Phân rã trọng lượng được áp dụng cho tất cả các lớp ngoại trừ trọng lượng thiên vị / LayerNorm.
* Tối ưu: Trình tối ưu hóa mô hình (trình tối ưu hóa AdamW).
* lr\_scheduler\_type: Lịch trình tỷ lệ học tập.
* max\_steps: Số bước đào tạo.
* warmup\_ratio: Tỷ lệ các bước để khởi động tuyến tính.
* group\_by\_length: Điều này có thể cải thiện đáng kể hiệu suất và đẩy nhanh quá trình đào tạo.
* Cuối cùng là tinh chỉnh có giám sát (SFT). Là một bước quan trọng trong việc học tăng cường từ phản hồi của con người (RLHF), nó đào tạo các mô hình ngôn ngữ bằng cách sử dụng học tăng cường, bắt đầu với tinh chỉnh có giám sát, sau đó là mô hình phần thưởng và cuối cùng là tối ưu hóa chính sách gần (PPO).
  1. **Kiểm thử và cải tiến**
     1. **Quy trình kiểm thử**

Trong giai đoạn kiểm thử của quá trình phát triển chatbot, tôi tập trung vào việc đánh giá hiệu suất và độ chính xác của chatbot trong các tình huống khác nhau. Mục tiêu chính là kiểm tra khả năng của chatbot trong việc truy xuất và tạo ra các phản hồi dựa trên các câu hỏi từ người dùng, khả năng xử lý các loại đầu vào khác nhau, và hiệu suất của chatbot dưới các điều kiện tải khác nhau.

Tôi sử dụng framework LangChain để cấu trúc tương tác giữa mô hình ngôn ngữ và hệ thống truy xuất thông tin. Cụ thể, tôi đã kiểm tra khả năng của chatbot trong việc xử lý các câu hỏi thường gặp (FAQ) liên quan đến trường đại học, chẳng hạn như quy trình tuyển sinh, học phí và thông tin khóa học. Chatbot được đánh giá về độ chính xác trong việc truy xuất thông tin chính xác từ dữ liệu đã được vector hóa lưu trữ trong Pinecone, cũng như khả năng tạo ra các phản hồi mạch lạc và phù hợp với ngữ cảnh.

Quá trình fine-tuning, được thực hiện bằng cách sử dụng các kỹ thuật PEFT và LoRA, nhằm mục đích nâng cao hiệu suất của chatbot. Thông qua các phương pháp này, tôi đã có thể điều chỉnh hiệu quả các tham số của mô hình để phù hợp hơn với nhu cầu cụ thể của hệ thống thông tin của trường đại học, mà không yêu cầu nguồn lực tính toán lớn.

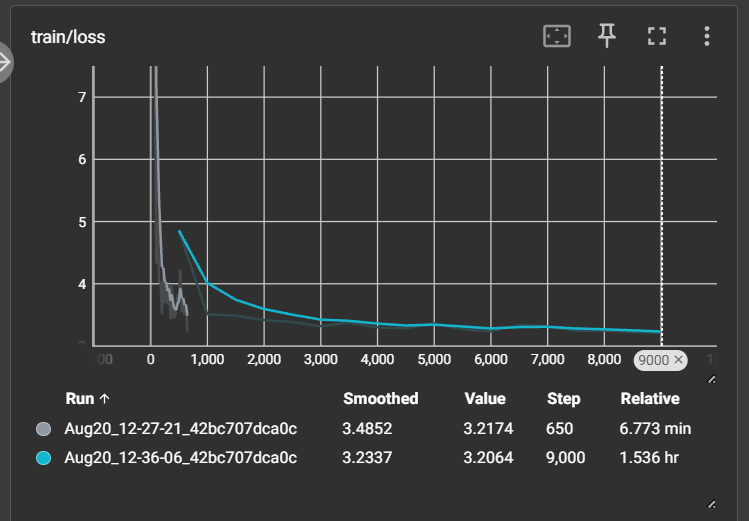
* + 1. **Các chỉ số đánh giá**

Hiệu quả của chatbot được đo lường bằng một số chỉ số chính:

* **Biểu đồ train loss**

Nội dung: Biểu đồ này thể hiện quá trình giảm thiểu loss (tổn thất) trong quá trình huấn luyện mô hình qua các epoch.

Phân tích: Biểu đồ cho thấy giá trị loss giảm dần qua các epoch, điều này chứng tỏ rằng mô hình đang học và tối ưu hóa tốt theo thời gian. Một train loss giảm đều và ổn định là một dấu hiệu tốt, cho thấy mô hình không bị overfitting và vẫn đang tiếp tục học từ dữ liệu nhưng đã gần như bão hòa ở mức tổn thất 3.0.



