TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Hệ thống Hỏi Đáp Open Domain tiếng Việt

LƯƠNG SƠN TỊNH
tinh.ls194186@sis.hust.edu.vn

Ngành: Khoa học máy tính

Giảng viên hướng dẫn: ThS. Ngô Văn Linh _____

Chữ kí GVHD

Khoa: Khoa Học Máy Tính

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 07/2023

LỜI CẢM ƠN

Em xin bày tỏ sự cảm ơn sâu sắc đến thầy, Thạc sĩ Ngô văn Linh với sự hướng dẫn và chỉ bảo ân cần không chỉ trong quá trình thực hiện đồ án mà còn trong quá trình tại Trường Công nghệ thông tin và Truyền Thông, Đại Học Bách Khoa Hà Nội. Đồng thời, em cũng cảm ơn tất cả các thầy cô giảng dạy ở Đại học Bách Khoa Hà Nội đã đưa đến cho em những kiến thức và bài học để em trưởng thành hơn. Nhờ sự giúp đỡ của các thầy cô mà em đã chững trạc hơn rất là nhiều.

Em xin gửi lời cảm ơn đến gia đình và người thân vì luôn là điểm tựa về tinh thần và luôn luôn ủng hộ quyết định của em. Những điều này tạo động lực để em phấn đấu mỗi ngày.

Em xin cảm ơn những người bạn, những người anh chị em mà em đã gặp làm quen, giúp đỡ em trong 4 năm Đại học tại Đại học Bách khoa Hà Nội, đặc biệt là những người bạn ở lớp Khoa học máy tính 04. Sự giúp sức của các bạn đã giúp em định hướng con đường mình muốn đi và hoàn thiện bản thân mình trong quá trình học đại học này.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong lĩnh vực xử lý ngôn ngữ tự nhiên, Bài toán Hỏi đáp đóng vai trò quan trọng và có nhiều ứng dụng đa dạng như trong hệ thống chăm sóc khách hàng và chatbot. Tuy nhiên, bài toán này đối diện với nhiều thách thức trong tiếng Việt, như việc hiện tại các phương pháp giải quyết thường dựa vào trích xuất câu trả lời. Điều này gây ra sự không tự nhiên và ảnh hưởng đến trải nghiệm người dùng khi tích hợp vào các ứng dụng chăm sóc khách hàng.

Gần đây, nhiều người quan tâm đến việc sử dụng các mô hình sinh, đặc biệt là các mô hình ngôn ngữ lớn được huấn luyện bằng dữ liệu chỉ dẫn như ChatGPT, Bard, Alpaca,... để giải quyết bài toán Hỏi đáp và các vấn đề trong Xử lý ngôn ngữ tự nhiên. Tuy nhiên, việc sử dụng ChatGPT yêu cầu người dùng phải trả phí, một số mô hình khác không dễ dàng sử dụng (Bard, Claude) hoặc không hỗ trợ tiếng Việt (Flan-T5, Alpaca). Một số khác như mT0 và Bloomz không cho kết quả tốt bằng ChatGPT và vẫn dựa vào trích xuất câu trả lời.

Để giải quyết các vấn đề này, đồ án này tập trung vào việc xử lý các bộ dữ liệu dạng chỉ dẫn để huấn luyện các mô hình ngôn ngữ khác nhau như mô hình Bartpho (tiếng Việt) và các mô hình mT0 (đa ngôn ngữ). Mục tiêu là nâng cao khả năng hiểu và trả lời của mô hình để có thể đưa ra các câu trả lời tự nhiên hơn. Bên cạnh đó, việc huấn luyện các mô hình có số lượng tham số lớn như mT0-xxl với 13 tỷ tham số yêu cầu áp dụng các phương pháp tối ưu trên phần cứng có hạn.

Như vậy, thông qua đồ án này, em mong muốn giải quyết những thách thức trong bài toán Hỏi đáp tiếng Việt và đóng góp cho việc phát triển các mô hình ngôn ngữ đạt hiệu quả cao và đáp ứng nhu cầu thực tiễn trong các ứng dụng xử lý ngôn ngữ tự nhiên.

Hà Nội, ngày tháng năm

Tác giả ĐATN

Họ và tên sinh viên

ABSTRACT

In the field of Natural Language Processing, the Question-Answering problem plays a crucial role and has diverse applications such as in customer care systems and chatbots. However, this problem faces several challenges when dealing with the Vietnamese language. Currently, the prevailing methods for solving this problem rely heavily on answer extraction, resulting in unnatural responses and impacting the user experience when integrated into customer care applications.

Recently, there has been growing interest in using generative models, especially large language models trained with supervised data, such as ChatGPT, Bard, Alpaca, to address the Question-Answering problem and other challenges in Natural Language Processing. Nevertheless, using ChatGPT often requires payment from users, and some other models like Bard and Claude are not user-friendly for Vietnamese or lack support for the language (Flan-T5, Alpaca). Additionally, models like mT0 and Bloomz do not perform as well as ChatGPT and still rely on answer extraction.

To tackle these issues, this thesis focuses on processing instruction-like datasets to train various language models, such as Bartpho (Vietnamese) and mT0 models (multilingual). The objective is to enhance the understanding and response capabilities of these models to provide more natural answers. Furthermore, training models with a large number of parameters, like mT0-xxl with 13 billion parameters, necessitates implementing optimization methods on limited hardware resources.

In conclusion, this thesis aims to address the challenges in Vietnamese Question-Answering and contribute to the development of highly effective language models that meet practical needs in natural language processing applications.

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	**Ý nghĩa**
CV	Computer Vision (Thị giác máy tính)
DL	Deep Learning (Học sâu)
LLM	Large Language Model (Mô hình ngôn ngữ lớn)
LSTM	Long short-term memory (Mạng bộ nhớ dài ngắn hạn)
ML	Machine Learning (Học máy)
MRC	Machine Reading Comprehension (Máy đọc hiểu)
NLG	Natural Language Generation (Sinh ngôn ngữ tự nhiện)
NLP	Natural Language Processing (Xử lí ngôn ngữ tự nhiên)
PEFT	Parameter-Efficient Fine-tuning (Tinh chỉnh Tham số Hiệu quả)
RNN	Recurrent Neural Network (Mạng neural hồi quy)

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong những năm gần đây, lĩnh vực Trí tuệ nhân tạo (AI) và Xử lý ngôn ngữ tự nhiên (NLP) đã đạt được tiến bộ đáng kể. Nhờ sự phát triển của các công nghệ mới như Mạng nơ-ron nhân tạo (ANN), Học sâu (deep learning) và dữ liệu lớn (big data), các hệ thống AI hiện nay đã có khả năng thực hiện nhiều nhiệm vụ trước đây được xem là chỉ có con người mới có thể làm, như dịch văn bản sang các ngôn ngữ khác nhau, trả lời câu hỏi và viết kịch bản.

Những tiến bộ này đã làm cho các hệ thống AI được triển khai rộng rãi trong nhiều lĩnh vực khác nhau, bao gồm:

- Hỗ trợ khách hàng: Các hệ thống AI được sử dụng để trả lời các câu hỏi của khách hàng, giải quyết các vấn đề và cung cấp hỗ trợ kỹ thuật.
- Giáo dục: Các hệ thống AI được sử dụng để cá nhân hóa việc học, cung cấp phản hồi và đánh giá cho học sinh.
- Tài chính: Các hệ thống AI được sử dụng để giao dịch chứng khoán, quản lý rủi ro và phân tích dữ liệu tài chính.

Trong lĩnh vực xử lý ngôn ngữ tự nhiên, bài toán Hỏi đáp (Question Answering - QA) đóng một vai trò quan trọng. Bài toán này tập trung vào việc xây dựng các hệ thống tự động trả lời câu hỏi dựa trên thông tin từ các nguồn dữ liệu văn bản như tài liệu, sách, bài báo hoặc nguồn thông tin trực tuyến. Tác vụ này đóng vai trò quan trọng trong nhiều ứng dụng thực tế, chẳng hạn như trợ lý ảo (virtual assistants), hệ thống tìm kiếm thông tin, truy vấn tri thức và cải thiện trải nghiệm người dùng trong các ứng dụng thương mại điện tử và dịch vụ trực tuyến. Tuy nhiên, để ứng dụng bài toán này vào các ứng dụng tương tác trực tiếp với người dùng, yêu cầu câu trả lời của hệ thống Hỏi đáp phải chính xác và gần gũi với người dùng.

Hiện tại, đối với tiếng Việt, hầu hết các mô hình hỏi đáp vẫn sử dụng phương pháp trích xuất câu trả lời, dẫn đến việc câu trả lời của mô hình thường mang tính cứng nhắc và ảnh hưởng đến trải nghiệm người dùng. Sử dụng các mô hình ngôn ngữ lớn có sẵn như ChatGPT thường đòi hỏi nhiều chi phí. Do đó, việc xây dựng một mô hình có khả năng trả lời câu hỏi của người dùng một cách chính xác và trôi chảy trở nên vô cùng quan trọng.

1.2 Các giải pháp hiện tại và hạn chế

1.2.1 Sử dụng mô hình ngôn ngữ lớn

Sau khi Chatgpt ra đời và chứng minh được khả năng của nó trên nhiều tác vụ khác nhau, việc sử dụng các mô hình ngôn ngữ lớn để thực hiện tác vụ hỏi đáp trở lên phổ biến dần. Các câu trả lời của các mô hình ngôn ngữ lớn cũng rất giống như đang hỏi đáp người với người.

Để mô hình ngôn ngữ lớn thật sự hiệu quả cần phải có nhiều điều kiện:

- Mô hình phải có sự hiểu biết đủ lớn để trả lời. Để làm được điều này, mô hình cần phải được huấn luyện với lượng dữ liệu khổng lồ trên các lĩnh vực khác nhau.
- Mô hình cần phải đủ thông minh. Với các câu hỏi khó cần sự suy luận, mô hình phải dựa trên nhiều thông tin khác nhau mới đưa ra được câu trả lời phù hợp.

Tuy vậy, phương án này vẫn còn nhiều điểm hạn chế:

- Do lượng thông tin của mô hình bị giới hạn lại tại thời điểm huấn luyện nên khi hỏi đáp về những sự việc mới đang xảy ra thì mô hình khó có thể trả lời được hoặc có thể đưa ra câu trả lời không chính xác. Việc huấn luyện mô hình để cập nhật kiến thức mới cũng có nhiều rủi ro do chi phí huấn luyện mô hình là rất lớn và có thể ảnh hưởng chất lượng của mô hình trên các tác vụ khác nhau.
- Do việc mô hình trả lời rất thật, tương tự như con người nên khó có thể phân biệt được giữa câu trả lời sai và câu trả lời đúng. Vì vậy việc ứng dụng phương pháp này vào các hệ thống đòi hỏi độ chính xác cao vẫn phải cân nhắc.

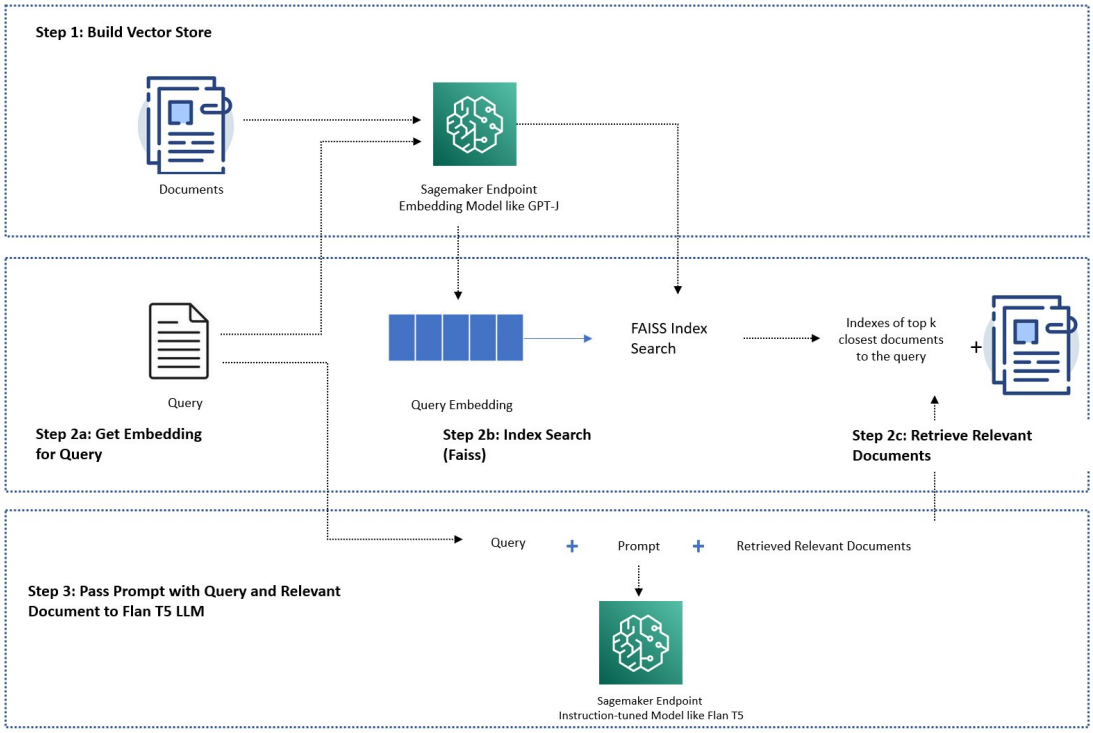
1.2.2 Sử dụng kết hợp hệ thống tìm kiếm

Khi kết hợp với hệ thống tìm kiếm, các câu trả lời được mô hình đưa ra trở lên đáng tin cậy hơn vì câu trả lời sẽ được dựa trên một nguồn thông tin nào đó thay vì nó tự suy luận câu trả lời. Nhưng để thực hiện phương pháp này vẫn cần hệ thống hỏi đáp đáng tin cậy để tìm ra câu trả lời đúng nhất cho câu hỏi của người dùng.

Có 2 phương án tiếp cận cho vấn đề trên:

- Sử dụng bộ trích xuất câu trả lời: Bộ trích xuất câu trả lời này sẽ nhận vào câu hỏi và đoạn văn có thể chứa câu trả lời và tìm ra vị trí của câu trả lời phù hợp.
- Sử dụng mô hình ngôn ngữ để sinh ra câu trả lời: Mô hình này nhận đầu vào tương tự với mô hình trích xuất nhưng thay vì tìm vị trí câu trả lời thì mô hình sẽ phải sinh ra câu trả lời dưới dạng chuỗi.

Đối với tiếng Việt, số lượng các mô hình phục vụ cho mục đích hỏi đáp khá hạn chế, loại trừ các mô hình ngôn ngữ lớn như ChatGPT và Bard thì hầu như các mô hình còn lại phục vụ cho mục đính hỏi đáp thường thiên về trích xuất câu trả lời như mô hình nguyenvulebinh/vi-mrc-base hay mô hình mT0 đa ngôn ngữ cũng trả lời ra những câu trả lời theo dạng trích xuất. Mô hình sinh có thể đưa ra câu trả lời chính xác và trôi chảy vẫn cần được nghiên cứu thêm.



