ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA HỌC MÁY TÍNH

KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG – CS313.O21

Đề tài

Triển khai hệ thống trợ lí giảng dạy ảo trong khóa học trực tuyến MOOCCubeX

Nhóm 2:

Lê Chí Thành	20521912
Trương Quốc Khánh	20520580
Đặng Quang Hải	21520027
Nguyễn Ngọc Lương	21522311
Nguyễn Lưu Trọng Tấn	21520103

Hồ Chí Minh, 30 tháng 06 năm 2024

Nội dung

1. Tổng quan đề tài 1
   1.1. Giới thiệu 1
   1.2. Định nghĩa bài toán 1
   1.3. Mục tiêu 1
   1.4. Đối tượng và phạm vi 1
   1.5. Thách thức của bài toán 2
2. Các công trình nghiên cứu liên quan 2
3. Cơ sở lý thuyết 3
   3.1. Data Mining 3
   3.2. Deep Learning 4
   3.3. Natural Language Processing 4
   3.4. Question Answering 5
   3.5. RAG 5
   3.6. Langchain 6
   3.7. Pinecone 6
4. Phương thức thực hiện 7
   4.1. Tổng quan các bước thực hiện 7
   4.2. Thiết kế kiến trúc dữ liệu lớn 7
5. Thực nghiệm 9
   5.1. Dataset 9
      5.1.1. Dữ liệu để xây dựng vector database 9
      5.1.2. Dữ liệu để đánh giá mô hình 10
   5.2. Độ đo đánh giá 11
   5.3. Phương pháp thực nghiệm 12
      5.3.1. BERT 12
      5.3.2. RoBERTa 12
      5.3.3. DeBERTaV3 12
      5.3.4. ALBERT 13
      5.3.5. Chinese MRC Roberta 13
   5.4. Kết quả thực nghiệm 13
6. Kết luận và hướng phát triển 14

Tài liệu tham khảo 14

1. Tổng quan đề tài

1.1. Giới thiệu

Trong thời đại ngày nay, việc phát triển các hệ thống trợ lí giảng dạy ảo đã trở thành một xu hướng quan trọng trong lĩnh vực giáo dục. Trợ lý giảng dạy ảo là một hệ thống thông minh được xây dựng dựa trên trí tuệ nhân tạo (AI) và có khả năng tương tác với học sinh hoặc sinh viên qua các kênh truyền thông như ứng dụng di động, trang web, hoặc chatbot. Đặc biệt, các nền tảng học trực tuyến đang ngày càng trở nên phổ biến, và nhu cầu về sự hỗ trợ trong quá trình học tập cũng tăng lên. Bài báo cáo này tập trung vào việc xây dựng một trợ lý giảng dạy ảo hiệu quả, có khả năng tương tác với người học, cung cấp thông tin, hỗ trợ giải đáp câu hỏi, và thậm chí đưa ra gợi ý cho việc học tập.

1.2. Định nghĩa bài toán

Trong bài báo cáo này, nhóm áp dụng các mô hình Question-Answering. Mô hình này có thể truy xuất câu trả lời cho một câu hỏi từ một văn bản nhất định, điều này rất hữu ích cho việc tìm kiếm câu trả lời trong tài liệu. Một số mô hình trả lời câu hỏi có thể tạo ra câu trả lời mà không cần ngữ cảnh.

Input: Câu hỏi của học viên.

Output: Câu trả lời được truy xuất từ tài liệu cho câu hỏi đó.

1.3. Mục tiêu

Mục tiêu chính của đề tài là xây dựng một hệ thống trợ lí giảng dạy ảo hiệu quả, hỗ trợ giảng viên trong việc giảng dạy và quản lý khóa học trực tuyến trên nền tảng MOOCCubeX. Cụ thể, các mục tiêu cụ thể bao gồm:

1. Nghiên cứu và đánh giá các công nghệ hiện có liên quan đến hệ thống trợ lí giảng dạy ảo.
2. Phát triển và tích hợp hệ thống trợ lí giảng dạy ảo vào nền tảng MOOCCubeX.
3. Đánh giá hiệu quả của hệ thống thông qua các thử nghiệm và phản hồi từ giảng viên và học viên.
4. Đề xuất các cải tiến và phát triển tương lai cho hệ thống trợ lí giảng dạy ảo.

1.4. Đối tượng và phạm vi

Đối tượng nghiên cứu của đề tài bao gồm giảng viên và học viên tham gia các khóa học trực tuyến trên nền tảng MOOCCubeX. Hệ thống trợ lí giảng dạy ảo sẽ hỗ trợ giảng viên trong việc quản lý khóa học, tương tác với học viên, cung cấp tài liệu học tập và

giải đáp các thắc mắc. Học viên sẽ nhận được hỗ trợ trực tiếp từ trợ lí giảng dạy ảo thông qua các tính năng như trả lời câu hỏi, cung cấp tài liệu học tập, và gợi ý nội dung liên quan.

Phạm vi của đề tài bao gồm việc nghiên cứu, phát triển, triển khai và đánh giá hệ thống trợ lí giảng dạy ảo trong môi trường MOOCCubeX. Các thử nghiệm sẽ được thực hiện trên một số khóa học cụ thể để đánh giá hiệu quả của hệ thống và thu thập phản hồi từ người dùng.

