Lecture 15: Tour of Modern LLMs

Giới thiệu

Hôm nay, tôi sẽ giới thiệu về một số mô hình ngôn ngữ lớn (LLMs) hiện đại. Hiện nay có rất nhiều mô hình ngôn ngữ lớn, nhưng tôi muốn tập trung vào những mô hình đặc biệt thú vị vì một số lý do. Một số mô hình công khai nhiều thông tin về cách chúng được huấn luyện, giúp chúng ta hiểu rõ hơn về quy trình đào tạo một mô hình ngôn ngữ lớn tiên tiến. Một số khác là những mô hình mạnh mẽ nhất mà bạn có thể tải xuống và sử dụng, như các mô hình ngôn ngữ mã nguồn mở tốt nhất hiện có. Ngoài ra, có những mô hình chuyên biệt cho một chủ đề cụ thể hoặc là những mô hình ngôn ngữ đóng tốt nhất. Tuy nhiên, tôi sẽ tập trung vào hai loại đầu tiên để mọi người có cái nhìn rõ ràng về những gì đang diễn ra trong các mô hình mà bạn sử dụng cho các nhiệm vụ khác nhau.

What Makes a Model?

Trong lĩnh vực phát triển mô hình ngôn ngữ lớn, có ba yếu tố quan trọng cần xem xét: kiến trúc mô hình, dữ liệu sử dụng và quyết định huấn luyện. Một cuộc thảo luận trên Twitter với Tom Wolf, một lãnh đạo tại Hugging Face, đã nhấn mạnh rằng dữ liệu là một trong những phần quan trọng nhất, trong khi kiến trúc mô hình ngày nay ít quan trọng hơn. Điều này có phần đúng, vì hầu hết các mô hình hiện nay sử dụng kiến trúc tương tự nhau, nhưng khả năng và độ chính xác của chúng lại khác biệt đáng kể, cho thấy dữ liệu và quyết định huấn luyện đóng vai trò lớn.

Tuy nhiên, cũng có lý do để cho rằng kiến trúc vẫn quan trọng. Chúng ta đã dành nhiều năm để tối ưu hóa kiến trúc LLaMa, và hiện tại nó hoạt động rất tốt khi huấn luyện các mô hình lớn với nhiều dữ liệu. Nếu sử dụng kiến trúc cũ như LSTM từ năm 2014, những tiến bộ hiện tại sẽ không thể đạt được. Do đó, kiến trúc vẫn có vai trò quan trọng trong việc cải thiện tốc độ và hiệu quả của mô hình.

Open vs. Closed Access

Open/ Closed Access (e.g. Liang et al. 2022)

Trước khi đi vào chi tiết cụ thể, tôi muốn thảo luận về vấn đề truy cập mở và đóng. Đây không phải là vấn đề liên quan trực tiếp đến mô hình hóa, nhưng nó quan trọng và giúp bạn hiểu rõ hơn về môi trường. Có một bài blog hay (Liang et al. 2022), thảo luận về các mức độ mở khác nhau trong việc phát hành các mô hình ngôn ngữ và hệ thống AI tiên tiến.

Chúng ta có thể nói về việc các trọng số có thể mở, được mô tả hay đóng; mã suy luận có thể mở, được mô tả hay đóng; quá trình huấn luyện có thể mở, được mô tả hay đóng; và dữ liệu cũng có thể mở, được mô tả hay đóng. Ví dụ, các mô hình có trọng số mở trên Hugging Face có thể chỉ có trọng số mở, nhưng mã suy luận cũng cần phải mở để có thể thực hiện suy luận. Tuy nhiên, điều này không có nghĩa là mã huấn luyện hay dữ liệu cũng mở.

Có nhiều mức độ mở khác nhau. Ví dụ, các mô hình như GPT-4 hay GPT thường đóng hoàn toàn, và chúng ta biết rất ít về chúng.

Licences and Permissiveness

Trong nghiên cứu và phát triển phần mềm, việc hiểu rõ về các loại giấy phép và mức độ cho phép của chúng là rất quan trọng. Điều này ảnh hưởng đến những gì bạn có thể làm hợp pháp trong các dự án nghiên cứu, đặc biệt là khi làm việc trong các công ty lớn. Các loại giấy phép phổ biến bao gồm:

- Public Domain hoặc CC-0: Cho phép bạn làm bất cứ điều gì với tác phẩm, bao gồm tải xuống, phân phối lại mà không cần ghi công, và sửa đổi. Ví dụ, các tác phẩm của chính phủ Mỹ thường thuộc phạm vi công cộng.