Hình 1.1: Pipeline mô tả phương pháp kết hợp bộ tìm kiếm và mô hình sinh cho hỏi đáp

1.3 Mục tiêu và định hướng giải pháp

Đồ án này của em tập trung vào phát triển mô hình sinh câu trả lời dựa trên câu hỏi và thông tin hỗ trợ. Mục tiêu chính là tạo ra các câu trả lời chính xác và tự nhiên, gần giống như cách mà con người trả lời. Khi tích hợp vào các hệ thống tương tác với người dùng, điều này có thể cải thiện trải nghiệm người dùng một cách đáng kể.

Định hướng giải pháp cho mục tiêu trên sẽ là tập trung vào việc huấn luyện các mô hình sinh sử dụng các tập dữ liệu dạng chỉ dẫn. Lí do cho việc này là trong thời gian gần đây, các mô hình huấn luyện sử dụng chỉ dẫn như ChatGPT hay những mô hình có kích cỡ nhỏ hơn đáng kể như Alpaca, Vicuna với tiếng Anh đang cho thấy khả năng đặc biệt trong những tác vụ khác nhau và trong tác vụ hỏi đáp, các mô hình kể trên cũng cho những câu trả lời rất chính xác và trôi chảy như người viết.

Đồng thời, tiếng Việt đang thiếu những mô hình đã được huấn luyện đa tác vụ như mT0, vậy nên đồ án sẽ kết hợp luấn luyện một mô hình đa tác vụ để kiểm tra

hiệu quả của phương pháp này.

1.4 Phạm vi của nguyên cứu

Mục tiêu của đề tài là xây dựng mô hình hỏi đáp trên tiếng Việt sử dụng các mô hình ngôn ngữ lớn. Mô hình sẽ phải hiểu yêu cầu của người dùng và đưa ra phản hồi phù hợp với yêu cầu đó. Đối với tiếng Việt, ngoài Chatgpt và gpt-4, các mô hình ngôn ngữ lớn khác vẫn không đủ khả năng để đưa ra các câu phản hồi có chất lượng cao.

Để xây dựng mô hình trên tiếng Việt, em sẽ tập trung vào phương pháp instruction fine-tuning tương tự như quá trình xây dựng các mô hình trên tiếng Anh hoặc đa ngôn ngữ như Flan t5, mT0, Alpaca,... Điều này đòi hỏi việc phải có 1 lượng dữ liệu dạng instructions chất lượng cao với tiếng Việt để tinh chỉnh mô hình. Đối với tiếng Anh, các bộ dữ liệu như Flan, P3, Alpaca, còn trong tiếng Việt, lượng data chất lượng cao tương tự các bộ dữ liệu trên rất hạn chế, đòi hỏi phải thu thập và dịch các bộ dữ liệu từ tiếng Anh sang tiếng Việt. Trong phạm vi đồ án, em xin phép không đề cập chi tiết quá trình dịch các bộ dữ liệu từ tiếng Anh sang tiếng Việt.

Do giới hạn tài nguyên nên đồ án của em không được thực nghiệm với nhiều loại mô hình khác nhau. Vì vậy, đồ án tập trung vào huấn luyện các mô hình mT0 large, mT0 xl, mT0 xxl và mô hình bartpho. Đồng thời, các phương pháp fine-tuning hiệu quả như Lora, Gradient checkpointing, 8-bit fine tuning,... cũng được áp dụng để tăng tốc độ training và tối ưu tài nguyên khi training.

Ngoài ra, việc đánh giá mô hình hỏi đáp cũng cần được cân nhắc. Trong thời gian gần đây, các nghiên cứu để đánh giá các mô hình cũng được nhắc đến rất nhiều. Trong phạm vi đồ án này, em sẽ chỉ đánh giá kết quả của mô hình dựa trên lượng nhỏ dữ liệu được tạo từ chatgpt để đưa vào đánh giá độ tương đồng so với nhãn và độ hữu ích của câu trả lời.

1.5 Đóng góp của đồ án

Đồ án này có đóng góp chính như sau:

1. Kiểm tra chất lượng của các mô hình ngôn ngữ khác nhau sau khi huấn luyện với dữ liệu chỉ dẫn với tác vụ hỏi đáp.
2. Huấn luyện mô hình đa tác vụ với tiếng Việt. So sánh hai mô hình trước và sau khi huấn luyện mô hình đa tác vụ bằng việc huấn luyện với bộ dữ liệu chỉ dẫn và kiểm tra độ hiệu quả trên tác vụ hỏi đáp.
3. Kiểm tra độ hiệu quả của các phương pháp tối ưu quá trình huấn luyện của mô hình .

1.6 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2: Trình bày phạm vi nghiên cứu của đồ án và lý thuyết, công trình nghiên cứu tương tự của đồ án, đặc biệt về chủ đề Tinh chỉnh mô hình sử dụng dữ liệu dạng chỉ dẫn, các tập dữ liệu dạng chỉ dẫn dễ dàng tiếp cận và các mô hình ngôn ngữ nổi tiếng được xây dựng theo phương pháp này.

Chương 3: Trình bày về phương pháp đề xuất để giải quyết bài toán hỏi đáp trên tiếng Việt bằng cách huấn luyện mô hình ngôn ngữ với dữ liệu dạng chỉ dẫn. Đồng thời trình bày về một số phương pháp để tối ưu quá trình huấn luyện.

Chương 4: Trình bày chi tiết về quá trình thực nghiệm bao gồm quá trình tinh chỉnh các mô hình và quá trình đánh giá và so sánh kết quả của các mô hình của các mô hình với nhau và so sánh kết quả của các mô hình với ChatGPT.

Chương 5: Trình bày lại vấn đề chính của đồ án, những vấn đề và kết quả đồ án mang lại. Đồng thời trình bày thêm về những hạn chế của đồ án và đưa ra những hướng phát triển trong tương lai.

CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT

2.1 Các kết quả nghiên cứu tương đồng

2.1.1 Mở rộng quy mô các Mô hình Ngôn ngữ Được Tinh chỉnh bằng chỉ dẫn

Bài báo "Scaling Instruction-Finetuned Language Models" được xuất bản bởi Google AI vào tháng 10 năm 2022. Bài báo đề xuất một phương pháp mới để đào tạo các mô hình ngôn ngữ lớn (LLM) có thể được điều chỉnh để thực hiện nhiều tác vụ khác nhau. Phương pháp này được gọi là instruction finetuning (tinh chỉnh mô hình sử dụng chỉ dẫn) và nó dựa trên ý tưởng sử dụng một tập dữ liệu bao gồm những hướng dẫn để mô hình cách thực hiện tác vụ cụ thể.

Bài báo đã chứng minh rằng phương pháp tinh chỉnh mô hình sử dụng chỉ dẫn có thể được sử dụng để đào tạo các LLM có hiệu suất tốt hơn so với các LLM được đào tạo theo cách truyền thống, vốn chỉ dựa trên việc huấn luyện autoregressive dựa trên lượng lớn dữ liệu. Phương pháp này cũng có thể được sử dụng để đào tạo các LLM có thể thực hiện nhiều tác vụ khác nhau, điều mà các LLM được đào tạo theo cách truyền thống không thể làm được.

Ngoài ra, bài báo cũng đề cập đến việc cách tạo ra các tập dữ liệu huấn luyện dưới dạng chỉ dẫn bằng cách tạo ra các chỉ dẫn mẫu cho các tác vụ cụ thể và ghép vào với từng mẫu dữ liệu. Ví dụ về tác vụ hỏi đáp sử dụng tập dữ liệu squad.

- source :"Read this and answer the question {context} {question}"
- target :"{answer}"

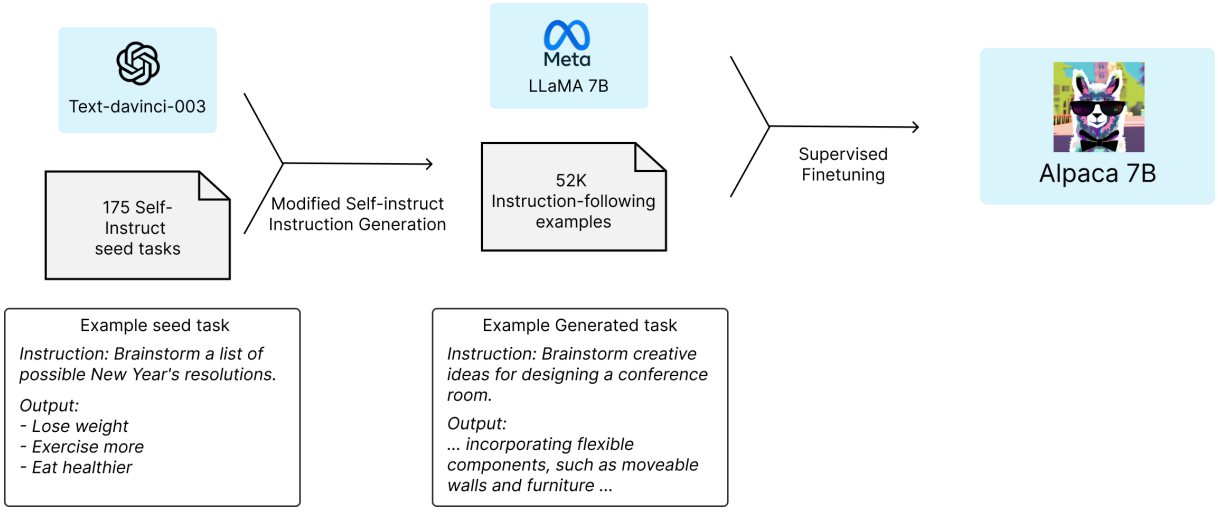
2.1.2 Mô hình Alpaca của StanfordAI

Alpaca: A Strong, Replicable Instruction-Following Model là bài báo nghiên cứu của nhóm tác giả đến từ StanfordAI nói về quá trình xây dựng dữ liệu huấn luyện và mô hình ngôn ngữ lớn sử dụng dữ liệu huấn luyện dạng chỉ dẫn được tạo bởi mô hình ngôn ngữ lớn khác (trong trường hợp này là chatgpt) để tinh chỉnh mô hình. Mô hình Alpaca được tạo theo phương pháp này đã đạt được kết quả khá tốt trên các tập đánh giá khác nhau.

Từ tập dữ liệu tạo dựa trên phương pháp Self-instruct[1] có chỉnh sửa, nhóm tác giả đã huấn luyện mô hình Llama được phát triển bởi MetaAI để thu được các mô hình Alpaca có số lượng tham số là 7, 13, 65 tỷ.

Để đánh giá mô hình sau khi huấn luyện, nhóm tác giả đã tạo bộ dữ liệu so sánh tương tự phương pháp bên trên, sau đó trộn lẫn kết quả của mô hình Alpaca-7B và

của Chatgpt. Kết quả thu được đáng kinh ngạc khi mà kết quả của mô hình Alpaca-7B khá tương đồng với ChatGPT (tỉ số 90-89). Mặc dù tập đánh giá khá nhỏ và không thể đánh giá mọi khía cạnh của mô hình nhưng cách làm này đã thúc đẩy vô số mô hình phát triển theo hướng sử dụng dữ liệu Self-instruct tương tự.



Hình 2.1: Quá trình xây dựng mô hình Stanford Alpaca

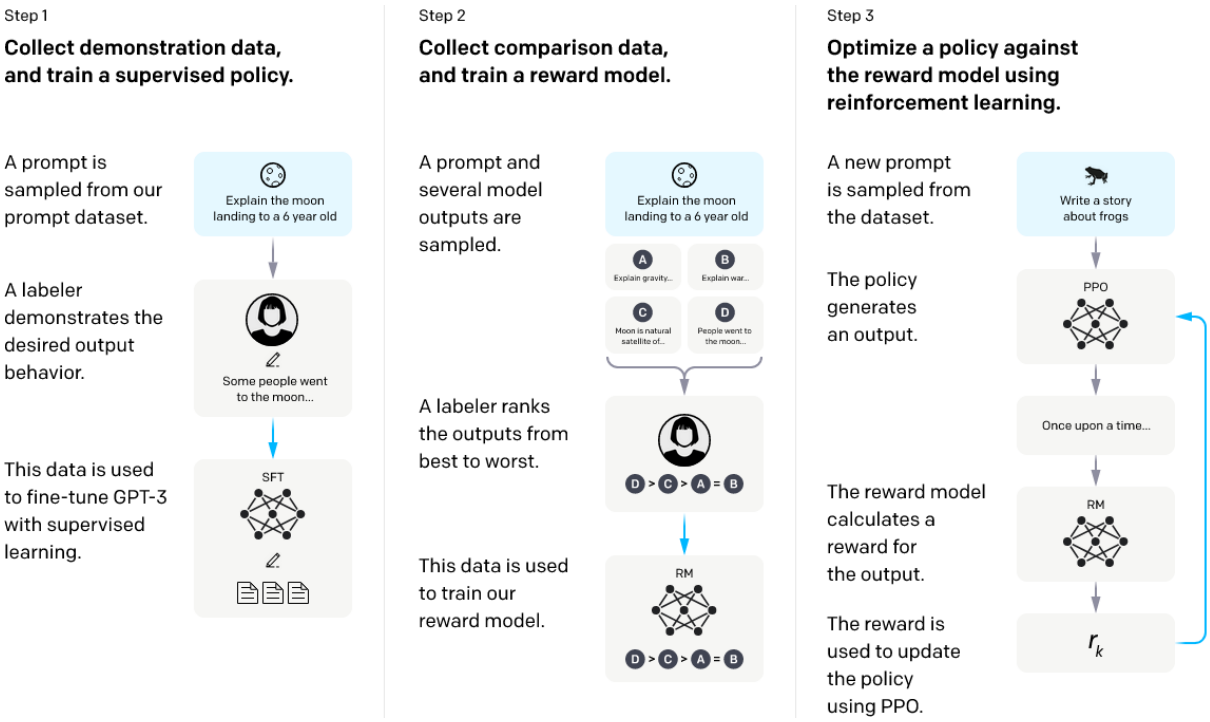
2.1.3 Huấn luyện mô hình để thực hiện chỉ dẫn dựa trên đánh giá người dùng

Bài báo Training language models to follow instructions with human feedback[2] được đề xuất bởi nhóm tác giả OpenAI đề xuất mô hình InstructGPT, đây là mô hình tương đồng nhất với ChatGPT. Mô hình GPT3[3] có 1.3 tỷ tham số được huấn luyện theo phương pháp đề xuất bởi bài báo này cho kết quả tốt hơn mô hình GPT3-175B trên những bài đánh giá bởi người gán nhãn. Bài báo này có đề cập đến vấn đề các mô hình trước đó là GPT2 và GPT3 mặc dù có số lượng tham số khổng lồ (với GPT3 là 175 tỷ tham số) và huấn luyện với lượng dữ liệu rất lớn nhưng các mô hình này vẫn chỉ là học để sinh token tiếp theo dựa vào các token trước đó và thường không trả lời đúng theo chỉ dẫn của người dùng.