1.5. Thách thức của bài toán

1. Xử lý dữ liệu: Dữ liệu về sinh viên, môn học và điểm có thể thiếu sót, chứa nhiều nhiễu và giá trị bất thường do nhiều yếu tố khác nhau. Mô hình có thể gặp khó khăn trong việc học các mẫu phức tạp. Việc xử lý dữ liệu thiếu sót nhưng vẫn đảm bảo tính chính xác của dữ liệu là một thách thức trong đề tài này.
2. Tính phức tạp của nội dung giảng dạy: Nội dung các khóa học trực tuyến thường rất đa dạng và phức tạp, đòi hỏi hệ thống trợ lí giảng dạy ảo phải có khả năng hiểu và xử lý thông tin đa dạng.
3. Tương tác người dùng: Hệ thống phải đảm bảo tương tác hiệu quả với cả giảng viên và học viên, đáp ứng các yêu cầu và thắc mắc một cách nhanh chóng và chính xác.

2. Các công trình nghiên cứu liên quan

Những tiến bộ gần đây trong các trợ lý giảng dạy dựa trên trí tuệ nhân tạo (AI) đã cho thấy kết quả đầy hứa hẹn trong việc nâng cao kết quả giáo dục và cung cấp trải nghiệm học tập cá nhân hóa trong nhiều bối cảnh khác nhau.

1. **Kwame for Science sử dụng Sentence-BERT** để hỗ trợ giáo dục khoa học ở Tây Phi. Các nguồn dữ liệu bao gồm giáo trình WASSCE, câu hỏi trong sách giáo khoa và các cặp câu hỏi-trả lời từ CK-12 và Wikipedia đơn giản. Sentence-BERT chuyển đổi các câu và câu hỏi thành các biểu diễn, và ElasticSearch truy xuất ba câu trả lời hàng đầu dựa trên độ tương đồng cosine, đạt độ chính xác cao với Top 1 và Top 3.

2. **Tác động của KnustBot trong giáo dục đại học ở Ghana** được đánh giá bằng thiết kế tiền kiểm tra và hậu kiểm tra bán thực nghiệm. Phân tích với Jamovi 2.0.0 và các phương pháp như Phân tích phương sai theo mô hình chia lô (SPANOVA) và kiểm tra độc lập (independent-samples T-tests) cho thấy sự cải thiện đáng kể trong điểm thi cuối kỳ của sinh viên sử dụng KnustBot, cho thấy hiệu quả tích cực của nó đối với việc học tập.
3. **Rexy, được xây dựng với IBM Watson Assistant,** áp dụng xử lý ngôn ngữ tự nhiên để quản lý ngữ cảnh hội thoại. Các giai đoạn tiền xử lý và hậu xử lý duy

trì tính toàn vẹn ngữ cảnh, với độ chính xác cao trong việc phân loại ý định và nhận diện thực thể, làm nổi bật hiệu quả của nó trong việc cung cấp các phản hồi phù hợp với ngữ cảnh.

4. **Xiao-Shih,** một bot trả lời câu hỏi cho các khóa học trực tuyến mở (MOOCs) dựa trên tiếng Trung, xử lý các câu hỏi từ nền tảng MOOCs bằng các kỹ thuật như phân tích N-gram và loại bỏ từ dừng. Bot đạt độ chính xác cao trong việc trả lời câu hỏi của sinh viên, cải thiện trải nghiệm học tập.

5. **LittleMu,** một trợ lý giảng dạy ảo trực tuyến được phát triển bằng cách tích hợp các nguồn dữ liệu khác nhau và chuỗi lời nhắc giảng dạy. LittleMu có thể hỗ trợ giáo viên trong việc giảng dạy và học tập trực tuyến bằng cách cung cấp các tài nguyên học tập được cá nhân hóa và phản hồi theo thời gian thực cho sinh viên. LittleMu đã được đánh giá trong một nghiên cứu với 100 sinh viên học đại học. Kết quả cho thấy LittleMu có thể cải thiện hiệu suất học tập của sinh viên và tăng mức độ hài lòng của họ với trải nghiệm học tập trực tuyến.

6. **Kwame,** một trợ lý giảng dạy song ngữ được phát triển để hỗ trợ sinh viên trong các khóa học SuaCode trực tuyến có khả năng trả lời các câu hỏi về lập trình của sinh viên bằng tiếng Anh và tiếng Pháp. Hệ thống được huấn luyện ngoại tuyến bằng cách sử dụng các cặp câu hỏi-trả lời được tạo từ bài kiểm tra, ghi chú bài học và câu hỏi của sinh viên từ các khóa trước và tìm kiếm đoạn văn bản có nghĩa tương tự nhất với câu hỏi thông qua độ tương đồng cosin.

Các nghiên cứu này làm nổi bật các phương pháp đa dạng và những triển khai thành công của các trợ lý giảng dạy dựa trên AI và các mô hình tăng cường truy xuất, cải thiện đáng kể kết quả giáo dục và chuyên môn bằng cách cung cấp hỗ trợ cá nhân hóa và hiệu quả cho người dùng.

3. Cơ sở lý thuyết

3.1. Data Mining

Data mining (Hay còn gọi là khai phá dữ liệu) là quá trình tìm kiếm và khám phá thông tin tiềm ẩn, mẫu và quy luật từ tập dữ liệu lớn. Nó sử dụng các phương pháp và thuật toán máy học để phân tích và tạo ra những thông tin giá trị. Quá trình này có thể giúp doanh nghiệp hiểu rõ hơn về khách hàng, phát triển chiến lược tiếp thị hiệu quả hơn, giảm chi phí và tăng doanh số bán hàng. Các chương trình khai phá dữ liệu chia nhỏ các mẫu và kết nối trong dữ liệu dựa trên thông tin mà người dùng yêu cầu hoặc cung cấp. Ứng dụng của Data mining trong thực tế là rất lớn, nó có thể được sử dụng để quản lý rủi ro tín dụng, phát hiện gian lận, lọc email rác, hoặc thậm chí để nhận dạng cảm xúc hoặc ý kiến của mọi người, ...