* **Biểu đồ epoch**

Nội dung: Biểu đồ này cho thấy số lần mô hình đã học qua toàn bộ dữ liệu huấn luyện.

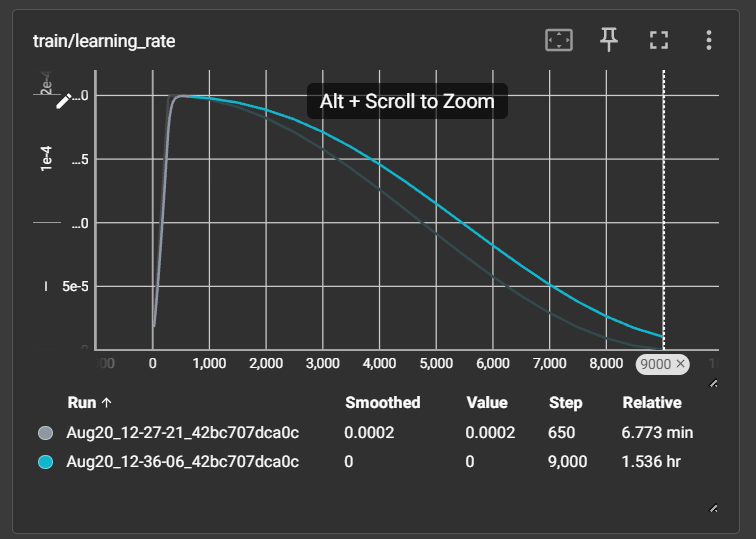
Phân tích: Số lượng epoch cần thiết để mô hình đạt hiệu suất tối ưu phụ thuộc vào nhiều yếu tố, bao gồm kích thước dữ liệu và độ phức tạp của mô hình. Train loss giảm đến một mức ổn định sau một số lượng epoch nhất định, là dấu hiệu rằng quá trình huấn luyện có thể dừng lại.



* **Biểu đồ learning rate**

Nội dung: Biểu đồ này theo dõi sự thay đổi của learning rate trong quá trình huấn luyện.

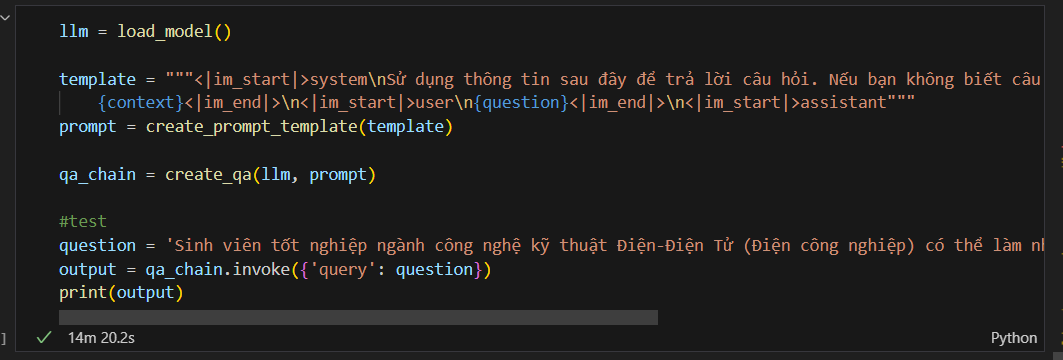
Phân tích: Learning rate đóng vai trò quan trọng trong việc điều chỉnh tốc độ học của mô hình. Một learning rate quá cao có thể khiến mô hình học quá nhanh và bỏ qua các chi tiết quan trọng, dẫn đến overfitting hoặc underfitting. Ngược lại, một learning rate quá thấp sẽ khiến quá trình huấn luyện chậm và có thể không hiệu quả. Biểu đồ này thể hiện sự điều chỉnh learning rate từ ban đầu cao và giảm dần để đạt hiệu suất tốt nhất.



* **Thời gian phản hồi**

Nội dung: Đo lường thời gian mà chatbot cần để đưa ra câu trả lời.

Phân tích: Thời gian phản hồi khá lâu cho thấy chatbot hoạt động cần dựa vào lưu lượng mạng để lấy dữ liệu từ cloud và phản hồi thông qua mức độ tính toán của máy. Chưa đáp ứng được nhiều với yêu cầu của người dùng.