Để giải quyết vấn đề trên, nhóm tác giả đã đề xuất giải pháp dựa vào 2 ý tưởng chính:

- Huấn luyện mô hình sử dụng chỉ dẫn.
- Sử dụng học tăng cường để tinh chỉnh mô hình để hạn chế đưa ra những kết quả không mong muốn.

Bộ dữ liệu chỉ dẫn của mô hình instructGPT cần rất nhiều công gắn nhãn. Ban đầu, người gán nhãn sẽ viết 1 số chỉ dẫn cho một số tác vụ khác nhau hoặc người gán nhãn sẽ chỉ dẫn cho mô hình GPT3 để sinh ra một số chỉ dẫn tương tự. Sau đó



Hình 2.2: Quá trình xây dựng mô hình InstructGPT

lọc lại các chỉ dẫn và viết những đầu ra mong muốn tương ứng với mỗi chỉ dẫn, có thể dựa vào đầu ra tạo bởi GPT3 để chỉnh sửa để tăng tốc độ hoặc viết mới. Sau đó tập dữ liệu này sẽ đưa ra để huấn luyện các mô hình. Đây là bước đầu tiên và khá quan trọng vì nếu bước đầu tiên này tệ thì khi qua các bước học tăng cường, nếu tất cả các mô hình đều đưa ra những kết quả tệ thì sẽ không thể gán nhãn tiếp được.

Theo những gì được biết thì ChatGPT và GPT4[4] đều được huấn luyện theo cách này, vì vậy có thể nói việc đưa chỉ dẫn vào để huấn luyện mô hình giúp cải thiện khả năng mô hình khá nhiều và giúp mô hình thực hiện theo yêu cầu người dùng tốt hơn.

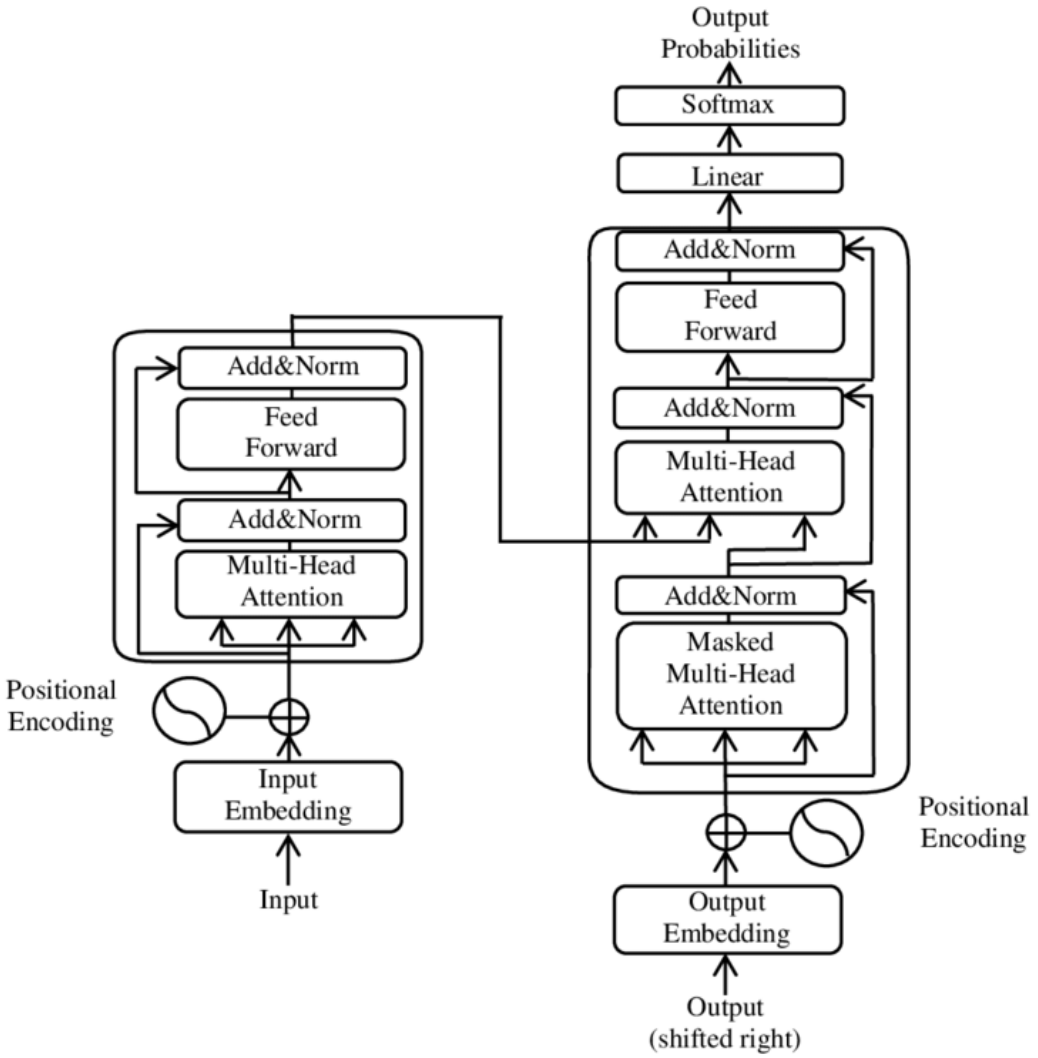
2.2 Transformers

2.2.1 Kiến trúc Transformers

Kiến trúc Transformers là một trong những thành tựu đáng kể trong lĩnh vực học sâu (deep learning) và đã đem lại những đột phá trong xử lý ngôn ngữ tự nhiên và các tác vụ dự đoán chuỗi. Nó được giới thiệu lần đầu tiên qua bài báo nổi tiếng "Attention is All You Need" [5] được xuất bản vào năm 2017 bởi nhóm nghiên cứu tại Google Brain.

Kiến trúc transformers đã tạo ra sự khác biệt lớn trong lĩnh vực học sâu và xử lý ngôn ngữ tự nhiên bằng cơ chế Attention. Cơ chế này giúp các mô hình sử dụng kiến trúc này có thể tập trung vào các phần khác nhau của dữ liệu đầu vào mà không phụ thuộc vào thứ tự. Điều này giúp cho mô hình có thể hiểu rõ ngữ cảnh

của văn bản đầu vào hơn. Điều này đã giúp cho các mô hình transformers đạt được nhiều thành công trong các tác vụ khác nhau như hỏi đáp, dịch máy, phân loại văn bản, ... Một số mô hình nổi bật có thể được kể đến đó là ChatGPT, Roberta[6], Flan-T5,... đều dựa trên kiến trúc transformers.



Hình 2.3: Kiến trúc Transformers

Một điểm đặc biệt của kiến trúc Transformers bao gồm:

- Self-Attention (chú ý riêng): Cơ chế chú ý này cho phép mô hình tập trung vào các phần quan trọng của đầu vào. Điều này giúp cho mô hình có thể xem xét toàn bộ ngữ cảnh của câu một cách hiệu quả và không bị hạn chế bởi cửa sổ trượt như trong các kiến trúc đệ quy.
- Multi-Head Attention (chú ý nhiều đầu vào): Thay vì sử dụng một cơ chế chú ý đơn lẻ, Transformer sử dụng nhiều cơ chế chú ý độc lập nhau, từ đó tăng khả năng học các mối quan hệ phức tạp trong dữ liệu.
- Positional Encoding (mã hóa vị trí): Vì kiến trúc Transformer không duy trì

thông tin về thứ tự từ thông qua các trạng thái ẩn, một mã hóa vị trí đặc biệt được thêm vào để đưa ra thông tin về vị trí của các từ trong chuỗi.

- Kiến trúc transformers gồm 2 khối chính là encoder và decoder, tùy theo loại mô hình thì có thể sử dụng 1 trong 2 khối hoặc có cả 2 khối trên. Thông thường, khối encoder nhận chuỗi đầu vào, xử lý tính toán và đưa ra biểu diễn chuỗi đầu vào, khối decoder nhận biểu diễn từ encoder và tiến hành sinh ra chuỗi đầu ra.

2.2.2 Mô hình ngôn ngữ lớn

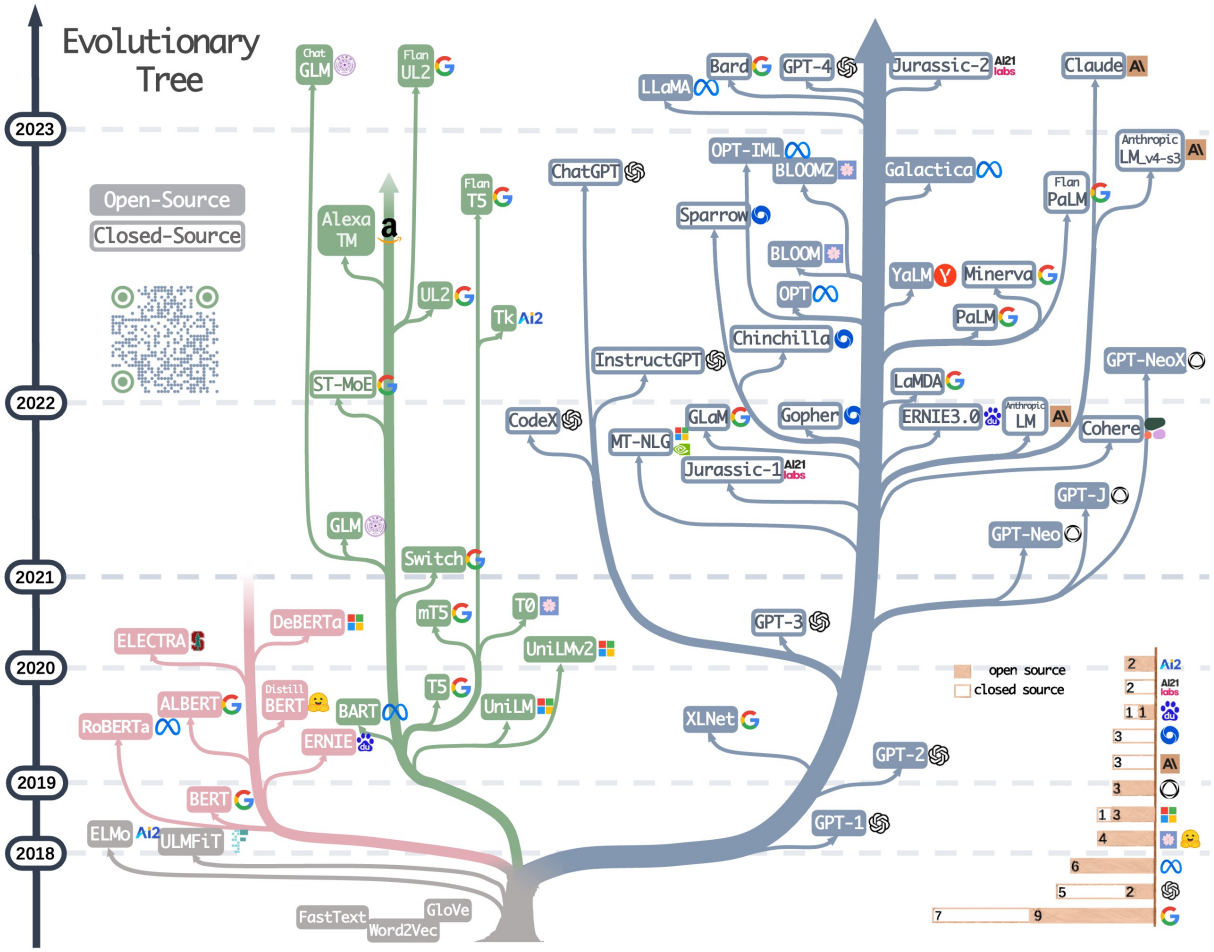
Gần đây, nhất là sau khi ChatGPT được ra đời, các nghiên cứu liên quan đến xử lý ngôn ngữ tự nhiên và đặc biệt là mô hình ngôn ngữ lớn ngày càng xuất hiện nhiều. Mô hình ngôn ngữ lớn là những mô hình có số lượng tham số lớn và cần đến rất nhiều dữ liệu để huấn luyện trên lượng dữ liệu rất lớn, có thể đến hàng petabyte. Các mô hình này có lượng tham số từ hàng tỷ đến hàng trăm tỷ tham số, vì vậy những mô hình này có thể học được biểu diễn phức tạp hơn rất nhiều. Gần đây, do việc nghiên cứu về mô hình ngôn ngữ lớn được cộng đồng quan tâm và đạt được nhiều tiến bộ nên các mô hình ngôn ngữ lớn dần trở lên thông minh hơn và càng làm tốt trong nhiều tác vụ khác nhau.

Một ví dụ cho khả năng mạnh mẽ của mô hình ngôn ngữ lớn là ChatGPT. ChatGPT có thể thực hiện vô số tác vụ từ hỏi đáp đơn giản, tóm tắt và tổng hợp thông tin cho đến thực hiện các chỉ dẫn phức tạp của người dùng như là nhập vai vào nhân vật, viết kịch bản cho vở kịch,... Các mô hình tương tự như ChatGPT như Bard, Palm[7], Claude, LLama[8] cũng dần được đưa vào sử dụng với các mục đích khác nhau.

2.2.3 Mô hình mT0 và Bartpho

Mô hình mT0 và mô hình Bartpho[9] đều được phát triển theo kiến trúc Encoder-Decoder nhưng được xây dựng và huấn luyện theo cách thức khác nhau.

Đối với mT0[10], được phát triển trên kiến trúc mT5[11] được phát triển bởi Google, mô hình này được phát triển bằng cách xây dựng bộ dữ liệu huấn luyện theo dạng text to text với đầu vào là 1 chuỗi và đầu ra là 1 chuỗi và tập dữ liệu huấn luyện cũng trên nhiều ngôn ngữ khác nhau. Điều này giúp mô hình mT0 thích ứng với việc thực hiện theo chỉ dẫn tốt. Dưa trên kiến trúc này, mT0 được nhóm tác giả BigScience xây dựng bằng cách huấn luyện mô hình mt5 với bộ dữ liệu đa ngôn ngữ xP3[10] để thu được các mô hình mT0. Mô hình mT0 có kết quả khá tốt trên các tác vụ khác nhau với ngôn ngữ khác nhau, thậm chí còn vượt với mô hình autoregressive được huấn luyện với dữ liệu tương tự là Bloomz (mT0 13B có kết quả tốt hơn Bloomz 65B).



Hình 2.4: Sự phát triển của các mô hình ngôn ngữ lớn qua các năm

Còn đối với mô hình Bartpho, dựa trên kiến trúc Bart phát triển bởi MetaAI, mô hình được xây dựng bằng phương pháp Denoising[12]. Phương pháp này tập trung vào việc thực hiện những thay đổi với 1 đoạn văn bản đích để đoạn văn bản mất đi 1 phần ý nghĩa, từ đó huấn luyện để mô hình viết lại những văn bản không hoàn chỉnh thành những văn bản hoàn chỉnh. Nhưng do chỉ sử dụng phương pháp này, mô hình Bart thường không thể thực hiện được chỉ dẫn của người dùng. Vì vậy, để sử dụng mô hình bart với các tác vụ cụ thể cần yêu cầu tinh chỉnh thêm.