3.2. Deep Learning

Deep Learning là một phương pháp của Machine Learning. Mạng nơ-ron nhân tạo trong Deep Learning được xây dựng để mô phỏng khả năng tư duy của bộ não con người.

Một mạng nơ-ron bao gồm nhiều lớp (layer) khác nhau, số lượng layer càng nhiều thì mạng sẽ càng "sâu". Trong mỗi layer là các nút mạng (node) và được liên kết với những lớp liền kề khác. Mỗi kết nối giữa các node sẽ có một trọng số tương ứng, trọng số càng cao thì ảnh hưởng của kết nối này đến mạng nơ-ron càng lớn.

Mỗi nơ-ron sẽ có một hàm kích hoạt, về cơ bản thì có nhiệm vụ "chuẩn hoá" đầu ra từ nơ-ron này. Dữ liệu được người dùng đưa vào mạng nơ-ron sẽ đi qua tất cả layer và trả về kết quả ở layer cuối cùng, gọi là output layer.

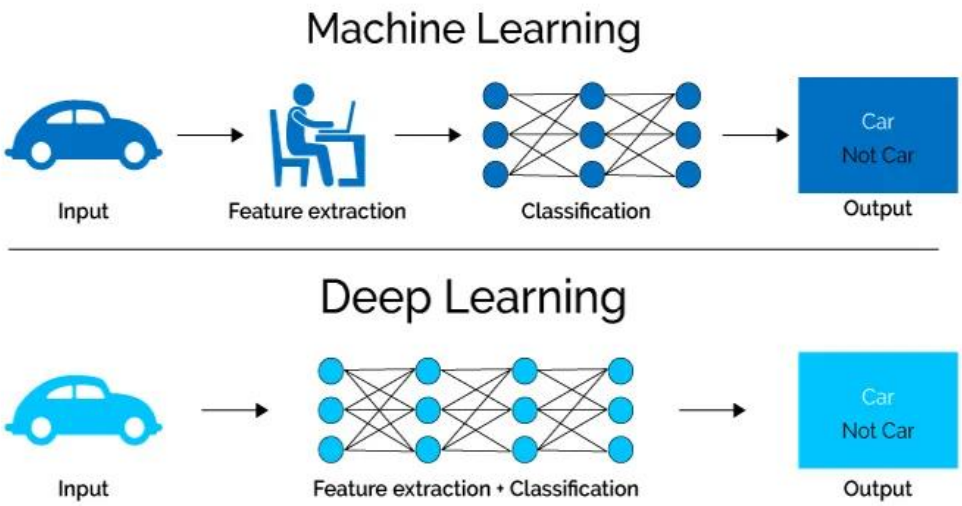

Figure 1: Cách thức hoạt động của Deep Learning

Trong quá trình huấn luyện mô hình mạng nơ-ron, các trọng số sẽ được thay đổi và nhiệm vụ của mô hình là tìm ra bộ giá trị của trọng số sao cho phán đoán là tốt nhất.

Các hệ thống Deep Learning yêu cầu phần cứng phải rất mạnh để có thể xử lý được lượng dữ liệu lớn và thực hiện các phép tính phức tạp. Nhiều mô hình Deep Learning có thể mất nhiều tuần, thậm chí nhiều tháng để triển khai trên những phần cứng tiên tiến nhất hiện nay.

3.3. Natural Language Processing

NLP là một lĩnh vực kết hợp giữa ngôn ngữ học và học máy, tập trung vào việc hiểu mọi thứ liên quan đến ngôn ngữ của con người. Mục đích của các tác vụ NLP không chỉ dừng ở hiểu từng từ đơn lẻ mà còn có thể hiểu ngữ cảnh của những từ đó.

Dưới đây là danh sách các tác vụ NLP phổ biến, với một số ví dụ về mỗi tác vụ:

- **Phân loại toàn bộ câu:** Nhận biết cảm xúc của bài đánh giá, phát hiện xem một bức thư điện tử có phải thư rác hay không, xác định xem một câu có đúng ngữ pháp hay không hoặc hai câu có liên quan về mặt logic hay không.
- **Phân loại từng từ trong câu:** Xác định các thành phần ngữ pháp của câu (danh từ, động từ, tính từ), hoặc các thực thể được đặt tên (người, vị trí, tổ chức).
- **Tạo nội dung văn bản:** Hoàn thành lời nhắc với văn bản được tạo tự động, điền vào chỗ trống trong văn bản có các từ bị che.
- **Trích xuất câu trả lời từ văn bản:** Cho một câu hỏi và ngữ cảnh, trích xuất câu trả lời cho câu hỏi dựa trên thông tin được cung cấp trong ngữ cảnh
- **Tạo câu mới từ văn bản đầu vào:** Dịch văn bản sang ngôn ngữ khác, tóm tắt văn bản.

NLP không giới hạn chỉ trong văn bản viết. Nó cũng giải quyết những thách thức phức tạp trong nhận dạng giọng nói và thị giác máy tính, chẳng hạn như tạo bản ghi chép từ âm thanh hoặc mô tả hình ảnh.

3.4. Question Answering

Question Answering là một công nghệ AI cho phép người dùng đặt câu hỏi và nhận được câu trả lời tự động từ hệ thống. Mục đích chính của QA là cung cấp thông tin đáng tin cậy và chính xác cho người dùng một cách nhanh chóng và thuận tiện.