- MIT và BSD: Đây là các giấy phép phần mềm phổ biến với rất ít hạn chế, chủ yếu yêu cầu duy trì thông báo bản quyền. Hệ điều hành MacOS của Apple dựa trên hệ điều hành BSD cũ.

- Apache và CC-BY: Yêu cầu ghi nhận tác giả gốc. Apache còn có điều khoản về bằng sáng chế, cho phép sử dụng mã và các bằng sáng chế liên quan trừ khi bạn kiện công ty phát hành.

- GPL và CC-BY-SA: Yêu cầu chia sẻ lại dưới cùng giấy phép nếu bạn sử dụng mã nguồn. Nhiều công ty tránh sử dụng GPL vì điều này có thể buộc họ phải công khai mã nguồn của toàn bộ hệ thống.

- CC-NC: Không cho phép sử dụng cho mục đích thương mại. Theo Open Source Initiative, các giấy phép có hạn chế sử dụng không được coi là mã nguồn mở.

- LLaMa, OPEN-RAIL: Các công ty như Meta và Hugging Face đã phát triển các giấy phép riêng cho mô hình của họ. Ví dụ, giấy phép Llama của Meta cấm sử dụng mô hình để huấn luyện các mô hình ngôn ngữ không phải từ Llama và không cho phép sử dụng cho mục đích quân sự.

- No License: Nếu bạn không đặt giấy phép cho mã nguồn trên GitHub, điều đó có nghĩa là không ai có thể sử dụng mã của bạn mà không có sự cho phép. Phần lớn văn bản trên internet thuộc loại "all rights reserved", nghĩa là bạn không thể sử dụng chúng mà không có sự cho phép từ chủ sở hữu bản quyền.

Fair Use

Trong bối cảnh sử dụng dữ liệu và tài liệu có bản quyền, khái niệm "fair use" (sử dụng hợp lý) trở nên rất quan trọng, đặc biệt là ở Mỹ. Quy định về fair use ở Mỹ cho phép sử dụng tài liệu có bản quyền trong một số trường hợp nhất định. Ví dụ, việc trích dẫn một lượng nhỏ tài liệu trong sách giáo khoa hoặc bài giảng thường được coi là hợp lý, miễn là không làm giảm giá trị thương mại của tác phẩm gốc.

Một ví dụ điển hình là nếu ai đó trích dẫn toàn bộ "Harry Potter" trong một sách giáo khoa và bán với giá rẻ, điều này sẽ không được coi là fair use vì làm giảm giá trị của tác phẩm gốc. Tương tự, việc tạo một kho sách lớn và cho phép mọi người truy cập miễn phí cũng không phải là fair use vì tác giả không nhận được tiền bản quyền.

Một tiêu chí khác để đánh giá fair use là mục đích sử dụng có mang tính thương mại hay không. Các trường đại học thường được áp dụng tiêu chuẩn linh hoạt hơn khi sử dụng cho mục đích nghiên cứu phi thương mại so với các công ty.

Hiện nay, phần lớn việc đào tạo mô hình AI dựa trên dữ liệu có bản quyền được thực hiện dựa trên giả định về fair use. Điều này có nghĩa là mô hình không thể tái tạo dễ dàng tài liệu gốc và không làm giảm giá trị thương mại của dữ liệu gốc. Tuy nhiên, vẫn có những vụ kiện liên quan đến vấn đề này.

Ví dụ, The New York Times đã kiện OpenAI và Microsoft vì sử dụng bài viết của họ để đào tạo mô hình mà không có sự cho phép. Một ví dụ khác là GitHub Copilot bị kiện bởi những người đã tải phần mềm lên GitHub, cho rằng GitHub không có quyền sử dụng phần mềm của họ để kiếm lợi nhuận.

Fair use rất phổ biến và quan trọng, nhưng cũng đang chịu nhiều thách thức trong bối cảnh các mô hình AI hiện nay.

Why Restrict Model Access?

