ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN CUỐI KỲ

MÔN: NHẬP MÔN HỌC MÁY

Giảng viên phụ trách: Thầy Bùi Tiến Lên
Thầy Lê Nhựt Nam
Thầy Bùi Duy Đăng

Nhóm sinh viên thực hiện: 22120053 - Lê Thành Đạt
22120123 - Nguyễn Minh Hưng
22120133 - Hà Đức Huy
22120153 - Trần Duy Khang
22120154 - Trịnh Hoàng Khang

Thành phố Hồ Chí Minh, năm 2025

MỤC LỤC

1 Danh sách thành viên và phân công công việc

1.1 Danh sách thành viên

STT	MSSV	Họ và tên	Email
1	22120053	Lê Thành Đạt	22120053@student.hcmus.edu.vn
2	22120123	Nguyễn Minh Hưng	22120123@student.hcmus.edu.vn
3	22120133	Hà Đức Huy	22120133@student.hcmus.edu.vn
4	22120153	Trần Duy Khang	22120153@student.hcmus.edu.vn
5	22120154	Trịnh Hoàng Khang	22120154@student.hcmus.edu.vn

Bảng 1: Danh sách thành viên

1.2 Phân công công việc

STT	Nhiệm vụ	Người thực hiện	Đánh giá
1	Thực hiện huấn luyện và finetune mô hình Mistral-7B Thực hiện quay video	Lê Thành Đạt	100%
2	Thực hiện tạo pipeline cho hệ thống Kiểm tra hệ thống	Nguyễn Minh Hưng	100%
3	Thực hiện huấn luyện và finetune mô hình Mistral-7B Thực hiện viết báo cáo	Hà Đức Huy	100%
4	Thực hiện huấn luyện và finetune mô hình Mistral-7B Kiểm tra hệ thống	Trần Duy Khang	100%
4	Thực hiện huấn luyện và finetune mô hình Mistral-7B Thực hiện làm slide	Trịnh Hoàng Khang	100%

Bảng 2: Bảng phân công công việc

2 Ý tưởng và mục tiêu Đề tài

2.1 Bối cảnh

Trong bối cảnh chuyển đổi số diễn ra mạnh mẽ trong lĩnh vực giáo dục, nhu cầu ứng dụng các công nghệ hỗ trợ học tập thông minh ngày càng tăng. Đặc biệt trong lĩnh vực Toán học – một lĩnh vực đòi hỏi tư duy logic, khả năng diễn giải theo từng bước rõ ràng, và tính chính xác cao – việc hỗ trợ người học bằng các công cụ số hóa có thể đóng vai trò quan trọng trong việc nâng cao hiệu quả học tập và giảng dạy.

Hiện nay, các công cụ truyền thống như máy tính bỏ túi (ví dụ CASIO), cũng như các hệ thống hỏi–đáp (QA) dựa trên trí tuệ nhân tạo đã bước đầu hỗ trợ phần nào nhu cầu tra cứu và giải toán. Tuy nhiên, những công cụ này vẫn còn gặp một số hạn chế nhất định khi áp dụng trong môi trường học thuật hoặc trong thực tiễn giảng dạy ở trường học:

- Khả năng suy luận nhiều bước hoặc giải thích các bước giải toán vẫn còn hạn chế, đặc biệt khi gặp các bài toán phức tạp hoặc có nhiều cách tiếp cận.
- Chưa có khả năng trích dẫn rõ ràng các định nghĩa toán học, khiến người học gặp khó khăn khi muốn đối chiếu hoặc tìm hiểu sâu hơn.
- Mặc dù đã có những bước tiến nhất định, việc nhận dạng và xử lý bài toán từ hình ảnh thực tế (ví dụ: đề thi, sách giáo khoa) vẫn còn hạn chế, đặc biệt trong việc bóc tách bố cục và chuyển đổi sang định dạng có thể hiểu được cho hệ thống.

Vì thế, vẫn còn nhiều tiềm năng để phát triển các hệ thống trợ lý học tập thông minh có khả năng hiểu ngôn ngữ tự nhiên tốt hơn, khả năng diễn giải toán học một cách hệ thống, và khả năng tương tác linh hoạt với nhiều định dạng đầu vào trong thực tế.

2.2 Ý tưởng

Từ những điều trên, nhóm đã hướng đến việc xây dựng một **trợ lý toán học**, ứng dụng các mô hình ngôn ngữ lớn (Large Language Models – LLMs) để hỗ trợ người học trong việc giải toán và tra cứu kiến thức toán học. Cụ thể, hệ thống sẽ được thiết kế để thực hiện ba chức năng chính như sau:

- **Giải toán có lập luận:** Hệ thống sử dụng mô hình ngôn ngữ `Mistral 7B v3`, được tinh chỉnh (fine-tune) bằng kỹ thuật LoRA. Dữ liệu huấn luyện được lấy từ bộ MetaMathQA (gồm 395.000 mẫu), trong đó nhóm chia thành nhiều phần nhỏ (mỗi phần khoảng 100.000 mẫu) để huấn luyện nhiều mô hình riêng biệt, từ đó tăng tính đa dạng trong suy luận. Quá trình huấn luyện sử dụng thư viện `Unsloth` để tối ưu tốc độ và tài nguyên phần cứng.
- **Nhận diện bài toán từ hình ảnh:** Hệ thống cho phép người dùng tải lên ảnh chứa đề bài về toán học, ví dụ như ảnh chụp đề thi hoặc sách giáo khoa. Sau đó, mô hình o4-mini (OpenAI API) được sử dụng để trích xuất nội dung văn bản từ ảnh. Văn bản này sẽ được chuyển tiếp cho mô hình giải toán xử lý và đưa ra lời giải chi tiết.
- **Tra cứu khái niệm toán học:** Hệ thống tích hợp cơ chế truy xuất định nghĩa toán học dựa trên độ tương đồng văn bản. Cụ thể, các khái niệm toán học trong cơ sở tri thức được lập chỉ mục bằng phương pháp `TF-IDF Vectorizer`, cho phép biểu diễn cả câu hỏi và dữ liệu dưới dạng vector. Khi người dùng nhập câu hỏi, hệ thống tính độ tương đồng cosine giữa truy vấn và các mục trong cơ sở dữ liệu, sau đó trả về định nghĩa phù hợp nhất.

2.3 Mục tiêu

Đề tài hướng đến việc xây dựng một hệ thống web hỗ trợ học tập môn Toán, tích hợp các công nghệ hiện đại như mô hình ngôn ngữ lớn (LLMs), trích xuất thông tin từ ảnh và truy xuất các khái niệm toán học, phục vụ nhu cầu học tập của học sinh – sinh viên. Cụ thể, hệ thống được thiết kế với các mục tiêu sau:

- Cho phép người dùng nhập đề bài môn toán dưới dạng văn bản hoặc tải lên hình ảnh đề bài. Hệ thống sẽ xử lý và trả về lời giải chi tiết, theo từng bước lập luận logic, sử dụng các mô hình Mistral-7B đã được fine-tune phù hợp với ngữ cảnh toán học.
- Cung cấp chức năng tra cứu khái niệm toán học: Người dùng có thể nhập một khái niệm bất kì (đã biết/chưa biết), hệ thống sẽ trả về định nghĩa của khái niệm đó. Thông tin về khái niệm đó được lấy ra từ cơ sở tri thức đã được lập chỉ mục bằng TF-IDF.
- Tích hợp giao diện người dùng đơn giản, trực quan, dễ sử dụng, phù hợp với đối tượng học sinh – sinh viên hay người học không chuyên về công nghệ.
- Đảm bảo tối ưu tài nguyên hệ thống, giảm tối đa độ trễ khi người dùng truy vấn giải toán hoặc tra cứu kiến thức.
- Đảm bảo chất lượng lời giải tốt nhất có thể. Hệ thống sẽ sinh ra nhiều lời giải cho cùng một bài toán, sau đó sử dụng cơ chế tái xếp hạng (re-ranking) với sự hỗ trợ từ API o4-mini để lựa chọn lời giải phù hợp nhất.

3 Phân tích và xác định vấn đề

3.1 Phân tích yêu cầu chức năng

Hệ thống **trợ lý toán học** được thiết kế để hỗ trợ người học thông qua ba chức năng cốt lõi, mỗi chức năng tương ứng với một luồng xử lý riêng biệt và có những yêu cầu kỹ thuật cụ thể.

3.1.1 Giải toán từ văn bản

Đây là chức năng chính và trọng tâm của hệ thống, cho phép người dùng nhập trực tiếp đề bài dưới dạng văn bản để nhận lại lời giải chi tiết theo từng bước.

- Hệ thống cần hiểu rõ nội dung bài toán: Phân tích ngữ cảnh, xác định biến số, mối quan hệ giữa các đại lượng và phép toán được đề cập.
- Sau khi phân tích, đề bài sẽ được gửi đến các mô hình Mistral-7B đã được tinh chỉnh trên bộ dữ liệu MetaMathQA (395k câu hỏi), được chia thành 4 phần dữ liệu riêng biệt để huấn luyện 4 mô hình độc lập.
- Mỗi mô hình sinh ra một lời giải với cách lập luận khác nhau. Các lời giải này được tái xếp hạng (re-rank) bằng mô hình o4-mini dựa trên tiêu chí: Tính đúng đắn của đáp án, mức độ logic của lập luận, và độ đầy đủ của lời giải.
- Kết quả cuối cùng được chọn là lời giải có chất lượng cao nhất, được trình bày rõ ràng, tuần tự như cách làm bài của con người.