* + 1. **Cải tiến**

Dựa trên kết quả kiểm thử, tôi đã thực hiện một số cải tiến cho chatbot:

Cải thiện xử lý dữ liệu: Tôi đã tinh chỉnh các bước tiền xử lý dữ liệu để loại bỏ các thông tin nhiễu và không cần thiết một cách hiệu quả hơn. Điều này giúp cải thiện độ chính xác của các phản hồi của chatbot.

Tối ưu hóa mô hình: Việc fine-tuning với LoRA và PEFT đã giảm đáng kể thời gian cần thiết cho việc huấn luyện mô hình, đồng thời vẫn duy trì chất lượng phản hồi cao. Điều này cũng giúp quá trình thử nghiệm diễn ra nhanh chóng hơn.

Xử lý phản hồi: Đã có những cải tiến trong cơ chế tạo phản hồi, đặc biệt là trong việc xử lý các trường hợp mà chatbot có thể hiểu sai câu hỏi của người dùng. Điều này được thực hiện bằng cách mở rộng tập dữ liệu huấn luyện để bao gồm nhiều tình huống đa dạng hơn và điều chỉnh quy trình truy xuất kết hợp tạo phản hồi (RAG).

Giao diện người dùng: Phản hồi từ người dùng cho thấy cần có một giao diện trực quan và thân thiện hơn. Do đó, tôi đã thực hiện các điều chỉnh đối với giao diện của chatbot, đảm bảo rằng các phản hồi không chỉ chính xác mà còn được trình bày một cách rõ ràng và ngắn gọn.

* Những cải tiến này đã đóng góp vào việc xây dựng một chatbot mạnh mẽ và đáng tin cậy hơn, có khả năng cung cấp thông tin chính xác và kịp thời cho người dùng tại Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh.

1. **Thành tựu của dự án**
   1. **Thành tựu và lợi ích**

Hoàn thành xây dựng chatbot thông minh: Đã triển khai thành công một hệ thống chatbot sử dụng công nghệ tiên tiến, có khả năng trả lời các câu hỏi từ sinh viên và cung cấp thông tin về các quy định, thông báo, và thủ tục của Trường Đại học Sư phạm Kỹ thuật TP.HCM.

Cải thiện trải nghiệm người dùng:Chatbot giúp sinh viên dễ dàng tiếp cận thông tin cần thiết mà không cần phải tìm kiếm thủ công qua các tài liệu hoặc liên hệ trực tiếp với nhà trường. Điều này không chỉ tiết kiệm thời gian mà còn giảm tải công việc cho các phòng ban liên quan.

Tự động hóa quy trình hỗ trợ sinh viên: Việc tự động hóa quá trình cung cấp thông tin qua chatbot giúp nhà trường tối ưu hóa quy trình hỗ trợ sinh viên, đảm bảo các yêu cầu và thắc mắc được giải quyết một cách nhanh chóng và hiệu quả.

Phát triển hệ thống học tập và quản lý hiệu quả: Chatbot có khả năng tích hợp với hệ thống quản lý học tập (LMS), hỗ trợ sinh viên trong việc tra cứu thông tin học vụ, điểm số, lịch học và đăng ký môn học, giúp việc quản lý học tập trở nên dễ dàng hơn.

Ứng dụng công nghệ ai hiện đại:Dự án đã áp dụng thành công các công nghệ AI tiên tiến như mô hình ngôn ngữ GPT, RAG (Retrieval-Augmented Generation), và các kỹ thuật fine-tuning như LoRA và PEFT, nâng cao khả năng xử lý ngôn ngữ tự nhiên và tương tác của chatbot.

Tiết kiệm chi phí và tài nguyên: Nhờ việc sử dụng chatbot, nhà trường có thể tiết kiệm chi phí và tài nguyên trong việc giải đáp thắc mắc của sinh viên, đồng thời tập trung nguồn lực vào các hoạt động giáo dục và quản lý khác.

* 1. **Hướng phát triển tương lai**

Trong tương lai, dự án chatbot hỗ trợ cung cấp thông tin cho Trường Đại học Sư phạm Kỹ thuật TP.HCM sẽ được phát triển thêm để phục vụ nhiều mục tiêu đa dạng hơn.

Mở rộng cơ sở dữ liệu: Trước hết, tôi sẽ tập trung mở rộng cơ sở dữ liệu của chatbot, bao gồm việc tích hợp thêm các nguồn dữ liệu mới từ các quy định, thông báo mới của nhà trường, cùng với việc cập nhật liên tục các câu hỏi thường gặp (FAQ) từ sinh viên.