Có ba lý do chính khiến các công ty thường hạn chế quyền truy cập vào mô hình của họ. Thứ nhất là vì lý do thương mại. Các công ty như OpenAI, Google và Anthropic kiếm tiền từ các API của họ như OpenAI API, Gemini API và Claude API.

Thứ hai là vấn đề an toàn. Có những lo ngại chính đáng rằng nếu phát hành các mô hình mạnh mẽ, chúng có thể bị lạm dụng cho mục đích xấu như tạo nội dung giả mạo trực tuyến hoặc thực hiện các cuộc tấn công “spear phishing” để lừa đảo.

Cuối cùng là trách nhiệm pháp lý. Việc huấn luyện mô hình trên dữ liệu có bản quyền là một vùng xám về pháp lý. Các công ty không muốn tiết lộ dữ liệu họ đã sử dụng để tránh nguy cơ bị kiện tụng.

Đây là những điều mà bất kỳ ai đang làm việc hoặc có ý định khởi nghiệp trong lĩnh vực này cần phải nhận thức. OpenAI đã từng thực hiện những hành động gây tranh cãi và hiện tại họ đang ở vị trí dẫn đầu, cho thấy rằng việc hoạt động trong vùng xám pháp lý là điều không thể tránh khỏi.

English-Centric Open Models

Birds-eye View

Tiếp theo, tôi sẽ nói về các mô hình mở. Đầu tiên, tôi sẽ giới thiệu tổng quan về năm mô hình khác nhau mà tôi đã chọn vì một lý do cụ thể. Hai mô hình đầu tiên là "Pythia" và "OLMo" vì chúng là mã nguồn mở và có thể tái tạo hoàn toàn. Chúng ta biết mọi thứ về chúng, bao gồm dữ liệu đào tạo và quy trình đào tạo. Bạn có thể tải xuống tất cả các tài liệu liên quan để hiểu rõ cách tạo ra một mô hình mạnh mẽ. "Pythia" có nhiều kích thước và checkpoints, điều này khá thú vị. "OLMo" có thể là mô hình tái tạo mạnh nhất hiện nay.

Tiếp theo, chúng ta có các mô hình với trọng số (weights) mở, nhưng không hoàn toàn mở vì không tiết lộ mọi thứ, chẳng hạn như dữ liệu đào tạo hoặc mã nguồn. Tôi sẽ nói về "LLaMa2", mô hình phổ biến nhất, được điều chỉnh an toàn mạnh mẽ. "Mistral" và "Mixtral" là những mô hình mạnh và nhanh, có khả năng đa ngôn ngữ ở mức độ nào đó. Cuối cùng, "Qwen" là một mô hình rất mạnh, đa ngôn ngữ hơn, đặc biệt tốt trong tiếng Anh và tiếng Trung vì được đào tạo trên các ngôn ngữ này.

Pythia - Overview

Đầu tiên, chúng ta sẽ tìm hiểu về Pythia, một dự án của Alther AI - một trong những tổ chức AI mã nguồn mở đầu tiên. Alther AI đã phát triển nhiều công cụ hữu ích như mã huấn luyện mô hình, tập dữ liệu huấn luyện và các phương pháp đánh giá được sử dụng rộng rãi.

Mục tiêu của Pythia là hiểu rõ hơn về động lực huấn luyện mô hình và khả năng mở rộng. Họ đã phát hành tám kích thước mô hình từ 70 triệu đến 12 tỷ tham số. Mỗi kích thước mô hình có 154 checkpoints trong suốt quá trình huấn luyện, với tổng cộng 300 tỷ token được sử dụng. Điều này cho phép nghiên cứu về tốc độ học của các mô hình nhỏ và lớn.

Về kiến trúc, các mô hình có nhiều điểm tương đồng, nhưng cũng có một số khác biệt nhỏ. Ví dụ, mô hình 7 tỷ tham số thường có độ rộng 4096, sâu 32, với 32 attention heads. Đây là một kiến trúc tiêu chuẩn của LLaMa 7B. Khi mở rộng kích thước, số lượng lớp và độ rộng sẽ tăng lên.