3.1.2 Giải toán từ hình ảnh

Chức năng này hỗ trợ người dùng nhập đề bài bằng cách tải lên ảnh chụp đề bài.

- Hệ thống sử dụng mô hình o4-mini (OpenAI API) để trích xuất văn bản toán học từ ảnh đầu vào (kết hợp OCR và mô hình ngôn ngữ).
- Văn bản sau khi trích xuất sẽ được xử lý như một đầu vào văn bản thông thường, tiếp tục đưa qua pipeline xử lý giải toán giống như chức năng Giải toán từ văn bản.
- Thách thức chính là đảm bảo độ chính xác khi trích xuất công thức toán học, biểu thức hoặc định dạng đặc biệt (ví dụ: Phân số, căn bậc hai, ...).
- Việc xử lý lỗi trích xuất, chuẩn hóa biểu thức và làm sạch văn bản đóng vai trò quan trọng để đảm bảo kết quả đầu ra chính xác.

3.1.3 Tra cứu khái niệm toán học

Chức năng này giúp người học tra cứu nhanh các khái niệm toán học như định nghĩa, định lý một cách chính xác và dễ hiểu.

- Người dùng nhập từ khóa hoặc mô tả ngắn gọn về khái niệm cần tìm.
- Hệ thống sử dụng phương pháp truy xuất văn bản dựa trên vector TF-IDF để thực hiện hai bước sau:
  1. **Truy xuất tri thức:** Câu hỏi được biến đổi thành vector truy vấn bằng TF-IDF Vectorizer, sau đó so sánh với tập vector khóa đã lập chỉ mục từ cơ sở dữ liệu khái niệm toán học.
  2. **Trả về kết quả:** Hệ thống chọn mục có độ tương đồng cao nhất và trả về định nghĩa rõ ràng, phù hợp với truy vấn người dùng.
- Kết quả đầu ra là phần định nghĩa hoặc mô tả khái niệm được trích xuất từ cơ sở tri thức, giúp người học dễ tiếp thu và tra cứu hiệu quả.

3.2 Xác định các thách thức chính

Trong quá trình phát triển hệ thống **trợ lý toán học**, nhóm phải đồng thời giải quyết nhiều thách thức kỹ thuật và tích hợp, bao gồm:

1. **Tinh chỉnh và quản lý nhiều mô hình ngôn ngữ lớn (LLMs):** Hệ thống sử dụng tổng cộng 4 mô hình `Mistral-7B`, mỗi mô hình được fine-tune độc lập trên các phân đoạn khác nhau của bộ dữ liệu `MetaMathQA` (gồm các tập con 100k, 100k, 100k và 95k mẫu). Mục tiêu là tăng tính đa dạng trong cách lập luận và phản hồi của hệ thống. Tuy nhiên, việc tinh chỉnh nhiều mô hình lớn đặt ra các thách thức về bộ nhớ, thời gian huấn luyện, và quản lý phiên bản mô hình. Để giải quyết, nhóm sử dụng kỹ thuật `LoRA hoặc QLoRa` để giảm tải tài nguyên GPU và thư viện `Unsloth` nhằm tối ưu tốc độ và hiệu năng huấn luyện.
2. **Tái xếp hạng lời giải từ nhiều mô hình một cách tự động và hợp lý:** Khi nhiều mô hình cùng sinh lời giải cho một đề bài, việc chọn ra phương án tốt nhất không thể thực hiện bằng tay. Nhóm xây dựng một cơ chế đánh giá tự động, trong đó mô hình `o4-mini` (API thị giác-ngôn ngữ) đóng vai trò đánh giá chất lượng từng lời giải và sắp xếp lại theo độ hợp lý và chính xác.

3. **Xử lý ảnh đầu vào và chuẩn hóa biểu thức toán học:** Đầu vào dưới dạng ảnh đề bài thi/sách giáo khoa có thể chứa nhiều ký hiệu toán học đặc thù, không dễ nhận dạng bằng các kỹ thuật thông thường. Thách thức nằm ở việc bóc tách đúng nội dung toán học (chữ, số, biểu thức) và chuẩn hóa văn bản đầu ra để phù hợp với yêu cầu của mô hình ngôn ngữ. Hệ thống cần tích hợp thêm bước phát hiện lỗi, kiểm tra cú pháp toán học và lọc nhiễu từ ảnh đầu vào.

4. **Xây dựng hệ thống truy xuất tri thức hiệu quả dựa trên TF-IDF:** Đối với chức năng tra cứu khái niệm, cần xây dựng một cơ sở tri thức toán học đủ phong phú và được chỉ mục hóa tốt để hỗ trợ tìm kiếm nhanh chóng. Việc áp dụng phương pháp TF-IDF giúp hệ thống biểu diễn dữ liệu dưới dạng vector, từ đó so sánh truy vấn với cơ sở dữ liệu bằng độ tương đồng cosine. Thách thức chính là đảm bảo độ chính xác cao trong kết quả truy xuất và giảm thiểu độ trễ. Ngoài ra, kết quả trả về cần ngắn gọn, đúng trọng tâm để hỗ trợ người học hiệu quả nhất.

5. **Thiết kế giao diện người dùng và tích hợp toàn bộ pipeline hệ thống:** Giao diện của hệ thống được phát triển bằng thư viện `Gradio`, với ba thành phần chính: Giải toán, Nhận diện ảnh, và Tra cứu khái niệm. Việc tích hợp đầy đủ pipeline từ nhập liệu (văn bản hoặc ảnh), xử lý mô hình, đến hiển thị kết quả một cách liền mạch là một thách thức trong cả thiết kế luồng dữ liệu và trải nghiệm người dùng. Hệ thống cần đảm bảo tính ổn định, thân thiện, và khả năng mở rộng trong tương lai.

4 Một số khái niệm liên quan

4.1 Chatbot

Chatbot là một chương trình máy tính ứng dụng trí tuệ nhân tạo, cho phép giao tiếp với con người thông qua văn bản hoặc giọng nói. Chatbot thường được triển khai trong các nền tảng nhắn tin để hỗ trợ người dùng tự động hóa các tác vụ thường gặp hoặc cung cấp thông tin theo yêu cầu.

Về nguyên lý hoạt động, một chatbot thông thường gồm ba thành phần chính:

- **Translator**: Chuyển yêu cầu người dùng thành dạng mà máy có thể hiểu.
- **Processor**: Xử lý yêu cầu và thực thi hành động tương ứng.
- **Respondent**: Tạo phản hồi và gửi lại kết quả tới người dùng qua nền tảng giao tiếp.

Chatbot có thể được phân loại thành ba nhóm chính:

- **Task-oriented Dialogue Systems (TODs)**: Chatbot định hướng tác vụ, chuyên xử lý các yêu cầu cụ thể trong một lĩnh vực, như đặt vé, hỗ trợ khách hàng,...
- **Intelligent Personal Assistants (IPAs)**: Trợ lý ảo cá nhân như Siri, Google Assistant, Alexa,... có khả năng thực hiện nhiều tác vụ, tích hợp sâu với thiết bị cá nhân và dữ liệu người dùng.
- **Chit-chat Dialogue Systems (CDDs)**: Chatbot trò chuyện tự nhiên với người dùng nhằm tạo sự đồng hành và tương tác cảm xúc, ví dụ như Xiaoice, Replika, BlenderBot,...

Về mặt lịch sử, chatbot đầu tiên là ELIZA ra đời năm 1966, mô phỏng một bác sĩ tâm lý. Đến nay, chatbot đã phát triển nhanh chóng nhờ vào sự tiến bộ của AI, đáp ứng nhu cầu tương tác kỹ thuật số 24/7 trong nhiều lĩnh vực. Mặc dù vẫn còn những hạn chế về khả năng hiểu ngữ cảnh và kiểm soát nội dung, chatbot được kỳ vọng sẽ tiếp tục đóng vai trò quan trọng trong hệ sinh thái công nghệ số tương lai.

4.2 Xử lý ngôn ngữ tự nhiên (NLP)

Xử lý ngôn ngữ tự nhiên (Natural Language Processing – NLP) là một lĩnh vực của trí tuệ nhân tạo, tập trung vào nghiên cứu và phát triển các phương pháp để máy tính hiểu, phân tích và sinh ngôn ngữ tự nhiên của con người. NLP bao gồm hai nhánh chính: xử lý tiếng nói (nhận dạng và tổng hợp giọng nói) và xử lý văn bản (hiểu và sinh văn bản).

Các bài toán tiêu biểu trong NLP bao gồm:

- **Mô hình hóa ngôn ngữ (Language Modelling):** Dự đoán từ tiếp theo dựa trên chuỗi từ trước đó, đóng vai trò quan trọng trong nhiều ứng dụng như nhận dạng giọng nói, dịch máy, hoặc sửa lỗi chính tả.
- **Phân loại văn bản (Text Classification):** Gán nhãn cho văn bản, ví dụ như phân loại thư rác hoặc phân tích cảm xúc.
- **Trích xuất và truy xuất thông tin (Information Extraction & Retrieval):** Tự động trích xuất thực thể, sự kiện từ văn bản và tìm kiếm tài liệu liên quan trong kho dữ liệu lớn.
- **Tác tử phần mềm hội thoại (Conversational Agent):** Mô phỏng đối thoại tự nhiên giữa người và máy, ứng dụng trong trợ lý ảo và chatbot.
- **Tóm tắt và hỏi đáp (Summarization & Question Answering):** Sinh ra bản tóm tắt hoặc câu trả lời phù hợp từ văn bản nguồn.
- **Dịch máy (Machine Translation):** Chuyển văn bản giữa các ngôn ngữ tự nhiên, ví dụ như Google Dịch.
- **Mô hình hóa chủ đề (Topic Modelling):** Khám phá các chủ đề ẩn trong tập văn bản lớn.