Cải thiện độ chính xác của mô hình: Đồng thời, việc cải thiện độ chính xác của mô hình bằng cách áp dụng các kỹ thuật tiên tiến như Reinforcement Learning from Human Feedback (RLHF) sẽ được chú trọng, nhằm giúp chatbot đưa ra các câu trả lời không chỉ chính xác mà còn phù hợp với ngữ cảnh và nhu cầu của người dùng.

Tối ưu hóa thời gian phản hồi và tính khả dụng:Một khía cạnh quan trọng khác là tối ưu hóa thời gian phản hồi và tính khả dụng của chatbot, để đảm bảo rằng hệ thống luôn sẵn sàng phục vụ người dùng trong mọi tình huống. Tôi cũng sẽ nghiên cứu tích hợp chatbot với các nền tảng giao tiếp phổ biến như Facebook Messenger, Zalo, hoặc hệ thống chatbot trên website của trường, giúp sinh viên dễ dàng tiếp cận thông tin mọi lúc, mọi nơi.

Nâng cao trải nghiệm người dùng: Cuối cùng, để nâng cao trải nghiệm người dùng, tôi dự định phát triển giao diện người dùng (UI) của chatbot trở nên thân thiện và dễ sử dụng hơn, cùng với việc hỗ trợ đa ngôn ngữ để phục vụ các sinh viên quốc tế. Các công nghệ AI mới, như GPT-4 hoặc các phiên bản nâng cấp, cũng sẽ được tích hợp vào hệ thống để cải thiện khả năng hiểu biết và tương tác của chatbot.

Tích Hợp Đa Kênh (Omnichannel Integration):Tôi cũng sẽ mở rộng nền tảng hỗ trợ của chatbot, xây dựng khả năng hoạt động trên nhiều nền tảng giao tiếp khác nhau như Facebook Messenger, Zalo, website, ứng dụng di động, và các hệ thống LMS (Learning Management System). Điều này không chỉ mở rộng phạm vi ứng dụng của chatbot mà còn nâng cao trải nghiệm người dùng thông qua tích hợp liền mạch trên nhiều kênh giao tiếp.

Cải Thiện Xử Lý Ngôn Ngữ Tự Nhiên (NLP):Việc phát triển khả năng xử lý đa ngôn ngữ cũng là một ưu tiên. Chatbot sẽ được cải thiện để hỗ trợ các ngôn ngữ khác ngoài tiếng Việt, nhằm phục vụ người dùng quốc tế và hỗ trợ các chương trình hợp tác liên trường. Điều này đòi hỏi mô hình NLP phải có khả năng hiểu và tương tác chính xác với các ngôn ngữ khác nhau, đồng thời đảm bảo độ chính xác trong các bối cảnh ngữ nghĩa đa dạng.

Tích Hợp Công Nghệ Tiên Tiến:Trong kế hoạch phát triển, tôi dự định nâng cấp chatbot với khả năng nhận diện và xử lý giọng nói (ASR - Automatic Speech Recognition). Kết hợp với các kỹ thuật NLP tiên tiến, chatbot sẽ có thể chuyển đổi giọng nói thành văn bản và ngược lại, giúp tăng cường tính tương tác và tạo ra trải nghiệm người dùng tự nhiên, liền mạch hơn.

Mở Rộng Chức Năng Chatbot:Chatbot cũng sẽ được mở rộng thêm các chức năng mới, chẳng hạn như tư vấn lộ trình học tập, chọn ngành học phù hợp, và cung cấp thông tin chi tiết về các quy trình tuyển sinh. Tích hợp với hệ thống LMS sẽ cho phép sinh viên thực hiện các thao tác quản lý học vụ trực tiếp qua chatbot, như tra cứu điểm số, lịch học, lịch thi, và đăng ký môn học. Cuối cùng, việc phát triển chức năng thông báo và nhắc nhở thông minh sẽ giúp sinh viên quản lý lịch trình học tập và các sự kiện quan trọng, đồng thời cung cấp các cảnh báo tự động về hạn chót và hoạt động ngoại khóa.

**🡪** Thông qua những cải tiến và mở rộng này, dự án chatbot không chỉ đáp ứng tốt hơn nhu cầu của sinh viên mà còn góp phần nâng cao chất lượng dịch vụ hỗ trợ sinh viên của Trường Đại học Sư phạm Kỹ thuật TP.HCM.