Trong QA, có nhiều phương pháp và mô hình AI được sử dụng để xử lý câu hỏi và tìm kiếm thông tin. Các phương pháp này bao gồm sử dụng các thuật toán học máy, mô hình ngôn ngữ, xử lý ngôn ngữ tự nhiên, và học sâu (deep learning). Mô hình như BERT, Transformer và LSTM được áp dụng để cải thiện khả năng hiểu câu hỏi và đưa ra câu trả lời chính xác.

3.5. RAG

Retrieval-Augmented Generation (RAG) là một kỹ thuật giúp nâng cao khả năng của mô hình sinh (language model generation) kết hợp với tri thức bên ngoài (external knowledge). Phương pháp này thực hiện bằng cách truy xuất thông tin liên quan từ kho tài liệu (tri thức) và sử dụng chúng cho quá trình sinh câu trả lời dựa trên LLMs.

Tóm tắt ngắn gọn quá trình của RAG như sau:

- **Create Vector database:** Đầu tiên, convert toàn bộ dữ liệu tri thức thành các vector và lưu trữ chúng vào một vector database.
- **User input:** User cung cấp 1 câu truy vấn (query) bằng ngôn ngữ tự nhiên nhằm tìm kiếm câu trả lời hoặc để hoàn thành câu truy vấn đó.
- **Information retrieval:** Cơ chế retrieval quét toàn vộ vector trong database để xác định các phân đoạn tri thức (chính là paragraphs) nào có ngữ nghĩa tương đồng với câu truy vấn của người dùng. Các paragraphs này sau đó được vào LLM để làm tăng context cho quá trình sinh ra câu trả lời.

- **Combining data:** Các paragraphs được lấy sau quá trình retrieval từ database được kết hợp với câu query ban đầu của user tạo thành 1 câu prompt.
- **Generate text:** Câu prompt được bổ sung thêm context sau đó được đưa qua LLM để sinh ra câu phản hồi cuối cùng theo context bổ sung.

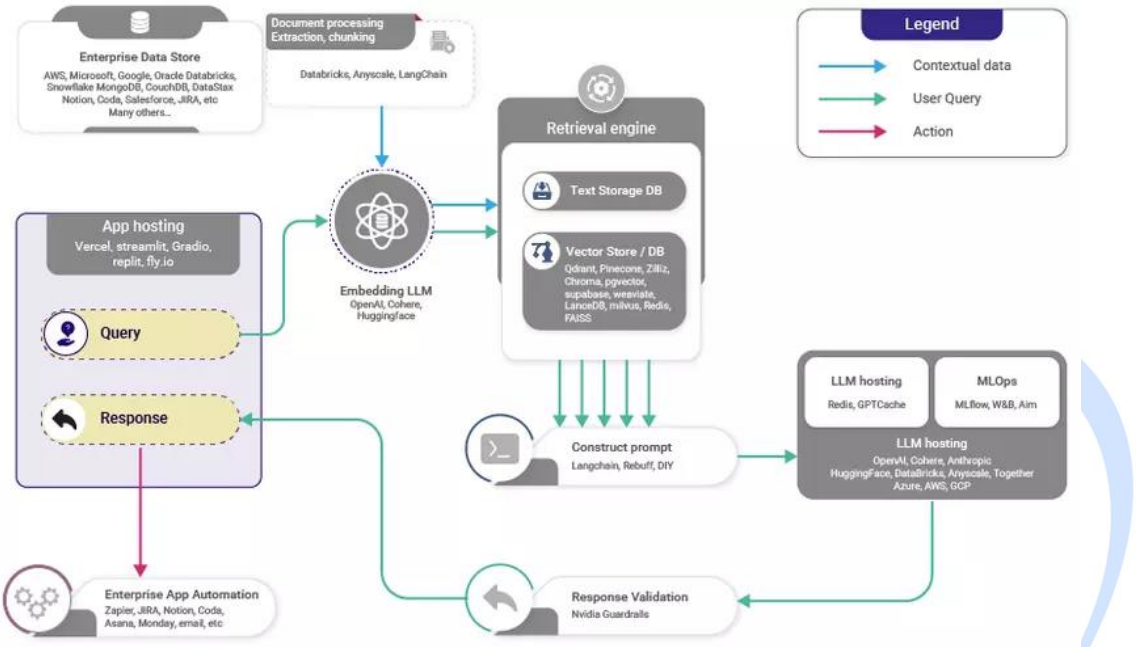

Figure 2: Cơ chế hoạt động của RAG

3.6. Langchain

LangChain là một framework được xây dựng xung quanh các mô hình ngôn ngữ lớn. Nó có thể được sử dụng để xây dựng chatbots, Generative Question-Answering(GQA), tóm tắt văn bản… Ý tưởng cốt lõi của thư viện là nối các thành phần khác nhau để tạo nên nhiều use cases nâng cao hơn từ LLMs.

3.7. Pinecone

Pinecone là một cơ sở dữ liệu vector, cho phép khởi tạo và sử dụng nhiều vector search trên các ứng dụng một cách dễ dàng. Pinecone kết hợp các thư viện vector search tiên tiến và được xây dựng với cơ sở hạ tầng phân tán nên có thể xử lý với hiệu năng cao và đáng tin cậy ở bất cứ quy mô nào.

Đặc điểm nổi bật của Pinecone:

- **Fast**: Có độ trễ cực thấp với bất kỳ quy mô nào, kể cả với hàng tỉ dữ liệu.
- **Fresh**: Cập nhật index trực tiếp ngay khi thêm, chỉnh sửa hoặc xóa dữ liệu.
- **Filtered**: Kết hợp vector search với metadata filters để cho kết quả chính xác và nhanh chóng hơn.
- **Fully managed**: Dễ dàng bắt đầu, sử dụng và nâng cấp một cách dễ dàng trong khi giữ cho mọi thứ hoạt động trơn tru và ổn định.