2.3 Tinh chỉnh mô hình ngôn ngữ sử dụng chỉ dẫn

2.3.1 Một số mô hình được xây dựng bằng phương pháp tỉnh chỉnh sử dụng chỉ dẫn

Phương pháp tinh chỉnh mô hình sử dụng chỉ dẫn được sử dụng ngày càng nhiều trong những năm gần đây, đặc biệt là sau khi OpenAI công bố và cho chạy thử ChatGPT thì càng chứng minh phương pháp rất đáng để áp dụng.

Mô hình tương tự nhất với ChatGPT là InstructGPT cũng được sử dụng phương pháp này để xây dựng. InstructGPT được xây dựng bằng cách training mô hình ban

đầu bằng tập dữ liệu dạng chỉ dẫn được tạo bởi những người gán nhãn (họ viết ra các chỉ dẫn, dựa vào kết quả của mô hình GPT3 để viết lại kết quả). Sau đó họ sử dụng học tăng cường dựa vào đánh giá của người dùng để tinh chỉnh mô hình này. Điều này giúp mô hình dần dần làm chính xác theo đúng chỉ dẫn của người gán nhãn và giảm đi việc kết quả đưa ra không đúng theo chỉ dẫn hoặc không đúng chuẩn mực.

Các mô hình được huấn luyện theo dạng chỉ dẫn được huấn luyện từ bộ dữ liệu ít phức tạp hơn có thể được kể đến là Flan-T5 và mT0. Các mô hình này được huấn luyện bằng các bộ dữ liệu được tạo từ các tập dữ liệu của các tác vụ khác nhau có sẵn và ghép và được viết lại dưới dạng chỉ dẫn để thu được các cặp chĩ dẫn-kết quả.

Trong những tháng gần đây, các mô hình được huấn luyện dựa vào dữ liệu được tạo bằng các mô hình ngôn ngữ lớn như ChatGPT và GPT4 xuất hiện ngày càng nhiều. Một số mô hình có thể được kể tên như là Alpaca, Vicuna[13] được huấn luyện từ các tập dữ liệu theo dạng chỉ dẫn được tạo từ ChatGPT. Độ phức tạp của các chỉ dẫn cũng dần tăng lên như việc tạo ra các tập chỉ dẫn-kết quả có đi kèm cả giải thích trong kết quả giúp các mô hình như Orca[14] tiếp cận gần đến chất lượng của GPT4 hơn.

2.3.2 Một số tập dữ liệu dạng chỉ dẫn được dùng để tinh chỉnh mô hình

a, P3 và Flan

P3(Public Pool of Prompts) được đề cập trong paper *Multitask Prompted Training enables zero-shot task generation* [15] là tập dữ liệu bao gồm tập chỉ dẫn của rất nhiều tác vụ khác nhau trong lĩnh vực xử lý ngôn ngữ tự nhiên như là hỏi đáp, tóm tắt văn bản, ... Vào tháng 10 năm 2022, nhóm tác giả đã thu thập được hơn 2000 loại chỉ dẫn cho hơn 275 tác vụ khác nhau. Tập dữ liệu này hiện được đăng công khai trên nền tảng Huggingface và có thể dễ dàng tải xuống và sử dụng.

Tương tự bộ P3, các tác giả của google cũng tạo ra bộ dữ liệu Flan và gần đây là Flanv2 tương tự theo hướng của nhóm tác giả BigScience. Bộ Flanv2[16] hiện tại có kích thước lớn hơn 200GB bao gồm các loại chỉ dẫn của các tác vụ khác nhau. Đặc biệt trong bộ FlanV2 có dữ liệu dạng CoT (Chain of Thoughts), dữ liệu dạng này thường có dạng chỉ dẫn cho tác vụ cụ thể kết hợp ví dụ đi kèm. Đồng thời, cách tạo bộ dữ liệu và toàn bộ code được nhóm tác giả đăng trong git Flan để mọi người có thể dễ dàng tái tạo lại.

b, Alpaca

Tập dữ liệu Alpaca được tạo theo phương pháp Self-instruct như đã đề cập bên trên. Các bước chính để xây dựng tập dữ liệu có thể được miêu tả bằng các bước

sau đây:

- Thu thập 175 cặp chỉ dẫn-kết quả từ tập dữ liệu khởi đầu của bộ dữ liệu Self-instruct. Các cặp dữ liệu này được viết bởi con người.
- Sử dụng ChatGPT để tạo ra những câu chỉ dẫn tương tự. Chi tiết prompt có thể được tìm thấy ở https://github.com/tatsu-lab/stanford_alpaca . Mỗi lần, nhóm tác giả có thể tạo được 20 chỉ dẫn khác nhau từ ChatGPT. Bộ dữ liệu thu thập được có tổng 52 nghìn câu chỉ dẫn khác nhau.
- Với mỗi chỉ dẫn được tạo từ bên trên, tiến hành sinh ra câu kết quả tương ứng với chỉ dẫn sử dụng ChatGPT.

Quá trình trên giúp nhóm tác giả ở Stanford có thể tạo được tập dữ liệu với tổng chi phí dưới 500$.

Sau này, dựa trên tập dữ liệu Alpaca, nhóm tác giả Tloen đã xây dựng tập dữ liệu Alpaca-GPT4 tương tự nhưng thay vì dùng ChatGPT để sinh ra những câu kết quả thì dùng GPT4 để sinh ra những câu kết quả. Bằng cách này, chất lượng của bộ dữ liệu cũng được đẩy lên với chất lượng tuyệt vời của GPT4.

c, Dolphin

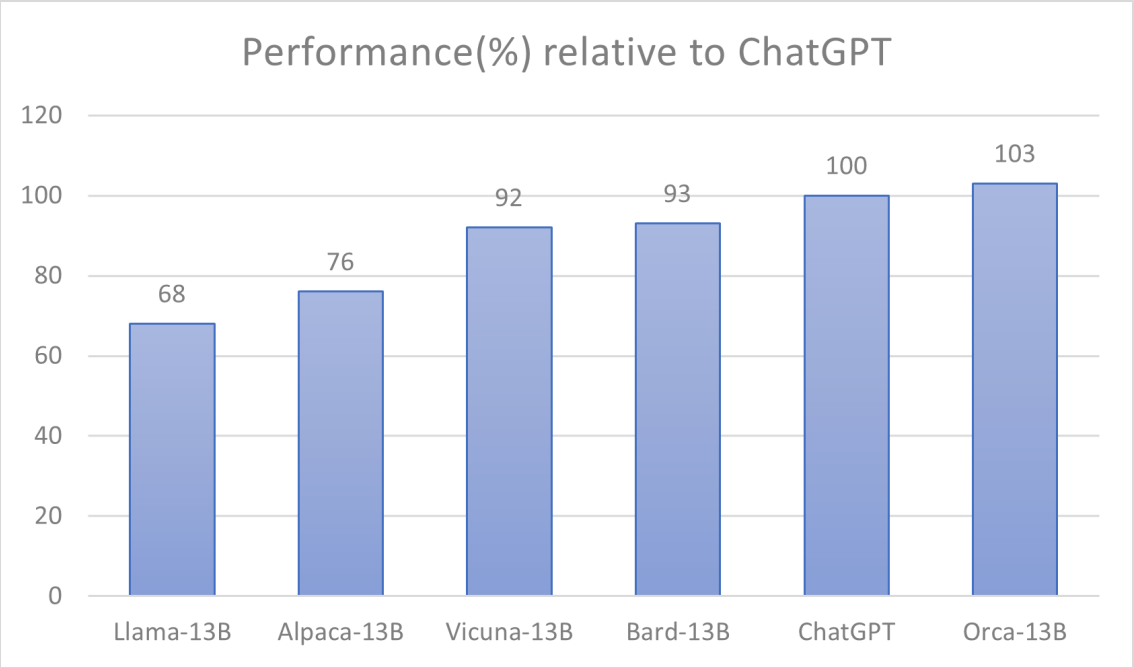
Tập dữ liệu dolphin là kết hợp của tập dữ liệu dạng như Flan và cách tạo dữ liệu dựa trên paper Orca: Progressive Learning from Complex Explanation Traces of GPT-4 [14] của nhóm tác giả đến từ Microsoft. Để tăng khả năng của các mô hình huấn luyện từ dữ liệu tạo bởi ChatGPT và GPT4, nhóm tác giả đã tạo bộ dữ liệu Orca bằng cách sử dụng chỉ dẫn của các tác vụ khác nhau từ các bộ dữ liệu như Flanv2 và Natural Instructions[17] rồi sử dụng GPT4 để tạo những câu trả lời chất lượng cao và giải thích lý do tại sao có câu trả lời như vậy. Điều này giúp mô hình Orca-13B đạt chất lượng tốt hơn ChatGPT với tập đánh giá Vicuna được đánh giá bởi GPT4 được minh họa bởi biểu đồ 2.5.

2.4 Tối ưu quá trình huấn luyện mô hình ngôn ngữ

Việc huấn luyện mô hình ngôn ngữ luôn đòi hỏi tài nguyên tính toán lớn và thời gian huấn luyện cũng rất dài, đặc biệt là những mô hình ngôn ngữ lớn. Do vậy, khi huấn luyện những mô hình này, việc tối ưu thời gian huấn luyện cũng như tài nguyên sử dụng cũng cần được chú ý đến.

2.4.1 Tinh chỉnh tham số hiệu quả

Các mô hình học sâu trong lĩnh vực Xử lý ngôn ngữ tự nhiên thường dựa trên kiến trúc Transformers, các mô hình này thường có số lượng tham số lớn từ hàng trăm triệu đến hàng trăm tỷ tham số. Do đó, mỗi khi tỉnh chỉnh các mô hình này với tác vụ cụ thể sử dụng tinh chỉnh đầy đủ (full fine-tuning) tốn rất tài nguyên



Hình 2.5: Kết quả trên tập đánh giá Vicuna được đánh giá bởi GPT4

và chi phí tính toán. Để giảm tải các chi phí trên, các phương pháp tập trung vào tinh chỉnh tham số hiệu quả được đề xuất và đưa vào sử dụng. Ví dụ điển hình là phương pháp Lora[18] được cộng đồng rất quan tâm trong thời gian gần đây.

Tinh chỉnh tham số hiệu quả tập trung vào việc tinh chỉnh một lượng nhỏ tham số trong mô hình và đóng băng (freeze) các tham số còn lại. Bằng cách này, các chi phí tính toán và tài nguyên sử dụng để cập nhật lại tham số trong mỗi bước huấn luyện được giải tải rất nhiều do chỉ phải huấn luyện và cập nhật lượng nhỏ tham số. Ví dụ khi sử dụng phương pháp Lora với mô hình mT0-XXL với 13 tỷ tham số, lượng tham số phải huấn luyện chỉ chiếm khoảng 0.22%, do đó giải tải lượng lớn tài nguyên tính toán và vram sử dụng.

a, Tinh chỉnh tham số hiệu quả với PEFT

PEFT[19] là thư viện được phát triển bởi tác giả Sourab Mangrulkar giúp cho việc áp dụng các phương pháp tinh chỉnh tham số hiệu quả vào huấn luyện mô hình học sâu, đặc biệt là các mô hình có số lượng tham số lớn như Llama, GPT, mT0, ...

PEFT tập trung vào việc sử dụng các phương pháp tinh chỉnh tham số hiệu quả để tinh chỉnh mô hình cho tác vụ cụ thể để tối ưu tài nguyên trong quá trình huấn luyện nhưng vẫn đạt được chất lượng tương xứng so với tinh chỉnh tham số đầy đủ mặc dù lượng tham số được huấn luyện nhỏ hơn rất nhiều.

Ngoài ra, bằng việc chỉ huấn luyện một mạng neuron rất nhỏ được thêm vào mô hình ngôn ngữ, việc lưu trữ các mô hình đã được huấn luyện cho các tác vụ khác nhau cũng được tối ưu hơn rất nhiều, ví dụ huấn luyện một mô hình ngôn ngữ lớn

cho 3 tác vụ khác nhau, sau khi huấn luyện chỉ cần lưu trữ mô hình ban đầu và 3 mạng neuron để thêm vào mô hình khi sử dụng thay vì phải lưu 3 phiên bản của mô hình ngôn ngữ lớn khi tinh chỉnh đầy đủ, điều này có thể giúp giảm cả trăm GB lưu trữ.

Một số phương pháp tinh chỉnh tham số hiệu quả được xây dựng trong PEFT là:

- LORA: LORA: LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS[18]
- Prefix Tuning: P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks[20]
- Prompt Tuning: The Power of Scale for Parameter-Efficient Prompt Tuning[21]
- P-Tuning: GPT Understands, Too[22]

Thư viện PEFT cũng được xây dựng để dễ dàng đưa vào huấn luyện các mô hình dựa trên kiến trúc Transformers và thư viện Transformers.

b, LORA

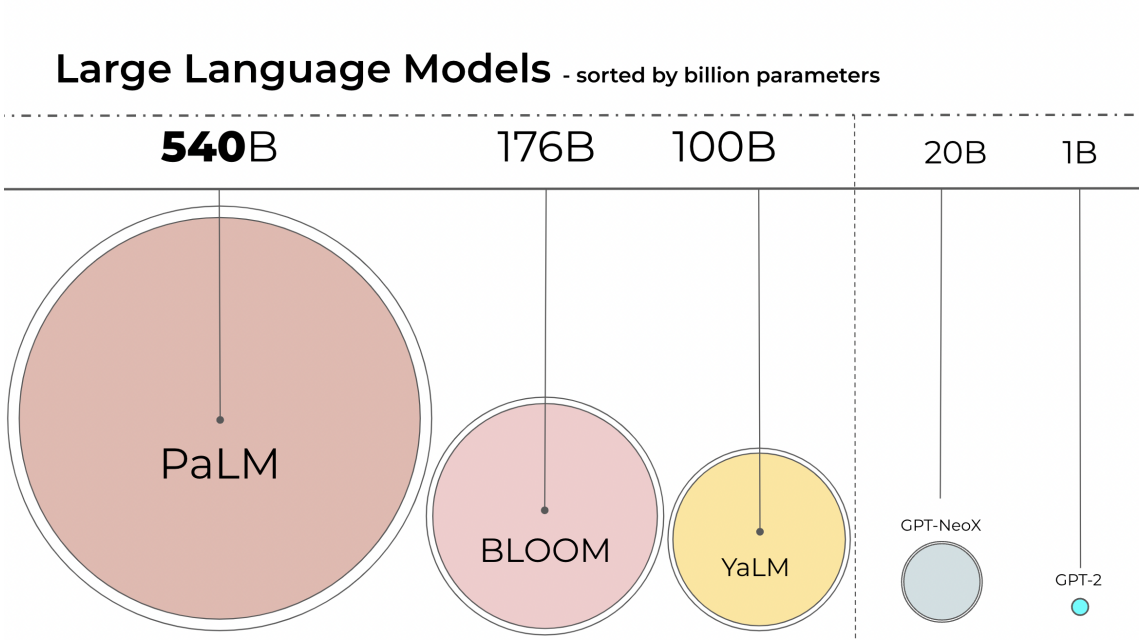
LORA là phương pháp tinh chỉnh tham số hiệu quả giúp tăng tốc quá trình huấn luyện các mô hình lớn nhưng sử dụng ít bộ nhớ VRAM hơn. Phương pháp này thêm các ma trận trọng số phân rã (rank-decomposition weight matrices) hay còn gọi là các ma trận cập nhật (update matrices). Sau đó tinh chỉnh mô hình bằng việc tinh chỉnh những ma trận cập nhật này thay vì toàn bộ mô hình. Điều này mang lại một số điểm mạnh sau đây:

- Do các tham số của mô hình được huấn luyện mô hình trước đó được đóng băng, chỉ huấn luyện các ma trận cập nhật được thêm vào nên tránh được hiện tượng quên những kiến thức đã được huấn luyện trước đó (catastrophic forgetting)
- Các ma trận cập nhật được thêm vào có số lượng tham số rất nhỏ so với mô hình ban đầu, do đó chi phí huấn luyện, lưu trữ cũng được giảm đi rất nhiều.
- Thường các ma trận cập nhật được đưa vào lớp attention của các mô hình, có thể dễ dàng điều chỉnh mức độ thích nghi với dữ liệu huấn luyện mới qua tham số scale.
- Vì việc tối ưu lượng tham số huấn luyện nên tạo điều kiện huấn luyện các mô hình lớn trên các phần cứng dễ dàng tiếp cận (T4, V100, P100, RTX2080,...)