Hầu hết các mô hình đều sử dụng Transformer, pre-layer Norm, RoPE embeddings và SwiGLU activation. Tuy nhiên, có sự khác biệt về độ dài ngữ cảnh: Pythia có ngữ cảnh 2K, trong khi LLaMa2 có ngữ cảnh 4K. Một số khác biệt khác bao gồm việc sử dụng bias và loại layer Norm.

Dữ liệu huấn luyện của Pythia bao gồm 300 tỷ token từ "the pile". Họ cũng thực hiện một lần huấn luyện với 207 tỷ token đã được loại bỏ trùng lặp để kiểm tra hiệu quả huấn luyện. Tốc độ học của mô hình 7B là 1.2 * e^-4, trong khi llama là 3 * e^-4. Kích thước batch của Pythia là 2 triệu token, so với 4 triệu token của LLaMa2.

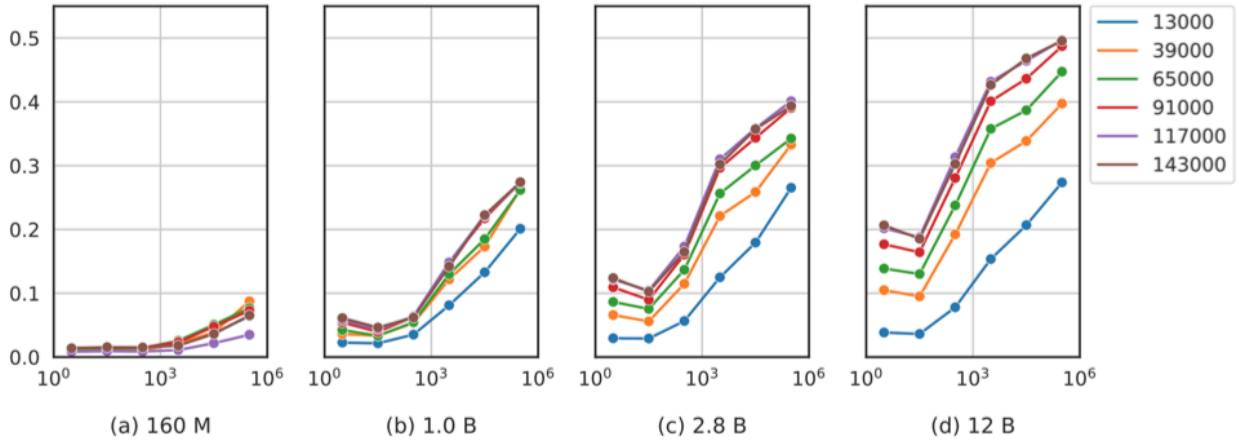
The Pile

“The Pile", một tập dữ liệu mở ban đầu được sử dụng để huấn luyện các mô hình ngôn ngữ lớn. Đây là một tập dữ liệu rất phong phú, bao gồm nhiều loại dữ liệu khác nhau. Đặc biệt, nó được huấn luyện trên dữ liệu học thuật, bao gồm các nguồn như PubMed, arXiv, Free Law, và US patent office. Ngoài ra, tập dữ liệu này còn bao gồm dữ liệu từ internet, được thu thập từ các phần khác nhau của mạng, cũng như từ StackExchange và Wikipedia.

"The Pile" cũng chứa một số dữ liệu văn học, như các bộ dữ liệu sách, dữ liệu mã nguồn, và dữ liệu hội thoại từ phụ đề. Tổng cộng, tập dữ liệu này có dung lượng 800 gigabyte, tương đương với khoảng 300 tỷ tokens.

Pythia - Findings

Một số phát hiện từ bài báo của nhóm nghiên cứu Pythia, nổi bật với việc phân tích khả năng ghi nhớ của mô hình ngôn ngữ và tốc độ học tập dựa trên số lượng token được cung cấp. Đồ thị trong nghiên cứu cho thấy, ở phía bên trái là một trong những mô hình nhỏ hơn với 160 triệu tham số, trong khi bên phải là mô hình lớn nhất với 12 tỷ tham số. Các đường biểu diễn các bước khác nhau trong quá trình huấn luyện, ví dụ như 13,000 bước, 39,000 bước, v.v.