NLP có nhiều ứng dụng quan trọng như công cụ dịch thuật, trợ lý ảo, phân tích cảm xúc trên mạng xã hội, phát hiện thư rác và tóm tắt văn bản tự động. Trong chatbot, NLP được sử dụng để:

- **Phân loại ý định người dùng:** Xác định mục đích của người dùng từ câu hỏi.
- **Trích xuất thực thể (Entity Extraction):** Nhận dạng các thông tin quan trọng trong câu hội thoại.
- **Quản lý hội thoại:** Duy trì trạng thái ngữ cảnh trong các cuộc hội thoại dài.
- **Tách từ:** Đặc biệt quan trọng trong tiếng Việt, giúp phân tách chính xác các đơn vị từ trong văn bản.

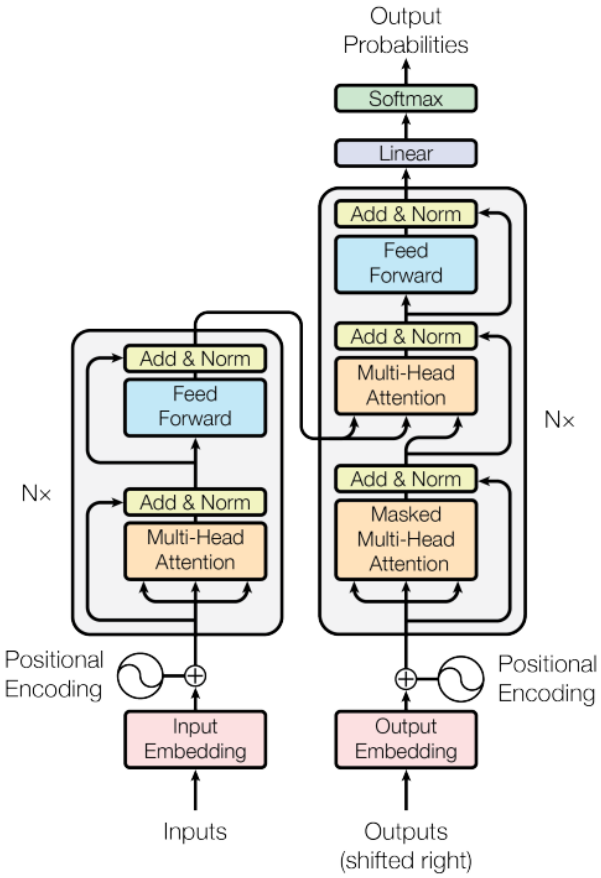
Nhờ sự phát triển của mô hình học sâu và dữ liệu lớn, NLP hiện nay là công nghệ nền tảng cho nhiều hệ thống thông minh hiện đại.

4.3 Transformer

Transformer là một kiến trúc mạng nơ-ron học sâu, được giới thiệu trong bài báo nổi tiếng *Attention is All You Need* (2017), chuyên dùng cho các bài toán xử lý ngôn ngữ tự nhiên như dịch máy, tổng hợp văn bản, và sinh ngôn ngữ. Khác với các mạng tuần tự truyền thống như RNN, Transformer xử lý dữ liệu theo cơ chế song song, nhờ đó tăng tốc độ huấn luyện và cải thiện khả năng học các mối quan hệ dài hạn giữa các từ.

Transformer gồm hai thành phần chính:

- **Bộ mã hóa (Encoder):** Gồm nhiều lớp, mỗi lớp bao gồm hai thành phần chính: Tầng *self-attention* giúp mô hình học mối quan hệ giữa các từ trong câu, và mạng *feed-forward* để trích xuất đặc trưng phi tuyến tính. Mỗi từ đầu vào được biểu diễn bằng cách kết hợp embedding và positional encoding để mô hình nhận biết được vị trí trong chuỗi.
- **Bộ giải mã (Decoder):** Cấu trúc tương tự encoder nhưng bổ sung tầng attention giữa bộ giải mã và bộ mã hóa. Decoder sinh đầu ra từng bước một, dựa trên các từ đã sinh trước đó và ngữ cảnh từ encoder.



Hình 1: Kiến trúc Transformer

Một điểm nổi bật trong kiến trúc này là cơ chế **Multi-head Attention** – cho phép mô hình tập trung vào nhiều phần khác nhau của chuỗi đầu vào cùng lúc, từ đó cải thiện khả năng biểu diễn ngữ nghĩa.

Transformer hiện là nền tảng cho nhiều mô hình ngôn ngữ lớn (LLMs) hiện đại như BERT, GPT, T5, hay Mistral.

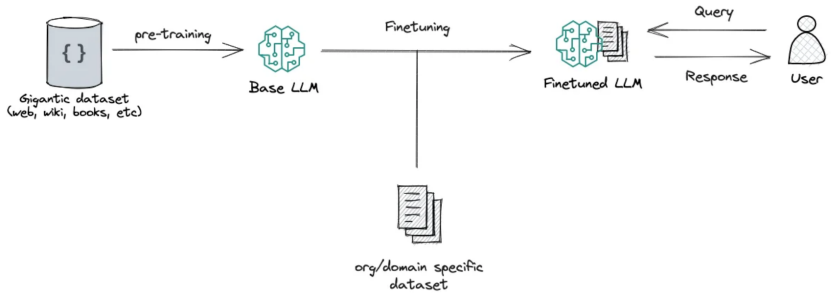
4.4 Large Language Model (LLM)

Large Language Model (LLM) là một mô hình ngôn ngữ được huấn luyện trên tập dữ liệu văn bản lớn, có khả năng xử lý và sinh ngôn ngữ tự nhiên với độ chính xác cao. Khác với các mô hình truyền thống, LLM có hàng tỉ tham số (thường từ 1B trở lên), cho phép nó học được các mối quan hệ ngữ nghĩa phức tạp và thể hiện năng lực như zero-shot learning.

LLM hoạt động bằng cách dự đoán từ tiếp theo trong chuỗi dựa trên ngữ cảnh, quá trình này lặp đi lặp lại để sinh ra văn bản hoàn chỉnh mà con người có thể hiểu được. Sau khi

huấn luyện tổng quát, LLM có thể được tinh chỉnh (finetuning) cho các nhiệm vụ cụ thể như chatbot, dịch máy, phân tích cảm xúc hoặc tạo nội dung.

Ứng dụng của LLM trải rộng trên rất nhiều lĩnh vực từ chăm sóc sức khỏe, tài chính, pháp lý, giáo dục đến giải trí. Việc tinh chỉnh mô hình trên dữ liệu chuyên ngành giúp nâng cao độ chính xác và khả năng áp dụng vào các lĩnh vực đặc thù như y học, ngân hàng, hay giáo dục đại học.



Hình 2: Quy trình huấn luyện và finetune

Quá trình huấn luyện và tinh chỉnh LLM gồm các bước cơ bản sau:

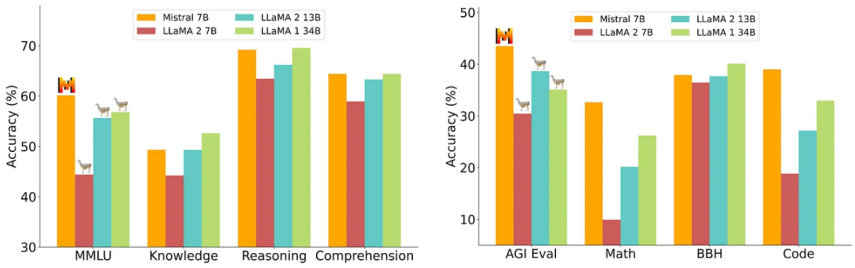
1. **Chọn một task fine-tuning:** Hỏi đáp về giải toán, về quy định sinh viên...
2. **Chuẩn bị cơ sở dữ liệu (dataset):** Bộ câu hỏi - trả lời, bộ dữ liệu về quy định, ...
3. **Chọn một model gốc:** Llama, Mistral, ...
4. Thực hiện huấn luyện, finetune mô hình và đánh giá mô hình

4.5 Mistral 7B

Mistral 7B, được phát hành vào tháng 9 năm 2023 bởi Mistral AI — một startup có trụ sở tại Paris được sáng lập bởi các cựu nhân viên của Meta và Google DeepMind — đánh dấu bước tiến quan trọng trong xu hướng phát triển các mô hình ngôn ngữ lớn (LLM) nhỏ gọn nhưng hiệu quả. Khác với xu hướng gia tăng số lượng tham số để nâng cao hiệu năng, Mistral 7B đạt hiệu suất vượt trội nhờ thiết kế kiến trúc tối ưu và cải tiến trong cơ chế attention.

- **Kiến trúc Decoder-Only**: Mistral 7B sử dụng kiến trúc chỉ gồm các khối decoder, phù hợp với các tác vụ sinh ngôn ngữ tự nhiên (Natural Language Generation – NLG), như hội thoại và trợ lý ảo. Mô hình được huấn luyện với hàm mất mát **Cross-Entropy Loss** theo cơ chế tự hồi tiếp (causal language modeling), trong đó mô hình học cách dự đoán token tiếp theo trong chuỗi văn bản.
- **Hiệu quả vượt trội với số tham số nhỏ**: Mặc dù chỉ có 7 tỷ tham số, Mistral 7B đạt hoặc vượt hiệu năng của các mô hình lớn hơn như LLaMA 2–13B hoặc thậm chí LLaMA 1–34B trong nhiều benchmark. Điều này giúp giảm đáng kể chi phí huấn luyện và suy luận, đồng thời thân thiện hơn với môi trường.
- **Các phiên bản Mistral 7B**: Mistral 7B được cung cấp dưới dạng mô hình *base* và *instruct*, trong đó bản instruct được tinh chỉnh để làm theo hướng dẫn. Tuy không có bản *chat* chính thức, bản base có thể sử dụng linh hoạt cho hội thoại.