2.4.2 Tinh chỉnh mô hình sử dụng phương pháp nhân ma trận 8bit hiệu quả

Phương pháp nhân ma trận 8bit hiệu quả được giới thiệu ở bài báo LLM.int8()[23] để giải quyết vấn đề suy giảm chất lượng các mô hình ngôn ngữ lớn sau khi được lượng tử hóa[24] (quantization). Phương pháp này phân quá trình nhân ma trận bằng việc chia quá trình tính toán thành 2 phần khác nhau, các phần trạng thái ẩn ngoại lai (có giá trị lớn và khi lượng tử hóa sẽ bị đưa về đầu khoảng) sẽ không lượng tử hóa và tính toán ở 16bit hoặc 32bit, và phần đã được lượng tử hóa sẽ được tính toán ở 8bit.

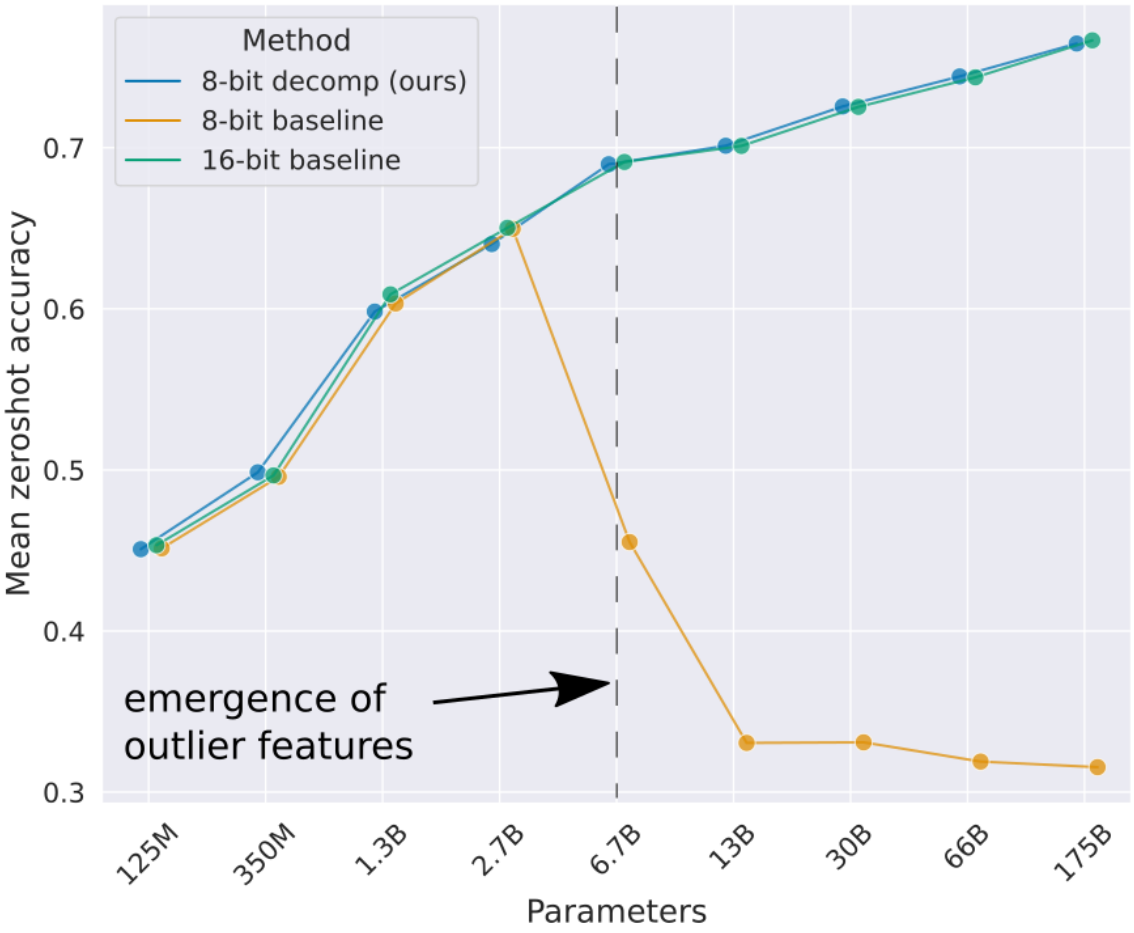
Phương pháp nhân ma trận 8bit hiệu quả này đã được áp dụng không chỉ trong quá trình suy luận mà còn được áp dụng cả trong các quá trình huấn luyện. Do đó, quá trình huấn luyện các mô hình sẽ tốn ít vram hơn. Đặc biệt khi các mô hình ngôn ngữ hiện nay có số lượng tham số rất lớn, như ở hình 2.6, việc huấn luyện các mô hình ở 32bit hoặc 16bit sẽ yêu cầu số lượng vram rất lớn.



Hình 2.6: Kích cỡ các mô hình ngôn ngữ hớn

Ngoài ra, điểm đặc biệt của phương pháp này ở chỗ chỉ lượng tử hóa những điểm dữ liệu thông thường, không lượng tử hóa những điểm dữ liệu ngoại lai (lớn hơn 1 ngưỡng nào đó). Trong quá trình thực nghiệm, tác giả có đưa ra việc khi lượng tử hóa những điểm dữ liệu ngoại lai này sẽ gây tụt giảm nghiêm trọng về chất lượng mô hình được tác giả mô tả ở hình 2.7. Để mô hình sau khi lượng tử hóa (một phần) đạt được chất lượng như mô hình ban đầu thì cách thức thực hiện quá trình nhân ma trận như đề xuất của tác giả để tăng độ chính xác nhất có thể là rất quan trọng.

Phương pháp này sẽ bao gồm các bước chính:



Hình 2.7: So sánh chất lượng các phương pháp lượng tử hóa

1. Từ các trạng thái ẩn đầu vào, trích xuất các trạng thái ngoại lai (ví dụ: các giá trị lớn hơn một ngưỡng nhất định) theo cột.
2. Thực hiện phép nhân ma trận với các trạng thái ngoại lai trong định dạng FP16 (floating-point 16-bit) và các điểm không phải là ngoại lai trong định dạng int8 (số nguyên 8-bit).
3. Chuyển đổi ngược lại các kết quả không phải là ngoại lai về dạng FP16 và cộng cả kết quả của cả ngoại lai và không phải là ngoại lai lại với nhau để nhận được kết quả hoàn chỉnh trong định dạng FP16.

2.4.3 Gradient Checkpointing

Thông thường khi huấn luyện, những thành phần chính tiêu tốn bộ nhớ là:

- Bộ nhớ Mô hình (Model Memory): là bộ nhớ để lưu trữ các tham số của mô hình trong quá trình huấn luyện. Đây là loại bộ nhớ chính xuất hiện cả trong quá trình suy luận và quá trình huấn luyện mô hình.
- Bộ nhớ Tối ưu hóa (Optimization Memory) là loại bộ nhớ được sử dụng để lưu trữ các gradient và bất kỳ bộ đệm động nào trong quá trình huấn luyện.

Ví dụ, nếu sử dụng SGD tiêu chuẩn với momentum, có một giá trị động lượng tương ứng với mỗi trọng số trong mô hình. Trong quá trình backpropagation, chúng ta tính toán (ước lượng ngẫu nhiên của) gradient đối với tất cả các tham số huấn luyện của mô hình. Sau đó, các bộ đệm động được cập nhật dựa trên gradient đã tính toán này.

- Bộ nhớ Kích hoạt (Activation Memory): Trong quá trình huấn luyện, đầu ra của các tầng trong mạng được lưu trữ lại để sử dụng trong quá trình lan truyền ngược (do các dữ liệu này cần để tính toán gradient), đây còn được gọi là bộ nhớ kích hoạt tiến (forward activation memory). Còn trong quá trình lan truyền ngược, bộ nhớ kích hoạt lùi (backward activation memory) sử dụng để lưu trữ gradient của các tầng được tính toán tương ứng với đầu ra của các tầng này. Bộ nhớ Kích hoạt là tổng 2 loại bộ nhớ này.

Để giảm tải bộ nhớ trong quá trình huấn luyện, phương pháp Gradient Checkpointing[25] được đề xuất để tập trung vào việc giảm tải bộ nhớ kích hoạt. Thay vì lưu trữ toàn bộ bộ nhớ kích hoạt của các tầng trong mạng, phương pháp này chỉ lưu lại bộ nhớ kích hoạt của một số tầng nhất định. Trong quá trình lan truyền ngược, các đàu ra của tầng sẽ được tính toán lại khi cần tính toán gradient. Việc tính toán như thế này không ảnh hưởng đến kết quả đầu ra của mô hình, do đó không tồn tại trade off giữa lượng bộ nhớ và độ chính xác. Mặc dù vậy, quá trình lan truyền ngược sẽ đòi hỏi tính toán lại một số tầng nhất định, do đó sẽ làm giảm tốc độ huấn luyện.

CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT

3.1 Tổng quan giải pháp

Để xây dựng mô hình hoạt động hiệu quả cho tác vụ hỏi đáp, em sẽ thực hiện việc thu thập dữ liệu dạng chỉ dẫn cho các tác vụ liên quan đến hỏi đáp từ tập dữ liệu P3 và tập dữ liệu Alpaca-GPT4 (hiện đang ở tiếng Anh). Sau đó em sẽ dịch và lọc bỏ những trường hợp lỗi do quá trình dịch. Sau khi có lượng data trên tiếng Việt này, em sẽ thực hiện training với các mô hình khác nhau để thu được mô hình hỏi đáp.

3.2 Thu thập và xử lý dữ liệu huấn luyện

3.2.1 Thu thập và xử lý dữ liệu từ tập P3

Việc thu thập dữ liệu của tập P3 có thể được thực hiện đơn giản qua việc sử dụng thư viện Datasets [26] để lấy dữ liệu thẳng từ Huggingface. Tuy nhiên, cần phải chọn ra các tác vụ phù hợp để thu thập thay vì lấy hết tất cả dữ liệu từ bộ P3. Lý do của việc này là việc tài nguyên tính toán hạn chế để thu thập và xử lý hết tập dữ liệu P3, đồng thời việc dịch dữ liệu cũng mất nhiều thời gian nên cần phải chọn những bộ dữ liệu phù hợp để sử dụng.

Các tập dữ liệu chính từ bộ dữ liệu P3 sử dụng:

- **Adversarial qa[27]** (adversarial qa dbert answer the following q)
- **Squadv2[28]** (squad v2 Questions with Context)
- **Quoref[29]** (quoref Answer Question Given Context)
- **Wiki qa[30]** (wiki qa Jeopardy style)
- **Web questions[31]** (web questions potential correct answer)
- **Trivia qa[32]** (trivia qa unfiltered question with instruction)

Từ bộ dữ liệu này, em tiến hành dịch sang tiếng Việt. Để lọc ra những trường hợp lỗi dịch, em tiến hành tính tỉ lệ số từ của văn bản trước khi dịch và văn bản sau khi dịch của từng mẫu dữ liệu. Sau đó dựa trên phân phối của tỉ lệ này để lọc ra những mẫu dữ liệu có tỉ lệ bất thường (outliers). Từ đó thu thập được bộ dữ liệu có khoảng gần 400 nghìn mẫu dữ liệu cho các tác vụ hỏi đáp khác nhau và các tác vụ liên quan.

3.2.2 Thu thập và xử lý bộ dữ liệu Alpaca-GPT4

Tương tự bộ dữ liệu Alpaca của nhóm tác giả StanfordAI, bộ dữ liệu Alpaca-GPT4 cũng được xây dựng dựa trên phương pháp self-instruct nhưng thay vì việc

sinh các câu trả lời bằng Chatgpt, các câu trả lời của bộ dữ liệu này được tạo từ GPT4. Do đó, chất lượng câu trả lời của bộ dữ liệu cũng được tăng lên.

Với mỗi mẫu dữ liệu thì sẽ bao gồm 3 phần:

- **instruction** : Câu chỉ dẫn để giúp mô hình hiểu được cần phải làm gì.
- **input** : Đầu vào của tác vụ.
- **output** : Đầu ra của mô hình.

Từ những trường thông tin trên, để đưa vào huấn luyện mô hình thì cần phải ghép vào template của Alpaca:

"Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request. Instruction: {instruction} Input: {input} Response:".

Để sử dụng bộ dữ liệu này trên tiếng Việt, em tiến hành xử lý bộ dữ liệu qua các bước:

- Bước 1: Thu thập dữ liệu: Dữ liệu có thể dễ dàng tải xuống từ github:

  https://github.com/tloen/alpaca-lora.
- Bước 2: Dịch bộ dữ liệu sang tiếng Việt
- Bước 3: Tạo định dạng prompt tiếng Việt tương tự như định dạng tiếng Anh. Ghép dữ liệu vào đúng định dạng và chuẩn bị để đưa vào huấn luyện.

Lý do chính cần sử dụng bộ dữ liệu này là việc nếu chỉ huấn luyện mô hình với các tập dữ liệu từ các bộ như P3, Flan,... vốn được viết lại từ các bộ hỏi đáp dạng trích xuất câu trả lời thì kết quả đầu ra của mô hình cũng thường giống như là đang trích xuất lại câu trả lời và không gần gũi người dùng. Việc dùng dữ liệu chỉ dẫn từ Alpaca-gpt4 giúp mô hình thích ứng với cách phản hồi với người dùng một cách phù hợp hơn, trả lời giống con người hơn.