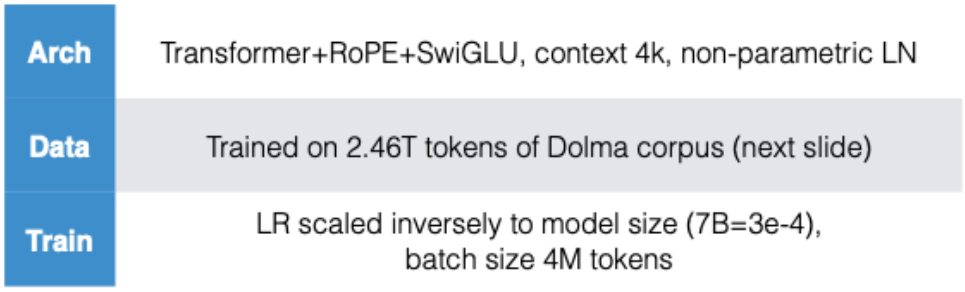
Trục x biểu thị tần suất của một sự kiện trong dữ liệu huấn luyện, còn trục y là độ chính xác của việc trả lời câu hỏi về sự kiện đó. Kết quả cho thấy, khi mở rộng mô hình, các mô hình lớn hơn học nhanh hơn, nhưng chỉ đến một mức độ nhất định. Ví dụ, mô hình 2.8 tỷ tham số có hiệu suất tương đương với mô hình 12 tỷ tham số ở giai đoạn đầu của quá trình huấn luyện. Tuy nhiên, về sau, mô hình 12 tỷ tham số có khả năng ghi nhớ và hồi tưởng nhiều sự kiện hơn.

Một điểm đáng chú ý là tất cả dữ liệu và các checkpoints đều được công khai, giúp tái tạo và phân tích dữ liệu mà mô hình đã được huấn luyện. Nhóm nghiên cứu cũng thực hiện các can thiệp vào dữ liệu để cân bằng tần suất xuất hiện của đại từ nam và nữ, từ đó giảm thiểu sự thiên vị của mô hình trong việc tạo ra đại từ nam.

OLMo - Overview

Tiếp theo, chúng ta sẽ tìm hiểu mô hình OLMo từ Allen Institute for AI. Điều đáng chú ý là cả hai mô hình mở mà tôi đề cập đều đến từ các tổ chức phi lợi nhuận. Điều này có nghĩa là họ ít lo ngại về việc bị kiện vì vi phạm sử dụng hợp lý và tập trung vào việc tạo ra các mô hình tốt hơn cho khoa học.

Mô hình này có hiệu suất hàng đầu và được tài liệu hóa đầy đủ. Một điểm khác biệt lớn là họ sử dụng "non-parametric layer Norm" thay vì "RMS Norm", tức là layer Norm không có tham số. Tuy nhiên, lý do cho sự lựa chọn này không được giải thích rõ ràng.



OLMo được huấn luyện trên 2.46 nghìn tỷ tokens, so với 300 tỷ tokens của phiên bản trước đó. Họ sử dụng một tập dữ liệu gọi là Dolma Corpus, cũng do Allen Institute for AI tạo ra. Mô hình này có tốc độ học là 3e-4, giống như LLaMa, và kích thước batch là 4 triệu tokens, cũng tương tự như LLaMa.

Dolma

Một tập dữ liệu ngôn ngữ lớn có tên là "Dolma", được phát triển với quy mô tương tự như "The Pile" nhưng lớn hơn, với ba nghìn tỷ token. Điều đặc biệt là tập dữ liệu này hoàn toàn mở và có thể tải xuống từ Hugging Face, nếu bạn có đủ dung lượng lưu trữ.

Quy trình xử lý dữ liệu của Dolma bao gồm lọc ngôn ngữ, lọc chất lượng, lọc nội dung, loại bỏ trùng lặp, trộn nguồn đa dạng và mã hóa token. Điều này rất đáng chú ý vì hầu hết các quy trình xử lý dữ liệu của các mô hình ngôn ngữ thường là độc quyền. Do đó, Dolma là một ví dụ tốt để tìm hiểu về quy trình xử lý dữ liệu trong việc huấn luyện mô hình.

Các loại tài liệu trong Dolma bao gồm Common Crawl (dữ liệu thu thập từ Internet với khoảng 2,2 nghìn tỷ token), The Stack (khoảng 400 tỷ token mã nguồn), C4 (dữ liệu web), Reddit, các bài báo khoa học, sách và Wikipedia. Tập dữ liệu này có độ phủ rộng lớn, mặc dù chủ yếu là tiếng Anh.