- **Hiệu năng benchmark cao**: Mistral 7B đạt kết quả nổi bật trên nhiều bộ đánh giá chuẩn như MMLU (kiến thức tổng quát), AGIEval và BBH (bài thi chuẩn hóa), cùng các bộ đánh giá khác về tư duy logic, hiểu văn bản, toán học và sinh mã nguồn.



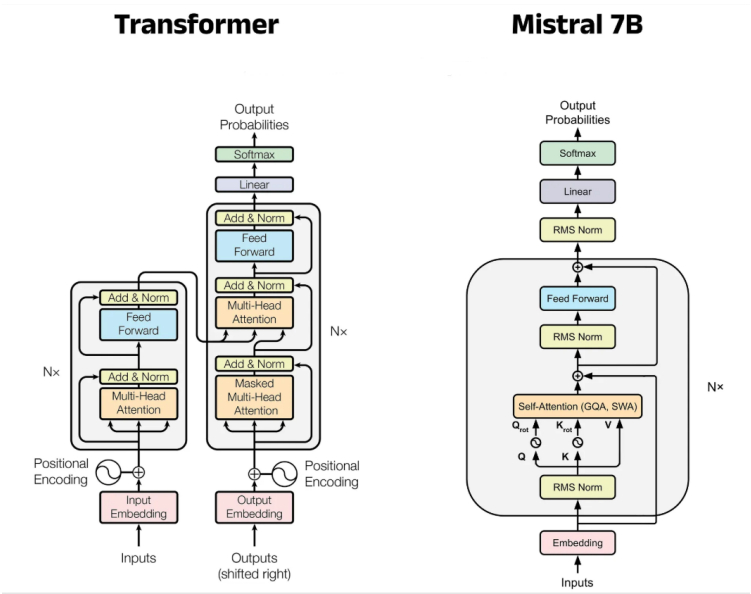
A comparison of Mistral 7B's performance with Llama and Llama 2 across a series of benchmarks [1].

Model	Modality	MMLU	HellaSwag	WinoGrande	PIQA	Arc-e	Arc-c	NQ	TriviaQA	HumanEval	MBPP	MATH	GSM8K
LLaMA 2 7B	Pretrained	44.4%	77.1%	69.5%	77.9%	68.7%	43.2%	24.7%	63.8%	11.6%	26.1%	3.9%	16.0%
LLaMA 2 13B	Pretrained	55.6%	**80.7%**	72.9%	80.8%	75.2%	48.8%	**29.0%**	**69.6%**	18.9%	35.4%	6.0%	34.3%
Code LLaMA 7B	Finetuned	36.9%	62.9%	62.3%	72.8%	59.4%	34.5%	11.0%	34.9%	**31.1%**	**52.5%**	5.2%	20.8%
Mistral 7B	Pretrained	**60.1%**	**81.3%**	**75.3%**	**83.0%**	**80.0%**	**55.5%**	**28.8%**	**69.9%**	**30.5%**	47.5%	**13.1%**	**52.1%**

A tabular view of the comparison above with the scores for each benchmark [1].

Hình 3: Hiệu năng trên các benchmark của một số mô hình

- **Các cải tiến kiến trúc chính**:
  - **RMSNorm** – Thay thế LayerNorm để chuẩn hóa nhanh và ổn định hơn.
  - **Rotary Position Embedding (RoPE)** – thay thế Absolute Positional Encoding.
  - **Grouped Query Attention (GQA)** – Cải tiến Multi-Head Attention để tiết kiệm bộ nhớ.
  - **Sliding Window Attention (SWA)** – Tăng tốc với chuỗi dài bằng cách cục bộ hóa attention.
  - **Rolling Buffer KV Cache** – Tối ưu truy xuất và lưu trữ trong inference.
  - **SwiGLU Activation Function** – Hàm kích hoạt hiệu quả cao thay thế cho ReLU trong Feed Forward sub-layers.



Hình 4: Kiến trúc Transfomer và Mistral 7B

4.6 Retrieval-Augmented Generation (RAG)

RAG (Retrieval-Augmented Generation) là phương pháp kết hợp giữa truy xuất thông tin từ dữ liệu bên ngoài và khả năng sinh ngôn ngữ của các mô hình LLM. Cách tiếp cận này giúp tăng độ chính xác, giảm ảo giác thông tin và mở rộng phạm vi hiểu biết của mô hình mà không cần huấn luyện lại.

Các mô hình LLM thường bị giới hạn về thời điểm huấn luyện, dễ sinh ra thông tin ảo (hallucination), không thể truy cập dữ liệu nội bộ và việc cập nhật kiến thức yêu cầu chi phí lớn. Điều này làm giảm độ tin cậy trong các ứng dụng thực tế.

RAG hoạt động dựa trên hai giai đoạn chính:

- **Retrieval (Truy xuất):** Khi người dùng đặt câu hỏi, hệ thống sẽ tìm kiếm các thông tin liên quan từ nguồn dữ liệu bên ngoài, như tài liệu, cơ sở dữ liệu hoặc website.
- **Sinh câu trả lời (Generation):** Các đoạn văn bản được truy xuất sẽ được kết hợp với câu hỏi để tạo thành một prompt đầu vào. Prompt này sau đó được gửi vào mô hình LLM để sinh ra câu trả lời chính xác, phù hợp với ngữ cảnh.

RAG là giải pháp hiệu quả giúp khắc phục hạn chế cố hữu của các mô hình ngôn ngữ lớn (LLM) truyền thống, bằng cách bổ sung khả năng truy xuất thông tin theo thời gian thực. Nhờ đó, hệ thống có thể cập nhật kiến thức linh hoạt, hạn chế thông tin sai lệch và dễ dàng tích hợp với dữ liệu nội bộ. Công nghệ này đặc biệt phù hợp với nhiều ứng dụng thực tế như chatbot, tìm kiếm thông minh hay trợ lý AI trong doanh nghiệp, mở ra hướng phát triển bền vững cho các hệ thống AI hiện đại.

4.7 TF-IDF (Term Frequency–Inverse Document Frequency)

TF-IDF là một kỹ thuật trong xử lý ngôn ngữ tự nhiên (NLP) và truy xuất thông tin, dùng để đánh giá tầm quan trọng của một từ trong một tài liệu tương đối với toàn bộ tập tài liệu (corpus). Phương pháp này giúp làm nổi bật những từ đặc trưng cho nội dung tài liệu, đồng thời làm giảm ảnh hưởng của các từ phổ biến.

- **Term Frequency (TF)**: Đại diện cho mức độ xuất hiện của một từ t trong tài liệu d, được tính theo công thức:

$$TF(t,d) = \frac{\text{Số lần từ } t \text{ xuất hiện trong } d}{\text{Tổng số từ trong } d}$$

  TF phản ánh tầm quan trọng cục bộ của từ trong tài liệu.

- **Inverse Document Frequency (IDF)**: Đo mức độ hiếm của từ t trong toàn bộ tập tài liệu D, tính theo công thức:

$$\text{IDF}(t,D) = \log\left(\frac{\text{Tổng số tài liệu trong tập } D}{\text{Số tài liệu chứa từ } t}\right)$$

  IDF có tác dụng giảm trọng số những từ phổ biến và tăng trọng số cho các từ hiếm gặp.

- **TF-IDF**: Là tích của TF và IDF, phản ánh tầm quan trọng tổng thể của từ trong một tài liệu cụ thể:

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D)$$

Phương pháp TF-IDF có nhiều ứng dụng quan trọng trong các hệ thống tìm kiếm, trích xuất thông tin và nhiều bài toán NLP khác. Các từ phổ biến như "the", "and", "is" thường có TF cao nhưng IDF thấp, do đó TF-IDF gần bằng 0, giúp loại bỏ nhiễu không cần thiết.

4.8 LoRA và QLoRA

Trong huấn luyện mô hình ngôn ngữ lớn (LLMs), **fine-tuning** truyền thống yêu cầu cập nhật toàn bộ tham số của mô hình, điều này dẫn đến tiêu tốn nhiều tài nguyên bộ nhớ và tính toán. Để khắc phục hạn chế này, các kỹ thuật như **LoRA (Low-Rank Adaptation)** và **QLoRA (Quantized LoRA)** ra đời, cho phép tinh chỉnh mô hình một cách hiệu quả về mặt tham số và bộ nhớ.

LoRA là kỹ thuật tinh chỉnh hiệu quả tham số bằng cách chèn thêm các ma trận hạng thấp (low-rank matrices) vào trong các lớp nhất định của mô hình, ví dụ như sau tầng attention hoặc feed-forward. Thay vì cập nhật toàn bộ trọng số của mô hình, LoRA chỉ thêm và huấn luyện một số ma trận hạng thấp trong các tầng transformer, trong khi các trọng số gốc được giữ nguyên (frozen). Cách tiếp cận này giúp giảm đáng kể số lượng tham số cần huấn luyện (thường chỉ khoảng 0.5–5% so với fine-tuning toàn phần), tiết kiệm bộ nhớ và tài nguyên tính toán, đồng thời hạn chế nguy cơ overfitting trên các tập dữ liệu nhỏ. LoRA còn có tính linh hoạt cao nhờ khả năng "hoán đổi" các adapter cho nhiều tác vụ khác nhau mà không cần huấn luyện lại toàn bộ mô hình. Sau khi huấn luyện, các adapter cũng có thể hợp nhất vào mô hình chính để đảm bảo không làm tăng độ trễ trong suy luận.