3.3 Huấn luyện mô hình Bartpho trên tiếng Việt

Do mô hình Bartpho trên tiếng Việt mới thực hiện quá trình huấn luyện theo phương pháp Denoising nên có thể mô hình bartpho khó có thể thích ứng được với các chỉ dẫn phức tạp của bộ dữ liệu Alpaca GPT4.

Do đó, em sẽ chia quá trình huấn huyện mô hình Bartpho làm 2 phần.

- Phần 1 tập trung vào huấn luyện mô hình Bartpho đa tác vụ sử dụng dữ liệu P3 đã dịch sang tiếng Việt để mô hình có thể đưa ra những câu trả lời tương tự như các mô hình mT0. Quá trình này tương tự quá trình huấn luyện các mô hình như các mô hình T0 và Flan-T5. Sau quá trình huấn luyện này, mô hình

sẽ có thể trích xuất câu trả lời như các mô hình đa tác vụ tương tự.

- Phần 2 là huấn luyện với dữ liệu chỉ dẫn từ bộ alpaca-gpt4 để mô hình có thể đưa ra câu trả lời giống với khi con người trả lời hơn thay vì chỉ trích xuất câu trả lời như các mô hình hỏi đáp dạng trích xuất hay các mô hình đa tác vụ như mT0.

Do mô hình Bartpho (large) có số lượng tham số không quá lớn (400 triệu tham số), do đó việc huấn luyện mô hình Barptho được huấn luyện bằng các phương pháp thông thường thay vì các phương pháp được sử dụng để tối ưu lượng bộ nhớ sử dụng.

3.3.1 Huấn luyện mô hình Bartpho với dữ liệu đa tác vụ

Để huấn luyện mô hình bartpho đa tác vụ, em sử dụng bộ dữ liệu đa tác vụ đã được dịch và xử lý từ bộ dữ liệu P3 được đề cập bên trên. Mô hình được huấn luyện bằng cách sử dụng Trainer được cung cấp bởi framework Transformers một cách không quá phức tạp.

Để chắc chắn mô hình sau khi huấn luyện hoạt động đúng mong muốn, em có giữ lại một số mẫu dữ liệu hỏi đáp từ bộ dữ liệu trên để tự kiểm tra lại, nếu mô hình trả lời ra kết quả đúng như nhãn thì được coi là phù hợp để chuyển sang bước huấn luyện tiếp theo.

3.3.2 Huấn luyện mô hình Bartpho với dữ liệu chỉ dẫn từ bộ Alpaca-GPT4-vi

Sau khi thu được mô hình bartpho đa tác vụ, sử dụng mô hình này để huấn luyện tiếp với bộ dữ liệu Alpaca-GPT4-vi đã được xử lý trước đó. Mô hình sau khi được huấn luyện kì vọng sẽ học được cách trả lời câu hỏi một cách chính xác và trôi chảy.

3.4 Huấn luyện mô hình mT0 với chỉ dẫn tiếng Việt

Do các mô hình mT0 đã được huấn luyện với tập dữ liệu xP3 bao gồm chỉ dẫn các tác vụ khác nhau với các ngôn ngữ khác nhau trong đó có tiếng Việt, việc huấn luyện lại mô hình mT0 với dữ liệu chỉ dẫn tiếng Việt có thể không tạo ra ảnh hưởng lớn, do đó để tối ưu quá trình huấn luyện thì em chỉ sử dụng bộ dữ liệu alpaca-gpt4-vi với các mô hình mT0. Ngoài ra, do kích thước của các mô hình mT0 thường lớn hơn rất nhiều so với mô hình Bartpho (mT0 base có 580 triệu tham số, mT0 xxl có 13 tỷ tham số, bartpho có 400 triệu tham số) nên việc tối ưu tài nguyên thời gian để huấn luyện cần được chú ý hơn.

3.4.1 Đối với mô hình có kích cỡ nhỏ

Đối với mô hình mT0-base và mt0-large có số lượng tham số lần lượt là 580 triệu và 1.2 tỷ tham số thì hoàn toàn có thể sử dụng phương pháp tinh chỉnh thông

thường kết hợp một số tham số đặc biệt để huấn luyện các mô hình này.

Khi huấn luyện mô hình mT0 large, do kích cỡ đầu vào và đầu ra mô hình lớn nên sẽ tốn nhiều vram hơn (hết nhiều hơn 40GB với batch size bằng 4) thì để tối ưu thì em sẽ sử dụng Gradient Checkpointing. Tham số này đã được tích hợp vào sẵn trong Trainer của framework Transformers nên có thể dễ dàng sử dụng. Sau khi dùng phương pháp này thì có thể huấn luyện dễ dàng và lượng vram sử dụng còn khoảng 20gb.

3.4.2 Đối với mô hình có kích cỡ lớn

Với mô hình mT0-xl có 3.7B tham số và mT0-xxl có 13B tham số thì việc finetune thông thường sẽ tốn rất nhiều tài nguyên. Ví dụ như mT0-xxl cần đến 56gb vram nên không thể sử dụng 1 GPU A100 40GB để tinh chỉnh. Do đó cần áp dụng nhiều phương pháp giúp tối ưu quá trình huấn luyện.

Chi tiết với từng mô hình:

- mT0-xl: Mô hình mT0-xl có 3.7 tỷ tham số và có kích cơ khoảng 16GB. Để huấn luyện mô hình này cần sử dụng một số phương pháp như:
  - Tính toán ở 16bit hay vì 32bit. Các mô hình mT0 cũng được huấn luyện ở 16bit để tối ưu tài nguyên hơn. Bằng cách này thì có thể tăng tốc quá trình tính toán với ma trận đồng thời giảm tiêu tốn vram 2 lần.
  - Sử dụng phương pháp nhân ma trận 8bit để huấn luyện. Phương pháp này lưu trữ những tham số trong khoảng nhất định ở dạng 8bit và những tham số còn lại ở 16 hoặc 32bit. Với mỗi bước tính toán, phương pháp này sẽ chuyển những tham số ở 8bit lên 16 hoặc 32 bit để tính toán. Cách này sẽ làm giảm lượng vram sử dụng nhưng đồng thời cũng tăng thời gian huấn luyện do số bước tính toán tăng lên.
  - Sử dụng Lora. Phương pháp này để hỗ trợ việc phải huấn luyện lại cả mô hình thì chỉ cần huấn luyện một mạng neuron nhỏ kết hợp với mô hình. Điều này giúp số lượng tham số cần phải cập nhật lại giảm rất nhiều (với một số mô hình lớn thì chỉ cần huấn luyện 0.25% tổng số lượng tham số của mô hình) từ đó tăng tốc độ tính toán.
  - Sử dụng Gradient Checkpointing. Việc sử dụng Gradient checkpointing giúp thay vì lưu trữ trạng thái của cả mô hình thì chỉ cần lưu trữ trạng thái của một số tầng nhất định trên vram. Điều trên giúp giảm lượng vram sử dụng nhưng do phải tính toán lại trạng thái của 1 số tầng khi cần nên có thể làm giảm tốc độ huấn luyện.
- Đối với mô hình mT0-xxl có 13B tham số và nặng khoảng 56gb, phương pháp

áp dụng để training có sự thay đổi so với mô hình mT0-xl. Phương pháp áp dụng để huấn luyện mô hình là qlora[33]. Điểm khác biệt chính của phương pháp này với bên trên là ở việc mô hình ban đầu sẽ được quantize 2 lần từ float 32 xuống 8bit và từ 8bit xuống 4bit. Quá trình huấn luyện xảy ra tương tự nhưng do mô hình chưa hỗ trợ hết nên khi huấn luyện cần phải lưu lại mạng lora sau đó khi huấn luyện xong mới ghép lại vào mô hình chính để sử dụng.

CHƯƠNG 4. ĐÁNH GIÁ THỰC NGHIỆM

4.1 Tập dữ liệu đánh giá

Để có được tập dữ liệu để đánh giá các mô hình đã được huấn luyện, em tiến hành thu thập dữ liệu từ nguồn báo chí và tạo 1 lượng nhỏ dữ liệu bằng cách sử dụng APi của ChatGPT.

Chi tiết quá trình tạo tập dữ liệu:

- Bước 1: Thu thập dữ liệu từ nguồn báo chí, tách và lọc các đoạn văn bản có chứa khoảng 100 đến 200 từ để tối ưu chi phí từ việc gọi APi chatgpt và các đoạn văn bản cũng không quá ngắn.
- Bước 2: Từ tập dữ liệu thu thập được, em tiến hành sinh ra câu hỏi và câu trả lời tương ứng bằng cách gọi APi từ ChatGPT. Tổng cộng thu thập được khoảng 160 câu hỏi và câu trả lời tương ứng.
- Bước 3: Lấy ngẫu nhiên một số mẫu dữ liệu để kiểm tra. Sau khi kiểm tra thấy không có lỗi thì đưa vào sử dụng.

4.2 Các tham số đánh giá

4.2.1 Đánh giá dựa trên độ trùng lặp

a, Đánh giá trên độ trùng lặp về từ

Với mỗi câu trả lời được sinh ra bởi mô hình, tiến hành so sánh với câu trả lời được đưa ra bởi ChatGPT. Tính các thông số precision, recall, f1 dựa trên số từ trùng lặp của 2 câu trả lời trên. Từ đó có thể đánh giá được độ tương đồng của câu trả lời của mô hình với câu trả lời được sinh bởi ChatGPT.

Các thông số có thể tính dựa trên các thông số sau:

- TruePositives là số từ xuất hiện cả trong câu trả lời của mô hình và câu trả lời của chatgpt.
- FalsePositive là số từ xuất hiện trong câu trả lời của mô hình nhưng không xuất hiện trong câu trả lời của chatgpt.
- FalseNegative là số từ xuất hiện trong câu trả lời của chatgpt nhưng không xuất hiện trong câu trả lời của mô hình.

Từ đó, ta có thể tính các thông số precision, recall, f1 dựa trên các công thức dưới đây:

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{4.1}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{4.2}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.3}$$

b, Đánh giá độ trùng lặp dựa trên Bert-Score

Tương tự như phương pháp trên, sử dụng BertScore để đánh giá độ tương đồng của câu trả lời được trả ra bởi mô hình với câu trả lời được tạo bởi ChatGPT. Điểm BertScore càng cao thì câu trả lời được sinh ra càng tương đồng với câu trả lời được tạo bởi ChatGPT.

BertScore được đề xuất bởi bài báo Evaluating Text generation with Bert[34] để đánh giá độ tương đồng của các văn bản với nhau. Phương pháp đánh giá này cho thấy kết quả tốt hơn nhiều so với các phương pháp đánh giá như bleu và rouge trên các tác vụ như tóm tắt văn bản hay hỏi đáp. Nhưng thay vì so khớp từ như ở phía trên, BertScore so sánh độ tương đồng về mặt ngữ nghĩa của mỗi từ trong văn bản dự đoán với mỗi từ trong văn bản tham chiếu để từ đó đưa ra độ tương đồng của 2 văn bản.

BertScore tính toán độ tương đồng qua các bước:

1. Tính embeddings cho từng từ của văn bản dự đoán và văn bản tham chiếu từ đó được 2 tập vector embeddings tương ứng là $x = \langle x_1, \ldots, x_k \rangle$ và $\hat{x} = \langle \hat{x}_1, \ldots, \hat{x}_m \rangle$.

2. Tính độ tương đồng cosine của từng vector tương ứng với từng token của văn bản dự đoán và văn bản tham chiếu. Ví dụ với x_i và $\hat{x}_j$ được tính theo công thức:
$$\text{cosine\_similarity}(x_i, \hat{x}_j) = \frac{x_i \cdot \hat{x}_j}{\|x_i\| \cdot \|\hat{x}_j\|} \tag{4.4}$$

3. Từ độ tương đồng của các token được tính như trên, có thể tính được các thông số precision, recall, f1 theo các công thức sau:

$$\text{RBERT} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \text{cosine\_similarity}(x_i, \hat{x}_j), \tag{4.5}$$

$$\text{PBERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \text{cosine\_similarity}(x_i, \hat{x}_j) \tag{4.6}$$

$$\text{FBERT} = \frac{2 \cdot \text{PBERT} \cdot \text{RBERT}}{\text{PBERT} + \text{RBERT}} \tag{4.7}$$

Prompt	Câu hỏi: **Câu hỏi** Thông tin: **Thông tin để đưa ra câu trả lời** Dưới đây là các câu trả lời khác nhau cho câu hỏi dựa vào thông tin bên trên, hãy chấm điểm cho các câu trả lời theo độ phù hợp của câu trả lời theo thang điểm 10. Câu trả lời 1_ : **Câu trả lời tạo ra bởi mô hình 1** Câu trả lời 2_ : **Câu trả lời tạo ra bởi mô hình 2** Câu trả lời 3_ : **Câu trả lời tạo ra bởi mô hình 3** Câu trả lời 4_ : **Câu trả lời tạo ra bởi mô hình 4** Câu trả lời 5_ : **Câu trả lời tạo ra bởi mô hình 5** Câu trả lời 6_ : **Câu trả lời tạo ra bởi mô hình 6** Trả về kết quả dưới dạng json: [["id_", "score"]] và không đưa ra giải thích gì thêm.
Sample Response	[["1_", 10], ["2_", 9], ["3_", 5], ["4_", 7], ["5_", 7], ["6_", 7]]

Bảng 4.1: Bảng mô tả mẫu prompt để thu thập dữ liệu đánh giá từ Chatgpt

Các thông số trên có thể cho thấy độ tương đồng của văn bản dự đoán và văn bản tham chiếu, trong trường hợp này là câu trả lời của mô hình và câu trả lời của chatgpt.

4.2.2 Điểm đánh giá câu trả lời của các mô hình sinh ra bởi Chatgpt

Dựa trên paper Instruction Tuning with GPT-4[35] đánh giá các mô hình sử dụng phương pháp Self-Instruct[1] để huấn luyện, nhóm tác giả sử dụng mô hình GPT4 để đánh giá xem chất lượng phản hồi với mỗi chỉ dẫn của các mô hình nào là tốt hơn. Từ phương pháp này, em tiến hành đánh giá các mô hình đã huấn luyện bằng cách đưa vào ChatGPT để thu thập đánh giá và xếp hạng các mô hình.

Thông thường, tác giả sẽ đánh giá chất lượng qua việc so sánh kết quả câu trả lời tương ứng câu chỉ dẫn xem mô hình nào cho kết quả tốt hơn. Tương tự như phương pháp trên, nhưng thay vì so sánh 2 mô hình với nhau thì em đưa tất cả kết quả của các mô hình ghép vào 1 prompt và yêu cầu chatgpt trả ra điểm đánh giá trên thang điểm 10, từ đó thu được đánh giá của nhiều mô hình cùng lúc.

Để đánh giá các câu trả lời của mô hình, em tiến hành ghép kết quả của các mô hình kèm với câu hỏi và thông tin chứa câu trả lời tương ứng vào mẫu prompt được miêu tả ở 4.1. Sau khi thu được điểm từ chatgpt thì tiến hành tổng hợp lại điểm và tính điểm trung bình câu trả lời cho từng mô hình, trong đó có cả câu được sinh bởi chatgpt.