4. Phương thức thực hiện

4.1. Tổng quan các bước thực hiện

Dưới đây là tổng quan các bước thực hiện đề tài xây dựng trợ lý giảng dạy ảo của nhóm:

- **Bước 1 - Xác định bài toán:** mục tiêu nhóm cần đạt được, ý nghĩa của đề tài
- **Bước 2 - Thu thập dữ liệu:** trong đồ án này, nhóm thu thập dữ liệu từ 2 file là video.json và problem.json của bộ dữ liệu MOOCCubeX
- **Bước 3 - Phân tích và khai phá dữ liệu:** sử dụng các hàm đã học để thống kê, mô tả, trực quan hóa, khai phá những tri thức trong bộ dữ liệu
- **Bước 4 - Tiền xử lý dữ liệu:** Làm sạch dữ liệu, xử lý các giá trị Null, nhiễu
- **Bước 5 - Xây dựng database vector:** sử dụng các công cụ như Langchain, pinecone để xây dựng một cơ sở dữ liệu vector
- **Bước 6 - Xây dựng mô hình truy xuất thông tin** từ cơ sở dữ liệu vector dựa vào câu hỏi
- **Bước 7 - Đánh giá mô hình:** Sử dụng các độ đo khác nhau để đánh giá mô hình, sau đó tinh chỉnh để mô hình có kết quả tốt hơn
- **Bước 8 - Serve:** Sau khi chọn ra mô hình tốt nhất, nhóm sẽ sử dụng mô hình này để triển khai cho web app

4.2. Thiết kế kiến trúc dữ liệu lớn

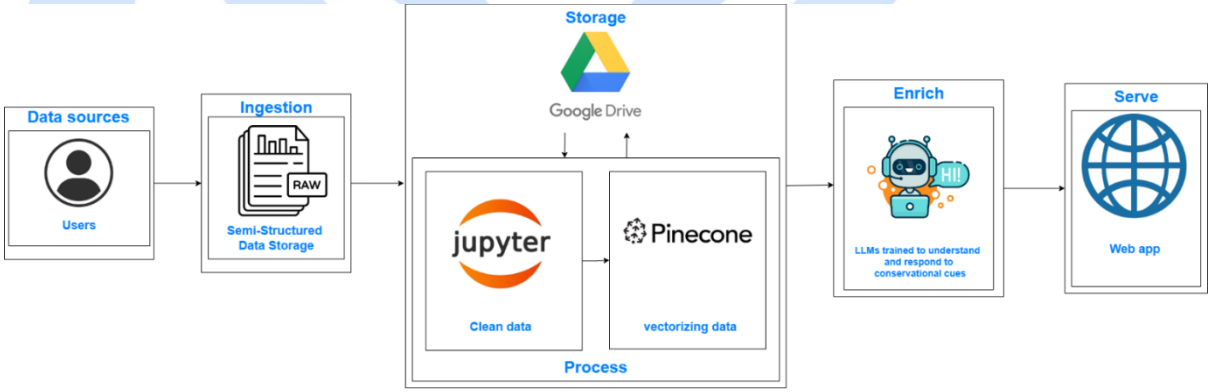

Figure 3: Kiến trúc dữ liệu lớn

A. **Data Sources:** 2 file dữ liệu video.json và problem.json mà nhóm sử dụng trong đồ án này được thu thập tại phần "The course recources" của bộ dữ liệu MOOCCubeX github.com/THU-KEG/MOOCCubeX/tree/main gồm theo dõi hành động của người dùng với video, bộ câu hỏi và đáp án của các khoá học trên XuetangX.

B. **Ingestion:**

- Thu thập dữ liệu: Dữ liệu từ các file JSON được thu thập định kỳ. Việc thu thập có thể được thực hiện tự động hoặc theo lịch trình nhất định.
- Lưu trữ bán cấu trúc: Các file JSON được lưu trữ trong một hệ thống lưu trữ bán cấu trúc. Điều này cho phép lưu trữ dữ liệu ở định dạng linh hoạt, dễ dàng truy xuất và quản lý mà không cần tuân theo một cấu trúc cố định nghiêm ngặt như trong cơ sở dữ liệu quan hệ.

C. **Storage:** Nhóm sử dụng Google Drive làm công cụ lưu trữ, Google Drive là dịch vụ lưu trữ đám mây được phát triển bởi Google. Google Drive cho phép người dùng lưu trữ tệp tin, đồng bộ hóa tệp giữa các thiết bị và chia sẻ tệp với người khác:

- Lưu trữ trung tâm: Google Drive được sử dụng làm nơi lưu trữ trung tâm cho tất cả các file dữ liệu thu thập được, đảm bảo tính nhất quán và dễ dàng truy cập.
- Quản lý file: Các file JSON từ bước Ingestion được tải lên Google Drive, nơi chúng có thể được tổ chức theo thư mục, ngày tháng, hoặc sự kiện cụ thể.
- Tích hợp API: Sử dụng Google Drive API để tự động hóa quá trình tải lên và quản lý file.

D. **Process:** Xử lý dữ liệu

- Clean Data (Jupyter):

- Jupyter Notebook: Sử dụng Jupyter Notebook để làm sạch dữ liệu. Các thao tác làm sạch bao gồm loại bỏ các bản ghi lỗi, dữ liệu trùng lặp, chuẩn hóa định dạng dữ liệu và xử lý các giá trị thiếu.
- Quy trình làm sạch:
  - Đọc dữ liệu: Sử dụng Pandas để đọc dữ liệu từ các file JSON.
  - Loại bỏ dữ liệu lỗi: Xác định và loại bỏ các bản ghi lỗi hoặc không hợp lệ.
  - Chuẩn hóa dữ liệu: Đảm bảo các giá trị có định dạng thống nhất (ví dụ: định dạng thời gian, định dạng ID).
  - Xử lý dữ liệu thiếu: Điền vào các giá trị thiếu hoặc loại bỏ các bản ghi thiếu dữ liệu quan trọng.