QLoRA (Quantized LoRA) là phiên bản mở rộng của LoRA, kết hợp kỹ thuật lượng tử hóa (quantization) để nén trọng số của mô hình gốc xuống định dạng 4-bit, giúp giảm mạnh yêu cầu bộ nhớ trong quá trình huấn luyện. Trong khi đó, các adapter LoRA vẫn được huấn luyện ở độ chính xác cao hơn (chẳng hạn 16-bit), giúp bù đắp các sai số do lượng tử hóa gây ra và duy trì độ chính xác của mô hình. Nhờ sự kết hợp này, QLoRA có thể fine-tune các mô hình cực lớn (hàng tỷ tham số) trên GPU tiêu dùng với VRAM rất thấp. Ngoài hiệu năng gần tương đương với LoRA hoặc fine-tuning đầy đủ trong nhiều tác vụ, QLoRA còn hỗ trợ double quantization, checkpointing và khả năng đặt adapter vào toàn bộ các tầng tuyến tính (không chỉ giới hạn ở query/key/value). Mặc dù có thể làm tăng nhẹ độ trễ do cần lượng tử hóa và giải lượng tử hóa, nhưng đổi lại, QLoRA mang lại khả năng mở rộng rất hiệu quả và tiết kiệm tài nguyên tối đa.

4.9 OCR (Optical Character Recognition)

OCR (Optical Character Recognition) là công nghệ nhận dạng ký tự quang học, cho phép chuyển đổi hình ảnh chứa văn bản (ảnh chụp hóa đơn, văn bản in, tài liệu giấy) thành dữ liệu văn bản có thể đọc, chỉnh sửa và xử lý được trên máy tính. Khi một tài liệu được scan, nó thường được lưu dưới dạng hình ảnh, và nội dung bên trong không thể tìm kiếm hoặc chỉnh sửa bằng các phần mềm xử lý văn bản thông thường. Công nghệ OCR giúp trích xuất các ký tự trong hình ảnh và chuyển đổi chúng sang dạng văn bản số, tạo điều kiện thuận lợi cho việc tự động hóa, phân tích dữ liệu, và nâng cao hiệu suất xử lý tài liệu.

Ứng dụng của OCR rất đa dạng trong thực tế. Trong ngân hàng, OCR được dùng để đọc dữ liệu từ séc và hóa đơn qua ứng dụng di động. Trong y tế, công nghệ này hỗ trợ quản lý hồ sơ bệnh nhân và truy cập thông tin nhanh chóng. Ngoài ra, OCR còn được dùng để nhận dạng biển số xe, quét danh thiếp lưu liên hệ, và tự động phân loại thư tín theo địa chỉ hoặc mã vùng trong ngành bưu chính.

Quy trình hoạt động của OCR bắt đầu bằng việc quét tài liệu vật lý thành ảnh số. Phần mềm OCR xử lý ảnh thành bản trắng đen, xác định vùng nền sáng và ký tự tối. Sau đó, từng ký tự hoặc từ được phân tích và nhận dạng bằng hai kỹ thuật chính: nhận dạng theo mẫu (pattern recognition) hoặc theo đặc trưng (feature recognition), cho ra văn bản số có thể tìm kiếm và chỉnh sửa.

Ưu điểm của OCR là chuyển văn bản in hoặc viết tay thành dạng số, dễ lưu trữ, tìm kiếm và xử lý. Công nghệ này giúp tiết kiệm thời gian, chi phí nhập liệu thủ công và hỗ trợ lưu trữ tài liệu số. Nhiều hệ thống còn hỗ trợ đa ngôn ngữ, phù hợp với tổ chức hoạt động toàn cầu.

Hạn chế của OCR là vẫn có thể xảy ra lỗi nhận dạng, nên thường cần kiểm tra lại, nhất là với tài liệu chất lượng thấp hoặc chữ viết tay. Công nghệ này cũng đặt ra lo ngại về bảo mật khi dữ liệu nhạy cảm bị trích xuất trái phép. Ngoài ra, phần mềm OCR chất lượng cao có thể tốn kém và thường thiếu khả năng hiểu ngữ cảnh, dễ gây nhầm lẫn nội dung.

OCR là một công cụ mạnh mẽ giúp kết nối văn bản in với thế giới số, hỗ trợ hiệu quả trong việc truy cập, xử lý và quản lý thông tin. Trong thời đại số hóa, OCR vẫn là công nghệ then chốt trong việc biến tài liệu vật lý thành dữ liệu thông minh.

5 Mô hình giải pháp đề xuất

5.1 Tổng quan

Hệ thống **trợ lý toán học** được thiết kế nhằm đáp ứng các nhu cầu cốt lõi trong quá trình học tập toán học, kết hợp các công nghệ xử lý ngôn ngữ tự nhiên (NLP), nhận dạng ký tự quang học (OCR), mô hình ngôn ngữ lớn (LLM), và truy xuất tri thức dựa trên độ tương đồng văn bản. Ba chức năng chính của hệ thống bao gồm:

- **Chức năng 1 – Giải toán từ văn bản:** Người dùng nhập bài toán dưới dạng văn bản. Hệ thống sử dụng các mô hình Mistral 7B đã được tinh chỉnh trên tập dữ liệu MetaMath để sinh ra lời giải từng bước.
- **Chức năng 2 – Giải toán từ hình ảnh:** Người dùng cung cấp hình ảnh bài toán (ảnh chụp, scan). Hệ thống sử dụng OCR để trích xuất nội dung và chuyển đến pipeline giải toán tương tự như chức năng 1.
- **Chức năng 3 – Tra cứu định nghĩa toán học:** Người dùng nhập khái niệm toán học cần tra cứu. Hệ thống sử dụng phương pháp truy xuất tri thức dựa trên độ tương đồng văn bản (TF-IDF kết hợp cosine similarity) để tìm kiếm định nghĩa phù hợp nhất từ cơ sở dữ liệu tri thức đã được chuẩn hóa.

Với ba chức năng cốt lõi được triển khai chặt chẽ, hệ thống hướng đến việc mang lại trải nghiệm học tập dễ dàng, hiệu quả, có khả năng diễn giải rõ ràng, đồng thời có thể mở rộng linh hoạt theo nhu cầu người dùng và quy mô dữ liệu trong tương lai.

5.2 Thu thập và xử lý dữ liệu

- **Chức năng giải toán bằng văn bản và hình ảnh:** Dữ liệu chính được sử dụng là tập **MetaMathQA** gồm 395.000 mẫu dữ liệu. Mỗi mẫu dữ liệu bao gồm câu hỏi, lời giải chi tiết, suy luận logic và kết quả cho câu hỏi đó. Dữ liệu được chia nhỏ thành bốn tập con (100k, 100k, 100k, 95k) để huấn luyện tinh chỉnh song song các mô hình.
- **Chức năng tra cứu khái niệm toán học:** Nguồn dữ liệu được thu thập từ các nguồn uy tín như Wikipedia, giáo trình toán học, v.v. Các khái niệm được làm sạch, rút gọn và tóm tắt nhằm đảm bảo dễ hiểu và dễ tra cứu. Dữ liệu sau đó được chuẩn hóa dưới định dạng `JSON`, gồm các trường: `term` (tên khái niệm), `definition` (định nghĩa ngắn gọn), giúp dễ dàng truy xuất và tăng độ tin cậy.

5.3 Thiết kế, huấn luyện và đánh giá hiệu suất mô hình

5.3.1 Quy trình huấn luyện mô hình

Quá trình huấn luyện và tinh chỉnh mô hình được thực hiện theo sơ đồ ở Hình 5 bên dưới. Cụ thể:

1. **Tiền xử lý dữ liệu:** Dữ liệu từ bộ MetaMath được giữ nguyên định dạng gốc, không áp dụng bước chuẩn hóa văn bản hay xử lý cú pháp phức tạp. Mỗi mẫu dữ liệu bao gồm câu hỏi, lời giải chi tiết, suy luận logic và kết quả cho câu hỏi đó. Các mẫu này được kiểm tra định dạng tổng quát (không rỗng, đúng cấu trúc JSON, v.v) nhằm đảm bảo tính nhất quán khi đưa vào quá trình huấn luyện.

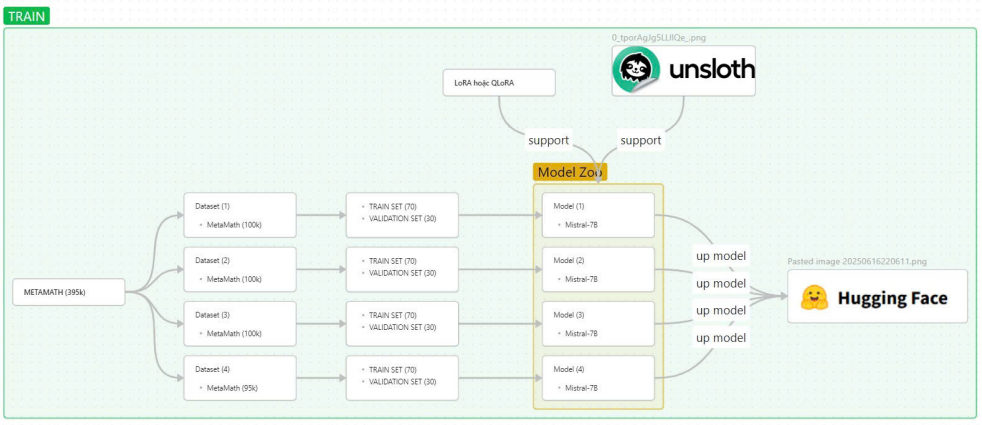
2. **Chia tập dữ liệu:** Dữ liệu được chia thành 4 tập nhỏ, sau đó từng tập được phân chia theo tỷ lệ 70:30 thành tập huấn luyện (train) và tập kiểm định (validation). Việc chia đều và độc lập các tập giúp tăng độ đa dạng và khả năng tổng quát của mô hình sau khi fine-tune.