4.3 Phương pháp thí nghiệm

Huấn luyện các mô hình theo phương pháp được đề xuất ở chương 3, từ đó thu thập được 5 mô hình bao gồm mT0-xxl, mT0-xl, mT0-large, mT0-base và Bartpho-

Model	Model size	Full Finetuning	Lora 8bit	Qlora	Gradient Checkpointing	Training time	Training VRAM
Bartpho	400M	TRUE	FALSE	FALSE	FALSE	4H	26gb
mT0-base	580M	TRUE	FALSE	FALSE	FALSE	5H	30gb
mT0-large	1.3B	TRUE	FALSE	FALSE	TRUE	13H	32gb
mT0-xl	3.7B	FALSE	TRUE	FALSE	TRUE	24H	28gb
mT0-xxl	13B	FALSE	FALSE	TRUE	TRUE	36H	38gb

Bảng 4.2: Bảng tóm tắt quá trình huấn luyện các mô hình

instruction. Thiết lập huấn luyện các mô hình với bộ dữ liệu Alpaca-GPT4 được mô tả ở bảng 4.2, ngoài ra độ dài đầu vào của mô hình (input context length) và batch size thay đổi có thể ảnh hưởng đến thời gian huấn luyện và lượng vram sử dụng.

Sau khi thu thập được các mô hình sau khi huấn luyện, tiến hành sử dụng các mô hình đã huấn luyện đưa vào đánh giá sử dụng bộ dữ liệu được tạo từ ChatGPT được nhắc đến bên trên. Tiến hành sử dụng các mô hình để sinh ra các câu trả lời cho các câu hỏi và thông tin đã tạo và tổng hợp lại để thu thập tập các câu trả lời tương ứng.

Để thu thập câu trả lời, ghép các câu hỏi và thông tin để đưa ra câu trả lời vào đúng định dạng chỉ dẫn để mô hình hiểu cần phải trả lời câu hỏi và sinh ra câu trả lời. Định dạng mẫu cho 1 cặp câu hỏi và thông tin để hỗ trợ đưa ra câu trả lời:

- Đầu vào: "Trả lời câu hỏi dựa trên thông tin dưới đây.

  Câu hỏi: Connor Slocombe đã giành giải quán quân của cuộc thi gì?

  Thông tin: Connor Slocombe, 12 tuổi, hôm qua giành giải quán quân trong cuộc thi Đôi giày Hôi nhất Toàn quốc lần thứ 42, sau ba lần thất bại ở những năm trước đó, theo Reuters. 4 giám khảo của cuộc thi đều ấn tượng trước mùi giày của Slocombe. George Aldrich, một chuyên gia hoá học tại NASA, cho biết đôi giày của Slocombe mang nhiều loại mùi, trong đó có " mùi thối rữa xộc lên mũi khiến bạn chảy nước mắt và sau đó buồn nôn ". " Cô của cháu có một trang trại và đôi khi cháu đến giúp đỡ cô. Khi nhìn thấy phân động vật, cháu sẽ bước lên để đôi giày thêm bẩn ", Slocombe tự hào nói. Cậu bé nhận được giải thưởng là 2.500 USD và vé xem biểu diễn Broadway."
- Đầu ra mong muốn: "Câu trả lời : Connor Slocombe, 12 tuổi, đã giành giải quán quân trong cuộc thi Đôi giày Hôi nhất Toàn quốc lần thứ 42, sau ba lần thất bại."

Sau khi thu thập được câu trả lời từ các mô hình khác nhau, tiến hành thực hiện 2 thực nghiệm chính để đánh giá kết quả thu được bởi các mô hình:

- So sánh độ tương đồng của câu trả lời sinh ra bởi các mô hình và câu trả lời

metrics	matching_f1	matching_presicion	matching_recall
mt0_xxl_qlora	0.511304045	0.487243135	0.625484839
mt0_xl_lora	0.479702546	0.454564829	0.612609528
bartpho_instruction	0.290314111	0.227743181	0.542607372
mt0_base	0.404065777	0.424008338	0.498314025
mt0_large	0.367549248	0.33211575	0.574763106

Bảng 4.3: Bảng đánh giá độ trùng lặp từ của câu trả lời của các mô hình với câu trả lời tạo bởi ChatGPT

đưa ra bởi chatgpt.
Với mỗi câu trả lời được sinh ra bởi các mô hình, tiến hành so sánh độ trùng lặp với câu trả lời được sinh ra bởi chatgpt bởi 2 tham số đánh giá là so khớp từ và bertscore.

- Sử dụng chatgpt để đánh giá độ phù hợp cho câu trả lời của các mô hình.
  Tổng hợp lại tất cả các câu trả lời được sinh ra bởi các mô hình và câu hỏi kèm thông tin hướng dẫn tương ứng. Lấy ngẫu nhiên 50 mẫu dữ liệu và tạo thành các prompt như ở 4.1. Sau khi thu được kết quả thì tổng hợp lại.

Ngoài ra, tiến hành huấn luyện mô hình bartpho trực tiếp với dữ liệu chỉ dẫn Alpaca-gpt4-vi để kiểm tra sự khác biệt của việc huấn luyện đa tác vụ trước khi huấn luyện với chỉ dẫn phức tạp và không huấn luyện đa tác vụ bằng việc so sánh kết quả tương đồng với câu trả lời tạo ra bởi ChatGPT như phía trên.

4.4 Đánh giá độ tương đồng của mô hình với câu trả lời tạo bởi chatgpt

Với mỗi câu trả lời của mô hình, tiến hành đánh giá độ tương đồng của câu trả lời với câu trả lời được sinh bởi ChatGPT. Câu trả lời có độ tương đồng càng cao thì càng tương đồng câu trả lời được tạo bởi ChatGPT.

4.4.1 Kết quả đánh giá độ trùng lặp về từ

Từ mỗi câu trả lời, tiến hành tính điểm f1, presicion, recall của câu trả lời với câu trả lời được tạo bởi chatgpt. Kết quả được thống kê ở bảng 4.3. Có thể nhận thấy ở trường hợp này, mô hình có càng nhiều tham số thì sau khi được huấn luyện sẽ có câu trả lời càng tương tự về mặt từ với ChatGPT.

4.4.2 Kết quả đánh giá độ trùng lặp bằng BertScore

Đánh giá độ trùng lặp của các câu trả lời bằng việc sử dụng BertScore. Do đánh giá bằng mô hình ngôn ngữ nên để khách quan hơn thì em sẽ tạo embedding của các câu trả lời bằng 2 mô hình ngôn ngữ khác nhau: Phobert và XLM-Roberta. Kết quả được biểu diễn ở bảng 4.4 và bảng 4.5.

Với những mô hình như mT0, số lượng tham số tăng cũng dẫn đến kết quả tăng.

metrics	bert_score_f1	bert_score_presicion	bert_score_recall
mt0_xxl_qlora	0.33897597	0.3305111	0.3492195
mt0_xl_lora	0.33754516	0.3276784	0.34968913
bartpho_instruction	0.34566087	0.32685286	0.36984953
mt0_base	0.33100483	0.32647496	0.33707148
mt0_large	0.32967025	0.31287536	0.35009968

Bảng 4.4: Bảng đánh giá độ trùng lặp BertScore-Phobert của câu trả lời tạo bởi các mô hình và tạo bởi ChatGPT

metrics	bert_score_f1	bert_score_presicion	bert_score_recall
mt0_xxl_qlora	0.9814348	0.98080987	0.98206216
mt0_xl_lora	0.98111475	0.9804632	0.9817692
bartpho_instruction	0.9808985	0.9794868	0.9823159
mt0_base	0.98029804	0.98014385	0.980456
mt0_large	0.9798423	0.97894686	0.98074114

Bảng 4.5: Bảng đánh giá độ trùng lặp BertScore-XLM-Roberta của câu trả lời tạo bởi các mô hình và tạo bởi ChatGPT

Riêng với mô hình Bartpho-instruction thì cho thấy độ tương đồng khá cao, cao nhất khi sử dụng đánh giá bằng embedding phobert và cao hơn mT0 base và large khi đánh giá bằng XLM-Roberta.

4.5 Xếp hạng câu trả lời dựa vào ChatGPT

Sau khi thu thập được dữ liệu đánh giá từ Chatgpt thì sẽ tiến hành tổng hợp lại và thu được bảng 4.6.

Đối với các mô hình mT0, có thể dễ dàng thấy được là mô hình có kích cỡ càng lớn thì điểm càng cao. Đặc biệt, mô hình mT0-xxl đạt được kết quả tốt hơn kết quả được sinh bởi ChatGPT, có thể lí do đạt được kết quả này là do dữ liệu huấn luyện của mô hình này được tạo từ GPT4, vốn tốt hơn ChatGPT nên mô hình cũng học được cách sinh ra kết quả tốt tương tự GPT4.

Đối với mô hình bartpho-instruction, điểm đánh giá của Chatgpt với mô hình này cũng tương đối tốt, bằng mT0 base và khá gần với mT0-large mặc dù mô hình bartpho có số lượng tham số nhỏ hơn nhiều (bartpho có 400 triệu tham số, mT0-base có 580 triệu tham số, mT0-large có 1.2 tỷ tham số).

Nhưng không phải tất cả mọi trường hợp mô hình có số lượng tham số lớn hơn đều cho kết quả tốt, kết quả của mô hình kèm đánh giá từ Chatgpt trong trường hợp này được đánh giá ở hình 4.1. Cũng có thể dễ dàng thấy được mặc dù dùng Chatgpt để đánh giá nhưng kết quả câu trả lời tạo bởi Chatgpt cũng không được thiên vị và đánh giá các kết quả bằng mắt thường cũng thấy được sự hợp lý của đánh giá này.

question	Tại sao tàu cá bị chìm?	
Context	Rạng sáng 30-10, ông Nguyễn Thanh Thuận (trú thị trấn Long Hải, huyện Long Điền, Bà Rịa - Vũng Tàu - chủ tàu BV 4469 TS) báo tin 24 ngư dân trên tàu cá đang về bờ tránh bão Yutu thì bị phá nước phần mũi. Nước tràn vào, tàu không thể điều khiển và nguy cơ chìm. Vị trí tàu thả trôi khoảng 46 hải lý về phía đông nam TP Vũng Tàu. Ngay sau khi nhận tin báo, Trung tâm Phối hợp tìm kiếm cứu nạn hàng hải khu vực III đã điều động tàu SAR 413 từ Vũng Tàu xuất bến ứng cứu. Do ảnh hưởng bão Yutu, lúc này ngoài biển có gió cấp 5-6, giật cấp 7, sóng biển cao trên 2m. Thuỷ thủ tàu SAR 413 đã nỗ lực tìm kiếm và tiếp cận cứu được 24 ngư dân vào trưa cùng ngày. Hiện tàu SAR 413 đang lai dắt tàu cá và đưa 24 ngư dân về bờ.	
Chatgpt	Tàu cá bị phá nước phần mũi.	6
mT0-xxl	Tàu cá bị chìm là do ảnh hưởng của bão Yutu.	5
mT0-xl	Tàu cá bị chìm do sóng biển và gió cấp 5-6, giật cấp 7, sóng biến cao trên 2m.	3
mT0-large	Tàu cá bị phá nước phần mũi do ảnh hưởng bão Yutu. Nước tràn vào, tàu không thể điều khiển và nguy cơ chìm. Do ảnh hưởng của bão Yutu, tàu cá không thể đối phó với nguy cơ chết. Trung tâm Phối hợp tìm kiếm cứu nạn hàng hải khu vực III đã điều động tàu SAR 413 từ Vũng Tàu xuất bến ứng cứu.	8
mT0-base	Tàu cá bị phá nước phần mũi do ảnh hưởng bão Yutu. Nước tràn vào, tàu không thể điều khiển và nguy cơ chìm. Do ảnh hưởng của bão Yutu, tàu cá không thể đối phó với nguy cơ chết. Trung tâm Phối hợp tìm kiếm cứu nạn hàng hải khu vực III đã điều động tàu SAR 413 từ Vũng Tàu xuất bến ứng cứu.	8
Bartpho	Tàu cá bị phá nước phần mũi do ảnh hưởng bão Yutu. Nước tràn vào, tàu không thể điều khiển và nguy cơ chìm. Do ảnh hưởng của bão Yutu, tàu cá không thể đối phó với nguy cơ chết. Trung tâm Phối hợp tìm kiếm cứu nạn hàng hải khu vực III đã điều động tàu SAR 413 từ Vũng Tàu xuất bến ứng cứu.	8

Hình 4.1: Ví dụ đánh giá kết quả bằng Chatgpt

model	ChatGPT score
chatgpt	8.479
mt0_xxl	8.628
mt0_xl	8.000
mt0_large	7.638
bartpho_instruction	7.596
mt0_base	7.596

Bảng 4.6: Bảng thông số đánh giá thu thập từ Chatgpt

4.6 So sánh mô hình Bartpho khi có huấn luyện đa tác vụ và không huấn luyện đa tác vụ

Đánh giá kết quả của mô hình Bartpho thông thường và mô hình đã huấn luyện đa tác vụ bằng cách so sánh độ tương đồng của câu trả lời của 2 mô hình sau khi huấn luyện với dữ liệu dạng chỉ dẫn Alpaca-gpt4-vi với câu trả lời được tạo từ ChatGPT. Kết quả được miêu tả ở bảng 4.7. Kết quả của mô hình huấn luyện từ Bartpho thường kém hơn khá nhiều so với kết quả được tạo bởi mô hình được huấn luyện từ Bartpho đa tác vụ.

		f1	presicion	recall
bartpho multitask	matching	0.2903	0.2277	0.5426
	BertScore phobert	0.3457	0.3269	0.3698
	BertScore xlm-roberta	0.9809	0.9795	0.9823
bartpho	matching	0.1432	0.1215	0.2291
	BertScore phobert	0.3091	0.2788	0.3493
	BertScore xlm-roberta	0.9778	0.9748	0.9808

Bảng 4.7: So sánh Bartpho thường và Bartpho đa tác vụ

4.7 So sánh kết quả của mô hình đa tác vụ và mô hình được huấn luyện bằng chỉ dẫn

Mô hình đa tác vụ như mT0 đã được huấn luyện với dữ liệu chỉ dẫn đa tác vụ, trong đó có tác vụ hỏi đáp. Nhưng các bộ dữ liệu hỏi đáp được sử dụng như Squad, Adversarial, Quoref đều ở dạng trích xuất câu trả lời. Do đó câu trả lời thường sẽ bị ngắn và không tự nhiên.