- Normalized, Vectorizing Data (Pinecone):

- Langchain: đầu tiên chia nhỏ các mẫu phụ đề hoàn chỉnh thành các đoạn nhỏ hơn (chunk) bằng cách sử dụng Recursive Character Text Splitter được cung cấp bởi LangChain, mỗi chunk được nhóm quy định có độ dài 500 kí tự và số lượng kí tự overlap là 20.
- Pinecone: nhóm lựa chọn embedding model cho tiếng Trung là bge-baze-zh để trích xuất embedding vector để bắt đầu xây dựng cơ sở dữ liệu vector, cuối cùng sử dụng pinecone để lưu trữ vector database.

E. **Enrich:** Sử dụng mô hình ngôn ngữ lớn LLMs:

- LLMs (Large Language Models): Sử dụng các mô hình ngôn ngữ lớn như GPT-3 để hiểu và phản hồi lại các ngữ cảnh hội thoại từ người dùng. Mô hình này có chức năng truy xuất tới vector database dựa vào câu hỏi đầu vào.

F. **Serve:**
- Ứng dụng web: Phát triển một ứng dụng web để người dùng có thể tương tác với hệ thống. Ứng dụng này cung cấp giao diện cho người dùng thực hiện các hành động và nhận phản hồi từ hệ thống.
- Tích hợp mô hình ngôn ngữ lớn: Mô hình ngôn ngữ lớn được tích hợp vào ứng dụng web để xử lý các yêu cầu từ người dùng và cung cấp các phản hồi phù hợp theo ngữ cảnh.
- Quy trình phát triển ứng dụng web:
    - Front-end: Xây dựng giao diện người dùng sử dụng các công nghệ như HTML, CSS, JavaScript, và các framework như React hoặc Angular.
    - Back-end: Xây dựng server để xử lý các yêu cầu từ front-end, sử dụng các framework như Flask, Django hoặc Node.js.
    - Tích hợp mô hình: Kết nối mô hình ngôn ngữ lớn với back-end để xử lý các câu hỏi và yêu cầu từ người dùng

5. Thực nghiệm

5.1. Dataset

5.1.1. Dữ liệu để xây dựng vector database

Bộ dữ liệu "video.json" mà nhóm đang sử dụng để huấn luyện mô hình chứa hơn 59.000 dòng dữ liệu, với mỗi dòng bao gồm 5 thuộc tính chính: ccid, name, start, end, và text.

Thuộc tính "ccid" đại diện cho một mã nhận dạng duy nhất được gán cho mỗi đoạn video, giúp phân biệt và truy xuất thông tin một cách dễ dàng. Thuộc tính "name" chứa tên của video hoặc đoạn video tương ứng, giúp việc quản lý và phân loại video trở nên thuận tiện hơn. Thuộc tính "start" và "end" lần lượt xác định thời điểm bắt đầu và kết thúc của đoạn video, cung cấp thông tin về thời lượng và vị trí của các đoạn trích cụ thể. Cuối cùng, thuộc tính "text" chứa nội dung văn bản của đoạn video, có thể là bản ghi lời thoại hoặc chú thích liên quan đến video.

Trong quá trình huấn luyện mô hình, nhóm tập trung chủ yếu vào hai thuộc tính là "ccid" và "text". Thuộc tính "ccid" giúp theo dõi và quản lý các đoạn video một cách hiệu quả, trong khi thuộc tính "text" cung cấp nguồn dữ liệu chính cho việc phân tích và huấn luyện mô hình. Bằng cách sử dụng kết hợp hai thuộc tính này, nhóm có thể xây

dựng và cải thiện mô hình một cách tối ưu, đảm bảo khả năng phân tích và dự đoán chính xác dựa trên dữ liệu văn bản từ các đoạn video.

Thuộc tính	Nội dung	Kiểu dữ liệu	Miền giá trị
ccid	Đây là một mã định danh duy nhất cho video	Chuỗi (String)	Một chuỗi ký tự bao gồm cả chữ cái và số.
name	Tên của video.	Chuỗi (String)	Một chuỗi văn bản, ví dụ: "Video".
start	Danh sách các điểm bắt đầu của các đoạn văn bản trong video (tính bằng giây).	Danh sách số thực (List of Floats)	Các giá trị số thực đại diện cho thời gian bắt đầu của mỗi đoạn, ví dụ: [1.031, 7.095, 8.935, ...]
end	Danh sách các điểm kết thúc của các đoạn văn bản trong video (tính bằng giây).	Danh sách số thực (List of Floats)	Các giá trị số thực đại diện cho thời gian kết thúc của mỗi đoạn, ví dụ: [4.255, 8.119, 10.033, ...]
text	Danh sách các đoạn văn bản tương ứng với các đoạn thời gian trong video.	Danh sách chuỗi (List of Strings)	Các đoạn văn bản, mỗi đoạn là một chuỗi văn bản, ví dụ: ["第二个就是短助记符在生成上面。有一个规定", "短助记符是这样说", ...]