3. **Tinh chỉnh mô hình bằng QLoRA:** Áp dụng kỹ thuật **QLoRA** với `rank=32` để fine-tune mô hình **Mistral 7B Instruct v0.3**. QLoRA cho phép lượng tử hóa mô hình ở mức 4-bit để tiết kiệm bộ nhớ, đồng thời chỉ cập nhật các tham số phụ nhỏ (low-rank adapters), rút ngắn thời gian huấn luyện và giảm yêu cầu phần cứng.

4. **Tối ưu tốc độ với Unsloth:** Thư viện `Unsloth` được sử dụng để tăng tốc độ xử lý và giảm đáng kể mức tiêu thụ VRAM. Unsloth hỗ trợ cơ chế chia lô hiệu quả, sử dụng context dài hơn và tối ưu hóa gradient checkpointing giúp huấn luyện ổn định với batch size lớn trên GPU.

5. **Cấu hình huấn luyện:** Mỗi mô hình được huấn luyện bằng phương pháp **Supervised Fine-Tuning (SFT)** với khoảng **1200 - 1500 bước (steps)**. Learning rate được thiết lập là `2e-4`, sử dụng thuật toán tối ưu `AdamW 8-bit` cùng lịch điều chỉnh learning rate dạng `cosine` hoặc `linear` tùy vào cách huấn luyện mô hình. Batch size trên mỗi thiết bị là `2`, kết hợp với `gradient accumulation` trong 4 bước, tạo ra tổng batch size hiệu dụng là `8`. Huấn luyện được thực hiện trên chuỗi đầu vào có độ dài tối đa là `2048 token`, và không sử dụng kỹ thuật packing nhằm đảm bảo độ chính xác cho các chuỗi toán học dài. Các log huấn luyện được ghi lại sau mỗi 10 bước để theo dõi hiệu suất.

6. **Đưa mô hình lên nền tảng Hugging Face:** Sau khi kiểm tra và đánh giá nội bộ, các mô hình được upload lên nền tảng **Hugging Face**. Việc đưa lên Hugging Face cũng giúp dễ dàng tái sử dụng, triển khai và tích hợp vào các hệ thống downstream khác.

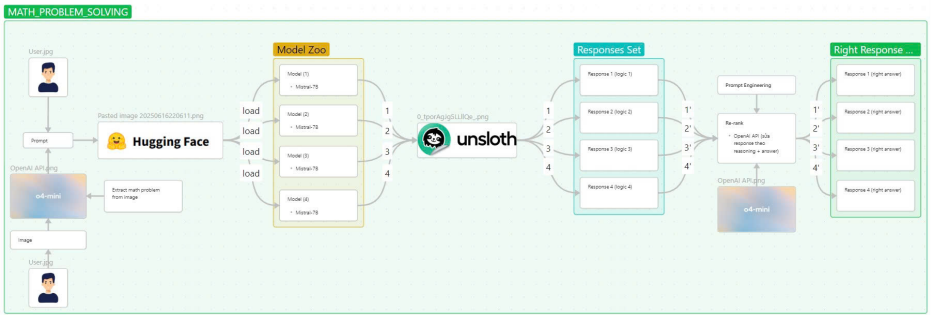


Hình 5: Sơ đồ huấn luyện và tinh chỉnh mô hình giải toán

5.3.2 Quy trình triển khai hệ thống và tổ chức chức năng

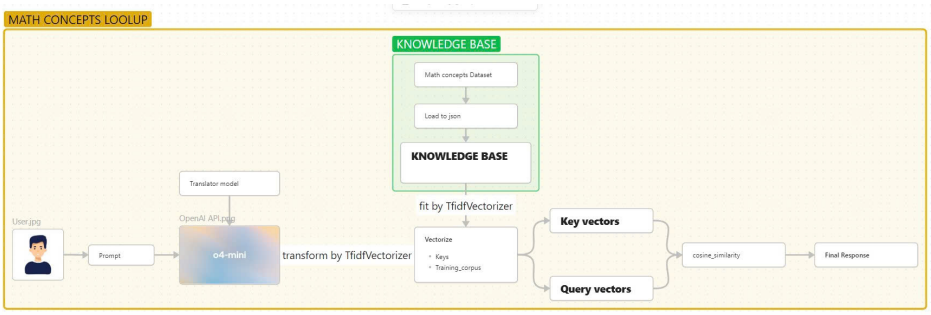
Các chức năng chính được triển khai độc lập theo pipeline chuyên biệt như mô tả trong Hình 6 và 7. Cấu trúc hệ thống đảm bảo khả năng mở rộng, tái sử dụng và chia tải giữa các mô hình.

- **Chức năng 1: Giải toán từ văn bản**
  - Người dùng nhập bài toán dạng văn bản thông qua giao diện web
  - Câu hỏi từ người dùng được gửi đồng thời tới **4 mô hình Mistral 7B**, mỗi mô hình phản hồi với một cách giải khác nhau (logic riêng biệt).
  - Các phản hồi được thu thập thành `Response Set`, mỗi phản hồi thể hiện một chuỗi suy luận logic độc lập.
  - Module **Re-ranking** sử dụng mô hình `o4-mini` để đánh giá và sắp xếp các lời giải theo tiêu chí chính xác và tính hợp lý của lập luận.
  - Kết quả cuối cùng được chọn ra là phản hồi có độ tin cậy cao nhất, được chuẩn hóa lại và trình bày chi tiết lại để hiển thị cho người dùng.
- **Chức năng 2: Giải toán từ hình ảnh**
  - Người dùng tải lên hình ảnh (định dạng `.jpg`, `.png`, v.v) chứa bài toán.
  - Ảnh được xử lý bằng OCR (`o4-mini`) để trích xuất đề bài – bao gồm cả văn bản tự nhiên và công thức toán học (nếu có).
  - Nội dung được trích xuất sẽ được chuyển sang pipeline xử lý như **Chức năng 1** để tiến hành giải bài toán.



Hình 6: Sơ đồ tổng thể chức năng Giải toán

- **Chức năng 3: Tra cứu khái niệm toán học**
  - Người dùng nhập truy vấn dưới dạng ngôn ngữ tự nhiên (có thể tiếng Việt hoặc tiếng Anh) trên giao diện web.
  - Nếu truy vấn không phải tiếng Anh, hệ thống sử dụng mô hình dịch tự động để chuyển sang tiếng Anh.
  - Truy vấn được vector hóa bằng `TF-IDF Vectorizer` để tạo thành vector truy vấn.
  - Tập dữ liệu khái niệm toán học (đã được xử lý và chuẩn hóa ở định dạng `JSON`) cũng được vector hóa tương tự để tạo ra các vector khóa.
  - Áp dụng phép đo `cosine similarity` để tìm ra khái niệm phù hợp nhất trong cơ sở tri thức.
  - Định nghĩa tương ứng được hệ thống trả về cho người dùng.

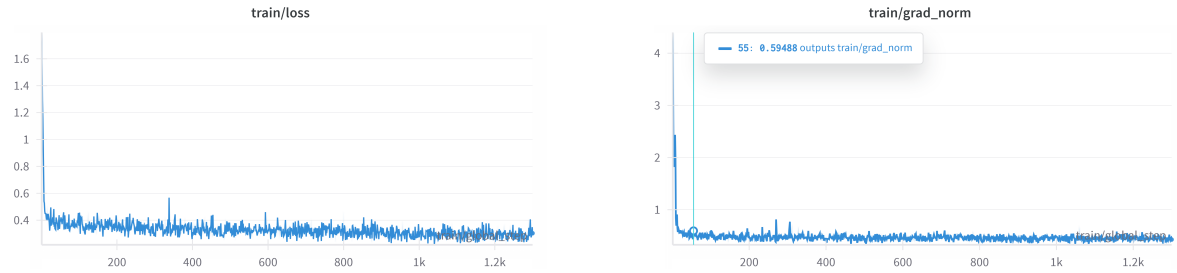


Hình 7: Sơ đồ tổng thể chức năng Tra cứu khái niệm

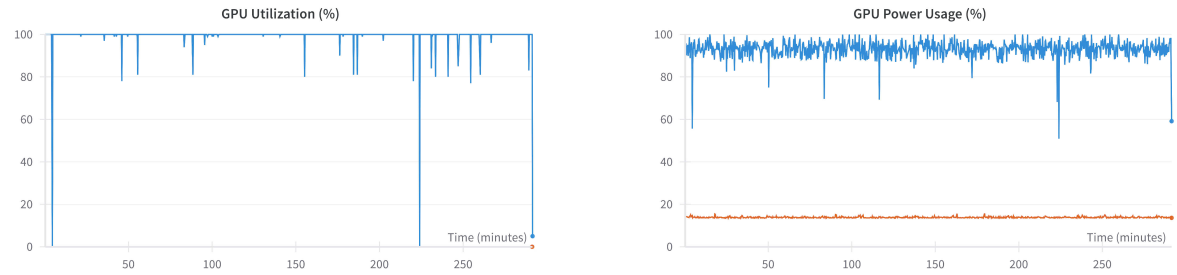
5.3.3 Đánh giá hiệu suất mô hình

Kết quả huấn luyện của một mô hình Mistral cụ thể như sau:

```
TrainOutput(
    global_step=1200,
    training_loss=0.3185,
    metrics={
        'train_runtime': 25627.21,
        'train_samples_per_second': 0.749,
        'train_steps_per_second': 0.047,
        'total_flos': 2.17e+17,
        'train_loss': 0.3185
    }
)
```



Hình 8: Thông số huấn luyện một mô hình Mistral



Hình 9: Thông số hiệu suất sử dụng GPU

Thông qua các biểu đồ và thông số huấn luyện trên, có thể rút ra một số nhận định về hiệu suất của mô hình như sau:

- **Loss hội tụ tốt:** Giá trị `train_loss` = 0.3185 cho thấy mô hình đã học hiệu quả trên tập huấn luyện, với sai số thấp. Biểu đồ "train/loss" cũng xác nhận điều này, cho thấy loss giảm nhanh chóng ở các bước đầu và ổn định ở mức thấp (khoảng 0.3 - 0.4) trong suốt quá trình huấn luyện, không có dấu hiệu overfitting rõ rệt.