Với mô hình đã được huấn luyện chỉ dẫn từ bộ Alpaca-GPT4, câu trả lời sinh ra của các mô hình đều có sự cải thiện về mặt ngôn từ và gần gũi người đọc hơn. Ví dụ của việc này có thể được minh họa ở hình 4.2

Question	Hoài Linh xây dựng nhà thờ Tổ nghiệp ở đâu?
Context	Nhà thờ Tổ nghiệp trăm tỷ của Hoài Linh. (Nguồn: Plo.vn) Sau nhiều năm chăm chỉ đi diễn, được biết, danh hài Hoài Linh đã giành dụm một số tiền lớn để của mua một mảnh đất rộng ở quận 9 (TP. HCM), xây một ngôi nhà thờ Tổ nghiệp hoành tráng. Theo tìm hiểu, công trình ngốn cả trăm tỷ nói trên của Hoài Linh xây dựng theo phong cách đền chùa miền Bắc xưa... nằm bên một con rạch nhỏ, trong không gian " quê " tĩnh lặng ở vùng ven thành phố. Nó mang nét cổ kính, lọt thỏm giữa một khuôn viên rộng lớn, và có nhiều hoa lá. Bên ngoài nhà thờ Tổ đang xây dựng của Hoài Linh. Nhiều người dân sống xung quanh công trình đang xây dựng này cho hay, một tuần, nghệ sĩ Hoài Linh tranh thủ tới thăm và đôn đúc thi công 2-3 lần. Trước cổng ngôi nhà thờ là hình ảnh hai con rồng bằng đá hướng đầu ra phía ngoài. Phía bên trong có đặt 3 tảng đá " phụ tử ". Trên tảng đá " phụ " có khắc chữ " Tâm " ở mặt trước, mặt sau khắc chữ " Đạo ". Trần Anh (Tổng hợp)
mT0-large	quận 9 (TP. HCM)
mT0-large-alpaca	Hoài Linh xây dựng nhà thờ Tổ nghiệp trăm tỷ ở quận 9 (TP. HCM) nằm bên một con rạch nhỏ, trong không gian "quê" tĩnh lặng ở vùng ven thành phố. Công trình ngốn cả trăm tỉ nói trên của Hoài Linh mang nét cổ kính, lọt thỏm giữa một khuôn viên rộng lớn và có nhiều hoa lá.

Hình 4.2: So sánh mô hình mT0 thông thường với mT0 đã huấn luyện với dữ liệu Alpaca

CHƯƠNG 5. KẾT LUẬN

5.1 Kết luận

5.1.1 Tóm tắt đồ án

Trong phạm vi đồ án này, em tập trung vào việc sinh ra những câu trả lời phù hợp đối với tác vụ hỏi đáp trên tiếng Việt. Đồ án tập trung vào việc kiểm tra khả năng của các mô hình hoạt động được trên tiếng Việt sử dụng dữ liệu dạng chỉ dẫn được dịch sang tiếng Việt đồng thời đánh giá khả năng sinh ra những câu trả lời chính xác và phù hợp hơn là chỉ đưa ra những câu trả lời được trích xuất. Đồ án cũng đề cập đến một số phương pháp được áp dụng để tối ưu tài nguyên khi huấn luyện mô hình ngôn ngữ, đặc biệt là mô hình ngôn ngữ lớn.

Dựa trên các kết quả đánh giá, ta có thể thấy việc sử dụng các bộ dữ liệu chỉ dẫn được tạo bởi phương pháp như Alpaca giúp cải thiện khả năng trả lời của mô hình, giúp mô hình có thể đưa ra những câu trả lời sát với yêu cầu và mong muốn của người dùng hơn. Đồng thời, việc huấn luyện đa tác vụ cũng giúp cải thiện khả năng thích ứng của mô hình đối với những chỉ dẫn phức tạp hơn, những tác vụ khó hơn.

5.1.2 Hạn chế của đồ án

Đồ án hiện tại tiến hành sử dụng dữ liệu dạng chỉ dẫn để cải thiện khả năng đưa ra những câu trả lời phù hợp của mô hình. Trong quá trình thực hiện đồ án, em nhận thấy có một số hạn chế sau:

- Về dữ liệu:
  - Dữ liệu sử dụng để huấn luyện chưa hoàn hảo: Do dữ liệu sử dụng là được dịch từ tiếng Anh sang tiếng Việt chứ không được tạo từ tiếng Việt ngay từ đầu dẫn đến việc trong quá trình dịch, ý nghĩa của dữ liệu có thể bị ảnh hưởng.

    Ví dụ: "What causes the northern lights?" dịch thành "Điều gì gây ra đèn phía bắc?"
  - Chưa cập nhật những bộ dữ liệu tốt nhất: Trong quá trình thực hiện đồ án, trong tháng 7 thì nghiên cứu về mô hình Orca được công bố và không lâu sau bộ dữ liệu Dolphin của tác giả Eric Hartford được tạo bằng phương pháp này đã được đăng tải lên Huggingface. Do quá trình dịch và xử lý tốn nhiều thời gian nên không thể áp dụng bộ dữ liệu này vào thử nghiệm.
  - Lượng dữ liệu thử nghiệm chưa đa dạng: Do việc tạo dữ liệu từ ChatGPT tốn nhiều chi phí nên em chỉ giới hạn số lượng của bộ dữ liệu ở mốc 160

mẫu dữ liệu (StanfordAI ban đầu đánh giá mô hình Alpaca với khoảng 200 dữ liệu). Ngoài ra do việc đánh giá các mô hình với những câu hỏi mà không đưa thông tin thêm sẽ khó xác nhận việc đâu là câu trả lời đúng và cần phải đánh giá bằng con người. Điều này sẽ tốn rất nhiều thời gian và sức người.

- Về mô hình:
  - Trên tiếng Việt thì các mô hình có chất lượng cao như Flan T5 hay Llama là không có. Do đó đòi hỏi phải sử dụng mô hình như đa ngôn ngữ như mT0 có chất lượng tương đối tốt trên tiếng Việt. Tuy nhiên, do là mô hình đa ngôn ngữ nên khi tokenize,thay vì mỗi từ tương ứng với 1 token thì 1 từ tiếng Việt phải dùng đến 2-3 token. Do đó số lượng token sử dụng nhiều hơn và tốn nhiều Vram hơn.
  - Việc huấn luyện các mô hình dù đã được tối ưu nhưng vẫn tốn rất nhiều thời gian. Đồng thời, khi sử dụng những mô hình trên để suy luận thì sẽ tốn rất nhiều thời gian và tài nguyên. Vì vậy, khi đưa các mô hình này vào hệ thống khác nhau đòi hỏi phải sử dụng các framework hỗ trợ tăng tốc độ suy luận như Ctranslate2 hoặc FasterTransformers.
  - Mô hình Bartpho với 400 triệu tham số đạt kết quả khá là khả quan mặc dù số lượng tham số rất là nhỏ. Điều này cho thấy nếu có một mô hình tiếng Việt đủ lớn thì có thể vượt được chất lượng của các mô hình đa ngôn ngữ với tiếng Việt.

5.2 Hướng phát triển trong tương lai

Một số hướng phát triển trong tương lại có thể được đề cập đến là:

- Đưa ứng dụng của mô hình và ứng dụng dạng Chatbot hoặc Trợ lý ảo (Virtual Assistant): Do mô hình có thể đưa ra được câu hỏi khá giống con người nên mô hình có thể tiếp tục phát triển để đưa vào những ứng dụng như kể trên. Để làm được điều này thì cần tích hợp hệ thống tìm kiếm để tìm được những thông tin phù hợp yêu cầu người dùng và từ đó đưa ra câu trả lời phù hợp. Ngoài ra mô hình cũng phải hoạt động tốt trong những trường hợp khó trong luồng hội thoại.
- Thử nghiệm mô hình với một số framework khác để tăng tốc độ suy luận như Ctranslate2. Kiểm tra việc sau khi chuyển qua framework khác có ảnh hưởng nhiều đến chất lượng câu trả lời không, tốc độ và tài nguyên sử dụng là bao nhiêu.
- Xử lý các bộ dữ liệu chất lượng cao được tạo từ GPT4 mới, dịch sang tiếng

Việt và kiểm tra chất lượng các mô hình đối với loại dữ liệu này.

TÀI LIỆU THAM KHẢO

[1] Y. Wang, Y. Kordi, S. Mishra **andothers**, *Self-instruct: Aligning language models with self-generated instructions*, 2023. arXiv: 2212.10560 [cs.CL].

[2] L. Ouyang, J. Wu, X. Jiang **andothers**, *Training language models to follow instructions with human feedback*, 2022. arXiv: 2203.02155 [cs.CL].

[3] T. B. Brown, B. Mann, N. Ryder **andothers**, *Language models are few-shot learners*, 2020. arXiv: 2005.14165 [cs.CL].

[4] OpenAI, *Gpt-4 technical report*, 2023. arXiv: 2303.08774 [cs.CL].

[5] A. Vaswani, N. Shazeer, N. Parmar **andothers**, "Attention is all you need," *Advances in neural information processing systems*, **jourvol** 30, 2017.

[6] Y. Liu, M. Ott, N. Goyal **andothers**, *Roberta: A robustly optimized bert pretraining approach*, 2019. arXiv: 1907.11692 [cs.CL].

[7] A. Chowdhery, S. Narang, J. Devlin **andothers**, *Palm: Scaling language modeling with pathways*, 2022. arXiv: 2204.02311 [cs.CL].

[8] H. Touvron, T. Lavril, G. Izacard **andothers**, *Llama: Open and efficient foundation language models*, 2023. arXiv: 2302.13971 [cs.CL].

[9] N. L. Tran, D. M. Le **and** D. Q. Nguyen, *Bartpho: Pre-trained sequence-to-sequence models for vietnamese*, 2022. arXiv: 2109.09701 [cs.CL].

[10] N. Muennighoff, T. Wang, L. Sutawika **andothers**, "Crosslingual generalization through multitask finetuning," *arXiv preprint arXiv:2211.01786*, 2022.

[11] L. Xue, N. Constant, A. Roberts **andothers**, *Mt5: A massively multilingual pre-trained text-to-text transformer*, 2021. arXiv: 2010.11934 [cs.CL].

[12] M. Lewis, Y. Liu, N. Goyal **andothers**, *Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*, 2019. arXiv: 1910.13461 [cs.CL].

[13] W.-L. Chiang, Z. Li, Z. Lin **andothers**, *Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality*, 2023. **url**: https://lmsys.org/blog/2023-03-30-vicuna/.

[14] S. Mukherjee, A. Mitra, G. Jawahar, S. Agarwal, H. Palangi **and** A. Awadallah, *Orca: Progressive learning from complex explanation traces of gpt-4*, 2023. arXiv: 2306.02707 [cs.CL].

[15] V. Sanh, A. Webson, C. Raffel **andothers**, *Multitask prompted training enables zero-shot task generalization*, 2021. arXiv: 2110.08207 [cs.LG].

[16] S. Longpre, L. Hou, T. Vu **andothers**, *The flan collection: Designing data and methods for effective instruction tuning*, 2023. arXiv: 2301.13688 [cs.AI].

[17] S. Mishra, D. Khashabi, C. Baral **and** H. Hajishirzi, "Cross-task generalization via natural language crowdsourcing instructions," **in***ACL* 2022.

[18] E. J. Hu, Y. Shen, P. Wallis **andothers**, *Lora: Low-rank adaptation of large language models*, 2021. arXiv: 2106.09685 [cs.CL].

[19] S. Mangrulkar, S. Gugger, L. Debut, Y. Belkada **and** S. Paul, *Peft: State-of-the-art parameter-efficient fine-tuning methods*, https://github.com/huggingface/peft, 2022.

[20] X. L. Li **and** P. Liang, *Prefix-tuning: Optimizing continuous prompts for generation*, 2021. arXiv: 2101.00190 [cs.CL].

[21] B. Lester, R. Al-Rfou **and** N. Constant, *The power of scale for parameter-efficient prompt tuning*, 2021. arXiv: 2104.08691 [cs.CL].

[22] X. Liu, K. Ji, Y. Fu **andothers**, "P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks," **in***Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* Dublin, Ireland: Association for Computational Linguistics, **may** 2022, **pages** 61–68. DOI: 10.18653/v1/2022.acl-short.8. **url**: https://aclanthology.org/2022.acl-short.8.

[23] T. Dettmers, M. Lewis, Y. Belkada **and** L. Zettlemoyer, *Llm.int8(): 8-bit matrix multiplication for transformers at scale*, 2022. arXiv: 2208.07339 [cs.LG].

[24] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. van Baalen **and** T. Blankevoort, *A white paper on neural network quantization*, 2021. arXiv: 2106.08295 [cs.LG].

[25] T. Chen, B. Xu, C. Zhang **and** C. Guestrin, *Training deep nets with sublinear memory cost*, 2016. arXiv: 1604.06174 [cs.LG].

[26] Q. Lhoest, A. Villanova del Moral, Y. Jernite **andothers**, "Datasets: A community library for natural language processing," **in***Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* Online **and** Punta Cana, Dominican Republic: Association for Computational Linguistics, **november** 2021, **pages** 175–184. arXiv: 2109.02846 [cs.CL]. **url**: https://aclanthology.org/2021.emnlp-demo.21.

[27] M. Bartolo, A. Roberts, J. Welbl, S. Riedel **and** P. Stenetorp, "Beat the AI: Investigating adversarial human annotation for reading comprehension," *Transactions of the Association for Computational Linguistics*, **jourvol** 8, **pages** 662–678, 2020. DOI: 10.1162/tacl_a_00338. **url**: https://doi.org/10.1162%2Ftacl_a_00338.

[28] P. Rajpurkar, J. Zhang, K. Lopyrev **and** P. Liang, *Squad: 100,000+ questions for machine comprehension of text*, 2016. arXiv: 1606.05250 [cs.CL].

[29] P. Dasigi, N. F. Liu, A. Marasović, N. A. Smith **and** M. Gardner, *Quoref: A reading comprehension dataset with questions requiring coreferential reasoning*, 2019. arXiv: 1908.05803 [cs.CL].

[30] Y. Yang, W.-t. Yih **and** C. Meek, "WikiQA: A challenge dataset for open-domain question answering," **in***Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* Lisbon, Portugal: Association for Computational Linguistics, **september** 2015, **pages** 2013–2018. DOI: 10.18653/v1/D15-1237. **url**: https://aclanthology.org/D15-1237.

[31] J. Berant, A. Chou, R. Frostig **and** P. Liang, "Semantic parsing on Freebase from question-answer pairs," **in***Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* Seattle, Washington, USA: Association for Computational Linguistics, **october** 2013, **pages** 1533–1544. **url**: https://aclanthology.org/D13-1160.

[32] M. Joshi, E. Choi, D. S. Weld **and** L. Zettlemoyer, *Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension*, 2017. arXiv: 1705.03551 [cs.CL].

[33] T. Dettmers, A. Pagnoni, A. Holtzman **and** L. Zettlemoyer, *Qlora: Efficient finetuning of quantized llms*, 2023. arXiv: 2305.14314 [cs.LG].

[34] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger **and** Y. Artzi, *Bertscore: Evaluating text generation with bert*, 2020. arXiv: 1904.09675 [cs.CL].

[35] B. Peng, C. Li, P. He, M. Galley **and** J. Gao, *Instruction tuning with gpt-4*, 2023. arXiv: 2304.03277 [cs.CL].