Table 1: Bảng Video

5.1.2. Dữ liệu để đánh giá mô hình

Nhóm sẽ sử dụng một phần nhỏ của tập tin problem.txt làm bộ dữ liệu để đánh giá mô hình. Cụ thể, 1872 mẫu dữ liệu chứa thông tin về câu hỏi của học viên và câu trả lời. Cụ thể, bộ dữ liệu để đánh giá chứa những nội dung như sau:

Thuộc tính	Nội dung	Kiểu dữ liệu	Miền giá trị
Id	ID của câu hỏi	Chuỗi (String)	Một chuỗi ký tự bao gồm chữ cái và số
Question	Câu hỏi của học viên	Chuỗi (String)	Các chuỗi ký tự chứa câu hỏi và

			thông tin liên quan (ví dụ: "1、《资治通鉴》卷1记载：智宣子将以瑶为后，智果曰：“……瑶之贤于人
Answer	Đáp án đúng của câu hỏi.	Chuỗi (String)	Chuỗi ký tự chứa câu trả lời

Table 2: Bảng Problems

5.2. Độ đo đánh giá

Báo cáo này sử dụng độ đo **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation) để đánh giá hiệu suất của mô hình và độ chính xác của dự đoán.

ROUGE là một tập hợp các chỉ số và phần mềm được sử dụng để đo lường độ chính xác của các hệ thống tóm tắt bằng cách so sánh giữa văn bản tóm tắt tự động và văn bản tóm tắt tham chiếu, thường được tạo bởi con người.

ROUGE có nhiều biến thể, nhưng phổ biến nhất là **ROUGE-1**, **ROUGE-2** và **ROUGE-L**.

ROUGE-N: đo số lượng n-gram phù hợp giữa văn bản do mô hình tạo ra và tham chiếu do con người tạo ra.

- $Recall\ (R): R = \frac{G}{Số\ từ\ trong\ văn\ bản\ tham\ chiếu}$
- $Precision\ (P): P = \frac{G}{Số\ từ\ trong\ văn\ bản\ dự\ đoán}$
- $F1 - Score\ (F1): F1 = \frac{2.P.R}{P+R}$

- **ROUGE-1:** đánh giá sự trùng khớp của các từ đơn lẻ (unigram) giữa văn bản dự đoán và văn bản tham chiếu.

  G = Tổng số từ đơn trùng khớp giữa văn bản dự đoán và văn bản tham chiếu.

- **ROUGE-2:** đánh giá sự trùng khớp của các cặp từ liên tiếp (bigram) giữa văn bản dự đoán và văn bản tham chiếu.

  G = Tổng số đôi đơn trùng khớp giữa văn bản dự đoán và văn bản tham chiếu.

- **ROUGE-L:** dựa trên chuỗi con chung dài nhất (LCS)

G = Chuỗi con chung dài nhất trùng khớp giữa văn bản dự đoán và văn bản t ham chiếu.

Chỉ số ROUGE dao động từ 0 đến 1, với điểm số cao hơn thể hiện sự tương đồng cao hơn giữa bản tóm tắt tự động và bản tham chiếu

5.3. Phương pháp thực nghiệm

5.3.1. BERT

BERT (Bidirectional Encoder Representations from Transformers) là viết tắt của Bidirectional Encoder Representations from Transformers. Đây là một mô hình học sẵn, còn được gọi là pre-train model, được phát triển bởi Google. BERT học ra các vector đại diện theo ngữ cảnh 2 chiều của từ (từ trái qua phải và từ phải qua trái). Mô hình này được sử dụng để transfer sang các bài toán khác trong lĩnh vực xử lý ngôn ngữ tự nhiên. BERT đã thành công trong việc cải thiện những công việc gần đây trong việc tìm ra đại diện của từ trong không gian số (không gian mà máy tính có thể hiểu được) thông qua ngữ cảnh của nó. BERT là một bước đột phá mới trong công nghệ xử lý ngôn ngữ tự nhiên và đã được áp dụng rộng rãi trong nhiều ứng dụng NLP.

5.3.2. RoBERTa

RoBERTa (A Robustly Optimized BERT) là một phiên bản cải tiến của mô hình BERT (Bidirectional Encoder Representations from Transformers). Được phát triển bởi Facebook AI, RoBERTa đã đạt được nhiều thành công trong xử lý ngôn ngữ tự nhiên (NLP). Dưới đây là một số điểm quan trọng về RoBERTa:

- Độ chính xác cao: RoBERTa có khả năng dự đoán văn bản tạo ra với độ chính xác rất cao, kể cả khi miền hoặc mô hình thay đổi. Tỷ lệ phát hiện chính xác của văn bản do AI tạo ra có thể lên tới trên 75%.
- Quy mô lớn hơn: RoBERTa sử dụng thông số mô hình lớn hơn so với BERT. Thời gian huấn luyện của RoBERTa được cung cấp trong bài báo là 1 ngày, sử dụng 1024 GPU V100 2.
- Cải tiến từ BERT: RoBERTa cải tiến từ BERT bằng cách tinh chỉnh quá trình tiền huấn luyện và loại bỏ các hạn chế về đối tượng tiền huấn luyện.

5.3.3. DeBERTaV3

DeBERTaV3 là một mô hình ngôn ngữ pretrained, cải tiến từ mô hình gốc DeBERTa. Mô hình này sử dụng nhiệm vụ tiền huấn replaced token detection (RTD) thay vì mask language modeling (MLM), giúp tiết kiệm dữ liệu huấn luyện hơn. DeBERTaV3 cải thiện hiệu suất so với DeBERTa trên nhiều tác vụ xử lý ngôn ngữ tự nhiên (NLU) với dữ liệu huấn luyện lên đến 80GB. Nó sử dụng cơ chế attention

disentangled và mask decoder cải tiến. Mô hình này có thể được sử dụng cho nhiều tác vụ NLP như phân loại văn bản, tóm tắt, và dịch máy.