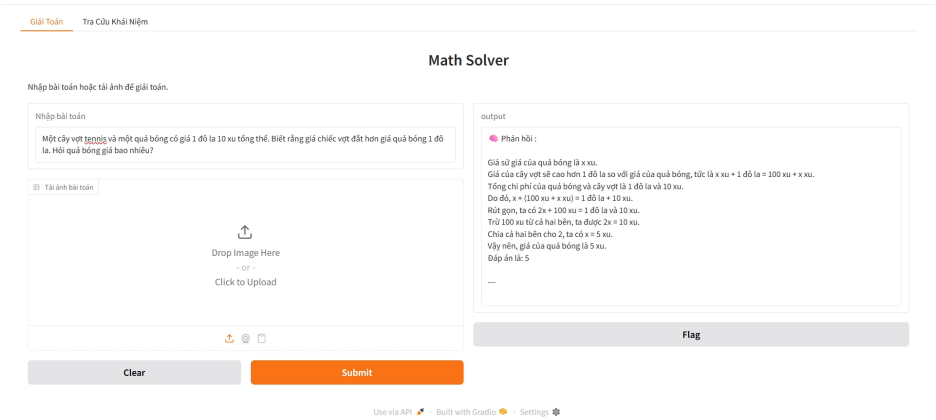
- **Gradient chuẩn (grad_norm) ổn định:** Biểu đồ "train/grad_norm" cho thấy giá trị chuẩn gradient giảm nhanh ở các bước đầu và sau đó duy trì ở mức thấp và ổn định (dưới 1). Điều này chỉ ra rằng quá trình tối ưu hóa diễn ra ổn định, không có hiện tượng gradient bùng nổ (exploding gradients) hay gradient biến mất (vanishing gradients), góp phần vào sự hội tụ tốt của mô hình.
- **Tốc độ huấn luyện ổn định:** Với `train_steps_per_second = 0.047`, mô hình đạt hiệu suất hợp lý trên phần cứng hỗ trợ lượng tử hóa 4-bit và gradient checkpointing (thông qua Unsloth).
- **Độ phức tạp mô hình cao:** Tổng số phép tính dấu chấm động thực hiện được (`total_flos = 2.17e+17`) là rất lớn, cho thấy mô hình được huấn luyện đủ sâu để nắm bắt các đặc trưng phức tạp trong dữ liệu toán học.
- **Hiệu suất sử dụng GPU cao:** GPU duy trì mức sử dụng gần như tối đa ($\approx 100\%$) trong suốt quá trình huấn luyện, chỉ thỉnh thoảng có độ rơi nhẹ do tải dữ liệu hoặc lưu checkpoint. Điều này cho thấy mô hình sử dụng tài nguyên GPU rất hiệu quả, không bị idle quá lâu.
- **Công suất tính toán ổn định:** Mức tiêu thụ điện năng của GPU ổn định ở khoảng 90–100%, ngoại trừ một vài điểm rơi ngắn, phản ánh quá trình tính toán diễn ra liên tục, đồng thời xác nhận rằng mô hình đang huấn luyện ở công suất gần như tối đa.
- **Kết quả đầu ra có tính nhất quán cao:** Dựa trên quá trình đánh giá thủ công và thử nghiệm đầu ra trên các ví dụ kiểm tra, mô hình tạo ra lời giải nhất quán và sát nghĩa với lời giải chuẩn.

5.3.4 Thử nghiệm và so sánh

- **Thử nghiệm chức năng 1: Giải toán từ văn bản**

  Với chức năng này, nhóm đưa vào một bài toán bằng tiếng Việt dưới dạng văn bản: *"Một cây vợt tennis và một quả bóng có giá 1 đô la 10 xu tổng thể. Biết rằng giá chiếc vợt đắt hơn giá quả bóng 1 đô la. Hỏi quả bóng giá bao nhiêu?"*. Câu hỏi được đưa vào hệ thống xử lý ngôn ngữ tự nhiên, mã hóa và chuyển thành input cho các mô hình Mistral 7B đã được tinh chỉnh với dữ liệu MetaMathQA thông qua kỹ thuật QLoRA.

  Các mô hình sau đó tạo ra lời giải theo từng bước, kèm theo biểu thức toán học trung gian và kết luận rõ ràng. Quá trình xử lý được mô tả như trong Hình 10.

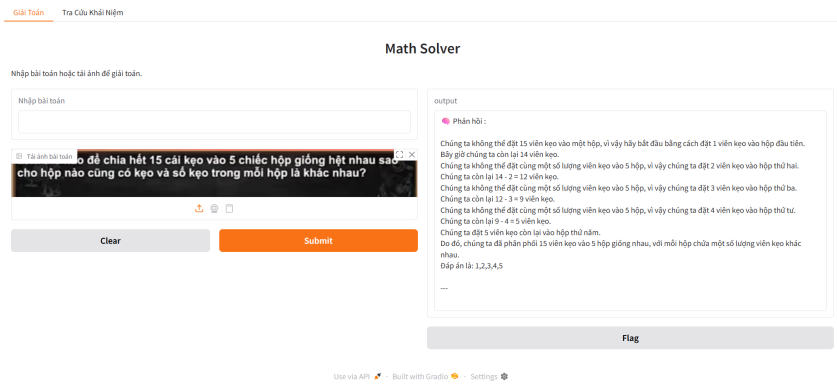


Hình 10: Thử nghiệm chức năng Giải toán từ văn bản

Kết quả cho thấy hệ thống không chỉ đưa ra đáp án đúng (*5 xu*), mà còn trình bày rõ ràng từng bước lập luận và suy luận toán học. Cách diễn giải tự nhiên, thân thiện, rất phù hợp cho người dùng.

- **Thử nghiệm chức năng 2: Giải toán từ hình ảnh**
  Với chức năng này, nhóm sẽ đưa vào một hình ảnh chứa đề bài môn Toán. Ví dụ, câu hỏi *"Làm thế nào để chia hết 15 cái kẹo vào 5 chiếc hộp giống hệt nhau sao cho hộp nào cũng có kẹo và số kẹo trong mỗi hộp là khác nhau?"* được nhập dưới dạng ảnh chứa văn bản.

  Hệ thống sử dụng OCR (Optical Character Recognition) để trích xuất nội dung văn bản từ ảnh, sau đó tiếp tục quá trình xử lý tương tự như chức năng 1. Kết quả được trình bày trong Hình 11.

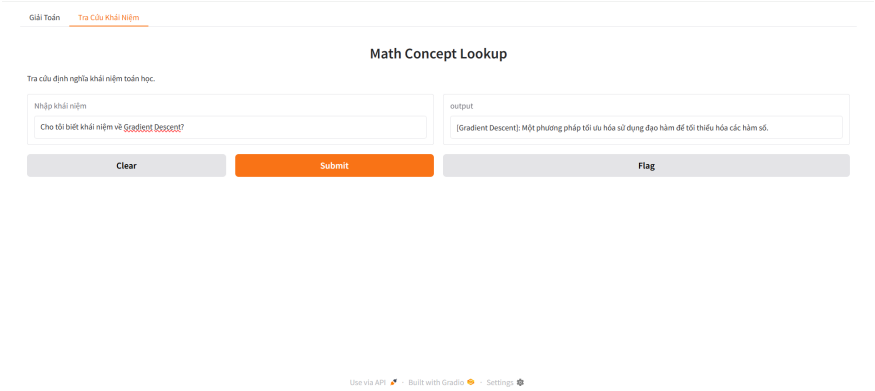


Hình 11: Thử nghiệm chức năng Giải toán từ hình ảnh

Kết quả cho thấy hệ thống hoạt động ổn định, nhận diện chính xác đề bài và trả về lời giải hợp lý, có đầy đủ bước phân tích, suy luận, trình bày tuần tự.

- **Thử nghiệm chức năng 3: Tra cứu khái niệm**
  Với chức năng này, hệ thống cho phép người dùng hỏi về các khái niệm toán học hoặc học thuật, chẳng hạn như: *"Cho tôi biết khái niệm về Gradient Descent"*. Câu hỏi được mã hóa và đưa vào pipeline xử lý thông qua cơ chế kết hợp giữa mô hình sinh ngôn ngữ và truy xuất ngữ nghĩa (semantic retrieval).

  Kết quả thu được được trình bày trong Hình 12. Câu trả lời được tạo ra ngắn gọn, súc tích và đúng trọng tâm, phù hợp với người dùng.