1. **Nhận xét của thành viên trong nhóm**

|  |
| --- |
|  |

|  |  |
| --- | --- |
| TÊN | NHẬN XÉT |
| Thành | Dự án còn nhiều thiếu sót và cần bổ sung thêm nhiều như giao diện, cải tiến độ chính xác, thời gian phản hồi… |

1. **Đánh giá và nhận xét của giảng viên**

|  |  |  |
| --- | --- | --- |
| **CHỦ ĐỀ** | **ĐIỂM SỐ** | **ĐÁNH GIÁ VÀ NHẬN XÉT** |
| Ý TƯỞNG | \_\_/10 |  |
| ỨNG DỤNG | \_\_/30 |  |
| KẾT QUẢ | \_\_/30 |  |
| QUẢN LÝ DỰ ÁN | \_\_/10 |  |
| THUYẾT TRÌNH VÀ BÁO CÁO | \_\_/20 |  |
| TỔNG KẾT | \_\_/100 |  |

**TỔNG KẾT**

Trong quá trình xây dựng chatbot cung cấp thông tin về Trường Đại học Sư phạm Kỹ thuật, chúng tôi đã trải qua nhiều giai đoạn từ thu thập dữ liệu, thiết kế hệ thống, triển khai và đánh giá. Mục tiêu của dự án là phát triển một chatbot sử dụng các công nghệ hiện đại như quy trình RAG, công nghệ LangChain, các mô hình ngôn ngữ lớn (LLM), và các phương pháp học sâu để tự động trả lời các câu hỏi của sinh viên và giảng viên một cách chính xác và nhanh chóng.

Kết quả đạt được từ dự án này không chỉ đóng góp vào việc nâng cao trải nghiệm của sinh viên và giảng viên tại Trường Đại học Sư phạm Kỹ thuật mà còn mở ra hướng đi mới cho việc ứng dụng trí tuệ nhân tạo trong giáo dục.

Dự án này đã chứng minh tính khả thi và hiệu quả của việc sử dụng các công nghệ AI tiên tiến để xây dựng một hệ thống chatbot thông minh, mang lại nhiều lợi ích thiết thực cho cộng đồng giáo dục. Với những kết quả đạt được, chúng tôi tin rằng dự án này sẽ là nền tảng cho các nghiên cứu và ứng dụng tiếp theo trong lĩnh vực trí tuệ nhân tạo và học máy.

**THAM KHẢO**

1. Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras, and Tensorflow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media.
2. Dave Ebbelaar. (2023, Dec 21). PostgreSQL as VectorDB - Beginner Tutorial. Youtube. From: <https://www.youtube.com/watch?v=Ff3tJ4pJEa4>
3. Abid Ali Awan. (2023, Oct). Fine-Tuning LLaMA 2: A Step-by-Step Guide to Customizing the Large Language Model. Datacamp. From: <https://www.datacamp.com/tutorial/fine-tuning-llama-2>
4. Maithili Badhan. (2023). A Comprehensive Comparison Of Open Source LLMs. Mercity. From: <https://www.mercity.ai/blog-post/comprehensive-comparison-of-llms-8-2023>
5. Akash Takyar. (2024). Comparison of Large Language Models (LLMs): A detailed analysis. From: <https://www.leewayhertz.com/comparison-of-llms/>
6. XDashAI. (2024). VinaLLaMA so sánh với các mô hình ngôn ngữ khác như PhoGPT?. From: <https://www.xdash.ai/search?q=VinaLLaMA%20so%20s%C3%A1nh%20v%E1%BB%9Bi%20c%C3%A1c%20m%C3%B4%20h%C3%ACnh%20ng%C3%B4n%20ng%E1%BB%AF%20kh%C3%A1c%20nh%C6%B0%20PhoGPT%3F&rid=zkaQuZ5PUJESGIgp0UXTe>
7. Chroma. (2024). Multimodal. From: <https://docs.trychroma.com/guides/multimodal>
8. Stackoverflow (2024). How do I extract a table from a pdf file using pymupdf. From: <https://stackoverflow.com/questions/56155676/how-do-i-extract-a-table-from-a-pdf-file-using-pymupdf>
9. AI VIET NAM. (2024, May). LLMs: Xây dựng ứng dụng RAG với LangChain (AIO2023). Youtube. From: <https://www.youtube.com/watch?v=tFIzbS7UryI>
10. Trường Đại học Sư phạm kỹ thuật. (2024). From: <https://hcmute.edu.vn/>
11. Thư viện số trường Đại học Sư phạm kỹ thuật (2024). From: <https://thuvienso.hcmute.edu.vn/>