5.3.4. ALBERT

ALBERT (A Lite BERT) là một mô hình ngôn ngữ tiền huấn được phát triển bởi Google Research và Viện công nghệ Toyota. Mô hình này được xem là phiên bản kế thừa của BERT (Bidirectional Encoder Representations from Transformers), nhưng có số lượng tham số ít hơn đáng kể. Dưới đây là một số điểm quan trọng về ALBERT:

- Hiệu suất vượt trội: ALBERT đã đạt được kết quả ấn tượng trên nhiều bài toán NLP, bao gồm GLUE, RACE và SQuAD, mặc dù có số lượng tham số ít hơn so với BERT.
- Kiến trúc nhỏ gọn: ALBERT sử dụng kiến trúc giống với BERT, nhưng có hai điểm khác biệt quan trọng:
    - Biểu diễn phụ thuộc ngữ cảnh ẩn: ALBERT huấn luyện biểu diễn phụ thuộc ngữ cảnh ẩn của các từ (token) và tạo ra các vector word embedding.
    - Phương pháp huấn luyện khác: ALBERT sử dụng phương pháp huấn luyện khác với BERT để tạo ra kết quả tốt hơn dựa trên kiến trúc và phương pháp huấn luyện, thay vì chỉ tăng kích thước mô hình.

5.3.5. Chinese MRC Roberta

Chinese MRC Roberta là một phiên bản cải tiến của mô hình RoBERTa (A Robustly Optimized BERT) được huấn luyện cho tiếng Trung Quốc. Dựa trên mô hình gốc Chinese Roberta, phiên bản này đã được điều chỉnh và tinh chỉnh trên tập dữ liệu cmrc2018 để cải thiện hiệu suất trong tác vụ hỏi đáp về văn bản tiếng Trung.

5.4. Kết quả thực nghiệm

STT	Phương pháp	ROUGE F1		
		ROUGE-1	ROUGE-2	ROUGE-L
1	BERT	0.0021	0.0000	0.0022
2	ALBERT	0.0053	0.0005	0.0053
3	RoBERTa	0.0039	0.	0.0037
4	DeBERTa	0.0122	0.0005	0.0120
5	mRoBERTa	0.0023	0.0001	0.0021

Table 3: Bảng kết quả thực nghiệm

Có thể thấy model DeBERTa hoạt động vượt trội hơn so với các model khác. Nguyên nhân chính là nhờ phương pháp tiếp cận mới của DeBERTaV3, giúp cải thiện

hiệu quả huấn luyện và chất lượng của dữ liệu huấn luyện. Phiên bản mDeBERTaV3 được sử dụng trong bài có độ chính xác cao hơn 1.91% so với phiên bản DeBERTa gốc và được huấn luyện trên bộ dữ liệu đa ngôn ngữ, bao gồm tiếng Trung, giúp dễ dàng ứng dụng trên bộ dữ liệu MOOCCubeX.

6. Kết luận và hướng phát triển

Kết quả nhận được tuy không mang tính quyết định, nhưng vẫn cho thấy được tiềm năng của việc sử dụng MOOCCubeX cho bài toán xây dựng trợ lý ảo. Quá trình thực nghiệm của nhóm còn gặp nhiều hạn chế, bao gồm kích thước bộ dữ liệu nhỏ và thời gian thực hiện ngắn. Nghiên cứu trong tương lai với kích thước mẫu lớn hơn và thời gian thử nghiệm dài hơn có thể cần thiết để xác nhận những phát hiện này.

Mặc dù kết quả hiện tại không khả quan, nhóm nghiên cứu nhận thấy vẫn còn tiềm năng để phát triển thêm về mô hình trợ lý ảo.

Các hướng phát triển tiếp theo của đề tài:

- Trích xuất nhiều dữ liệu hơn từ bộ dữ liệu MOOCCubeX để có thể xây dựng database đầy đủ hơn.
- Thử nghiệm thêm các phương pháp khác để cải thiện kết quả dự đoán.
- Tối ưu hóa các siêu tham số của mô hình: tinh chỉnh các siêu tham số để đạt được hiệu suất tốt nhất.

Tài liệu tham khảo

1. George Boateng, Samuel John, Andrew Glago, Samuel Boateng, and Victor Kum bol. 2022. Kwame for Science: An AI Teaching Assistant for Science Education in West Africa. arXiv preprint arXiv:2206.13703 (2022) 2206.13703v2.pdf (arxiv.org)
2. Hao-Hsuan Hsu and Nen-Fu Huang. 2022. Xiao-Shih: A Self-enriched Ques tion Answering Bot With Machine Learning on Chinese-based MOOCs. IEEE Transactions on Learning Technologies (2022). DOI:10.1109/TLT.2022.3162572
3. Luca Benedetto & Paolo Cremonesi.2019. Rexy, A Configurable Application for Building Virtual Teaching Assistants (2019). https://s.net.vn/Wtv6
4. Harry Barton Essel, Dimitrios Vlachopoulos, Akosua Tachie-Menson, Esi Eduafua Johnson & Papa Kwame Baah. The impact of a virtual teaching assistant (chatbot) on students' learning in Ghanaian higher education (2022)
5. Shangqing Tu, Zheyuan Zhang, Jifan Yu, Chunyang Li, Siyu Zhang, Zijun Yao, Lei Hou, Juanzi Li. LittleMu: Deploying an Online Virtual Teaching Assistant via Heterogeneous Sources Integration and Chain of Teach Prompts (2020)
6. George Boateng.Kwame: A Bilingual AI Teaching Assistant for Online SuaCode Courses AIED 2021