Hình 12: Thử nghiệm chức năng Tra cứu khái niệm

- **So sánh với các công cụ hiện có trên thị trường:**
  - **Chức năng giải toán (văn bản, hình ảnh):** So với các công cụ như WolframAlpha hay Symbolab, hệ thống sử dụng mô hình Mistral 7B được tinh chỉnh từ tập dữ liệu MetaMathQA có ưu điểm vượt trội về khả năng giải thích lời giải theo ngôn ngữ tự nhiên, chặt chẽ về mặt lập luận logic. Mô hình của hệ thống đưa ra lời giải chi tiết từng bước, phù hợp với người học muốn hiểu rõ cách giải chứ không chỉ là biết kết quả.
  - **Chức năng tra cứu khái niệm:** So với Google Search, hệ thống sử dụng truy hồi theo ngữ nghĩa (semantic retrieval) nên cho kết quả tốt tương tự, và diễn đạt thân thiện, đặc biệt hữu ích với người học.

6 Ứng dụng và demo thực tế

6.1 Mức độ tích hợp mô hình vào ứng dụng web

Hệ thống được triển khai dưới dạng ứng dụng web, sử dụng **Gradio** nhằm tích hợp hiệu quả các mô hình ngôn ngữ lớn (LLM) cùng với các pipeline xử lý chuyên biệt cho từng chức năng. Việc triển khai hướng đến khả năng mở rộng, dễ sử dụng và phục vụ thời gian thực.

- **Tích hợp pipeline giải toán:**
  - Người dùng có thể nhập đề bài dưới dạng văn bản hoặc tải ảnh đề toán.
  - Nếu là ảnh, hệ thống sử dụng `o4-mini` để thực hiện OCR và trích xuất văn bản.
  - Văn bản được gửi đến 4 mô hình Mistral-7B đã fine-tune độc lập trên các tập khác nhau của bộ dữ liệu MetaMathQA.
  - Kết quả từ các mô hình được gom lại và đánh giá lại bằng mô hình re-ranking (dựa trên lập luận và đáp án).
  - Lời giải tốt nhất được chọn lọc và hiển thị chi tiết cho người dùng.
- **Tích hợp chức năng tra cứu khái niệm toán học:**
  - Truy vấn người dùng sẽ được dịch sang tiếng Anh nếu cần thiết.
  - Truy vấn sẽ được vector hóa bằng `TF-IDF` và so sánh cosine với tập khái niệm toán học đã được chuẩn hóa, lưu trữ dưới dạng JSON.
  - Định nghĩa phù hợp với yêu cầu sẽ được phản hồi cho người dùng.
- **Triển khai đơn giản qua Gradio:**
  - Tất cả chức năng đều được tích hợp trực tiếp thông qua các hàm xử lý Python gắn với giao diện `gr.Interface`.
  - Ứng dụng hỗ trợ hai giao diện chính: *Giải Toán* và *Tra cứu Khái niệm*, được tổ chức bằng `gr.TabbedInterface`.
  - Việc triển khai không yêu cầu backend phức tạp, toàn bộ logic được đóng gói trong một script đơn giản, dễ chạy.
- **Tương tác thời gian thực:**
  - Giao diện Gradio cho phép người dùng nhập văn bản trực tiếp hoặc tải ảnh lên.
  - Toàn bộ quá trình xử lý (OCR, logic, sinh lời giải, re-rank) được phản hồi theo thời gian thực.

Hệ thống cho thấy khả năng tích hợp hiệu quả các mô hình học sâu vào ứng dụng thực tế, đồng thời đảm bảo được tốc độ, khả năng mở rộng và trải nghiệm người dùng mượt mà.

6.2 Giao diện và trải nghiệm người dùng

Giao diện người dùng được thiết kế tối giản, dễ sử dụng, hướng đến người dùng là học sinh và sinh viên. Hệ thống hỗ trợ hai chức năng chính thông qua các tab riêng biệt:

- **Giao diện đơn giản, trực quan:**
  - Bố cục trang web phân thành hai tab chính: **Giải Toán** và **Tra cứu Khái niệm**.
  - Hỗ trợ nhập trực tiếp hoặc tải ảnh đề bài với nhiều định dạng khác nhau.
- **Phản hồi rõ ràng, dễ hiểu:**
  - Lời giải được hiển thị mạch lạc, trình bày theo từng bước với định dạng dễ đọc.
  - Kết quả tra cứu khái niệm được hiển thị ngay bên cạnh ô nhập câu hỏi, kèm theo tên khái niệm và định nghĩa ngắn gọn.
  - Toàn bộ kết quả đều được trình bày dưới dạng văn bản thuần, giúp người dùng dễ dàng sao chép hoặc ghi chú lại.
- **Tốc độ xử lý:**
  - Truy vấn văn bản hoặc ảnh: **30–60 giây**, tùy vào độ dài và độ phức tạp bài toán.
  - Tra cứu định nghĩa toán học: **3 - 5 giây**.

Từ những điều ở trên, có thể thấy rằng hệ thống không chỉ đạt hiệu quả cao về mặt kỹ thuật mà còn mang lại trải nghiệm người dùng trực quan, thân thiện và đáng tin cậy trong môi trường ứng dụng thực tế.

7 Hướng phát triển trong tương lai

Trong quá trình xây dựng hệ thống hỗ trợ giải toán, ngoài việc đảm bảo tính chính xác và tiện ích ở thời điểm hiện tại, việc định hướng phát triển lâu dài cũng đóng vai trò quan trọng nhằm đáp ứng nhu cầu học tập ngày càng đa dạng của người dùng. Một số hướng phát triển tiềm năng được đề xuất như sau:

- **Nâng cấp mô hình sinh lời giải:** Xây dựng pipeline huấn luyện mới dựa trên các phiên bản tiên tiến hơn như `Mistral 7B DPO` hoặc `Mixtral 8x7B` để tăng khả năng xử lý các bài toán dài và phức tạp (long-form reasoning).
- **Tích hợp nhận diện công thức toán học nâng cao:** Kết hợp thêm các công cụ như `MathPix OCR` để nhận diện chính xác biểu thức viết tay hoặc công thức LaTeX trong ảnh, bổ sung cho pipeline giải toán từ hình ảnh.
- **Mở rộng cơ sở tri thức định nghĩa:** Thu thập thêm dữ liệu từ các nguồn toán học uy tín khác nhau (Wikipedia, giáo trình, arXiv, v.v.), đồng thời kết hợp truy xuất ngữ nghĩa (`semantic retrieval`) nhằm hiểu và phản hồi cả với các biểu đạt tương đương về mặt ý nghĩa.
- **Tăng cường tính tương tác:** Bổ sung chức năng nhập liệu bằng giọng nói và sinh lời giải bằng lời nói, nhằm hỗ trợ học sinh nhỏ tuổi hoặc người dùng đặc biệt.
- **Chuẩn hóa và triển khai API:** Xây dựng các `RESTful API endpoint` cho từng chức năng chính (giải toán, tra cứu, OCR), nhằm hỗ trợ tích hợp hệ thống vào các nền tảng học trực tuyến hoặc chatbot giáo dục trong tương lai.

Tài liệu tham khảo

[1] UnslothAI, "Unsloth Notebooks". Available at: https://github.com/unslothai/notebooks

[2] Pham Nam, "Tổng quan về Chatbot". Available at: https://viblo.asia/p/tong-quan-ve-chatbot-yMnKMByaZ7P

[3] VNTECHIES, "Tất cả những gì bạn cần biết về Chatbot". Available at: https://vntechies.dev/blog/solutions/tat-ca-nhung-gi-ban-can-biet-ve-chatbot

[4] FPT Digital, "Xử lý ngôn ngữ tự nhiên: Công nghệ giúp máy tính hiểu và giao tiếp với con người". Available at: https://digital.fpt.com/dxarticles/xu-ly-ngon-ngu-tu-nhien.html

[5] VinBigData, "Xử lý ngôn ngữ tự nhiên: Bài toán & công cụ bạn nên biết". Available at: https://vinbigdata.com/cong-nghe-giong-noi/xu-ly-ngon-ngu-tu-nhien-bai-toan-cong-cu-ban-nen-biet.html

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need". Available at: https://arxiv.org/abs/1706.03762

[7] 200Lab, "Large Language Model là gì ? Giải thích dễ hiểu". Available at: https://200lab.io/blog/large-language-model-la-gi

[8] Bradney Smith, "Mistral 7B Explained: Towards More Efficient Language Models". Available at: https://medium.com/data-science/mistral-7b-explained-towards-more-efficient-language-models-7f9c6e6b7251#bc24

[9] Albert Q. Jiang et al., "Mistral 7B". Available at: https://arxiv.org/abs/2310.06825

[10] 200Lab, "RAG (Retrieval-Augmented Generation) là gì? Giải thích dễ hiểu cho Developer". Available at: https://200lab.io/blog/rag-la-gi

[11] GeeksforGeeks, "Understanding TF-IDF (Term Frequency-Inverse Document Frequency)". Available at: https://www.geeksforgeeks.org/machine-learning/understanding-tf-idf-term-frequency-inverse-document-frequency/

[12] GeeksforGeeks, "Fine-Tuning using LoRA and QLoRA". Available at: https://www.geeksforgeeks.org/deep-learning/fine-tuning-using-lora-and-qlora/

[13] GeeksforGeeks, "What is Optical Character Recognition (OCR)?". Available at: https://www.geeksforgeeks.org/computer-science-fundamentals/what-is-optical-character-recognition-ocr/