OLMo - Findings

Trong một nghiên cứu gần đây về mô hình ngôn ngữ OLMo, có một số điểm thú vị đáng chú ý. Đầu tiên, OLMo có hiệu suất trung bình cạnh tranh. Đây là mô hình ngôn ngữ đầu tiên hoàn

toàn mở và được tài liệu hóa trong phạm vi 7 tỷ tham số, có thể cạnh tranh với các mô hình ít mở hơn. Ví dụ, LLaMa2 đạt 70.5 điểm trung bình trên các bộ dữ liệu, Falcon đạt 70.3, MPT đạt 69.8, và OLMo đạt 69.3. Trong khi đó, Pythia chỉ đạt 63 điểm, có thể do thời gian huấn luyện chưa đủ dài.

7B Models	arc challenge	arc easy	boolq	hella-swag	open bookqa	piqa	sciq	wino-grande	avg.
Falcon	47.5	70.4	74.6	75.9	53.0	78.5	93.9	68.9	70.3
LLaMA	44.5	67.9	75.4	76.2	51.2	77.2	93.9	70.5	69.6
Llama 2	48.5	69.5	80.2	76.8	48.4	76.7	94.5	69.4	70.5
MPT	46.5	70.5	74.2	77.6	48.6	77.3	93.7	69.9	69.8
Pythia	44.1	61.9	61.1	63.8	45.0	75.1	91.1	62.0	63.0
RPJ-INCITE	42.8	68.4	68.6	70.3	49.4	76.0	92.9	64.7	66.6
OLMo-7B	48.5	65.4	73.4	76.4	50.4	78.4	93.8	67.9	69.3

Table 6: Zero-shot evaluation of OLMo-7B and 6 other publicly available comparable model checkpoints on 8 core tasks from the downstream evaluation suite described in Section 2.4. For OLMo-7B, we report results for the 2.46T token checkpoint.

Một điểm đáng chú ý khác là hiệu suất của OLMo tiếp tục tăng khi thời gian huấn luyện kéo dài. Khi huấn luyện trên 500 tỷ token, hiệu suất đã cao hơn so với Pythia. Khi tăng lên 2.4 hoặc 2.5 nghìn tỷ token, hiệu suất vẫn tiếp tục cải thiện. Điều này cho thấy việc huấn luyện lâu hơn có thể giúp cải thiện mô hình.

Một câu hỏi đặt ra là liệu OLMo có bị overfitting với dữ liệu kiểm tra hay không. Nhóm nghiên cứu đã thực hiện loại bỏ trùng lặp để giảm thiểu khả năng này, do đó, có khả năng những cải thiện này là thực sự. Ngoài ra, một điểm thú vị khác là tất cả các mô hình này đều có lịch trình điều chỉnh tốc độ học, thường bắt đầu với giai đoạn tăng tốc độ học và sau đó giảm dần, nhưng chỉ giảm đến một mức sàn nhất định, thường là 1/10 tốc độ học ban đầu.

LLaMa2 - Overview

LLaMa2, một mô hình ngôn ngữ mở mạnh mẽ do Meta phát triển. Đây là một trong những mô hình ngôn ngữ mở mạnh nhất hiện nay, mặc dù có thể có những mô hình khác mạnh hơn. Mục tiêu của LLaMa2 là trở thành một mô hình ngôn ngữ mở mạnh mẽ và an toàn, với các phiên bản cơ bản và trò chuyện.

Một điểm nổi bật của LLaMa2 là các biện pháp bảo vệ an toàn mạnh mẽ. Nếu cần chọn một mô hình để sử dụng trong hệ thống giao tiếp trực tiếp với người dùng, tôi sẽ chọn LLaMa2 thay vì các mô hình khác như Mistral, mặc dù Mistral có thể thể hiện hiệu suất vượt trội trong một số trường hợp. Lý do là Mistral có thể đưa ra những câu trả lời không mong muốn cho người dùng.

Dữ liệu huấn luyện của LLaMa2 không được công khai, nhưng theo thông tin từ nhóm phát triển, mô hình được huấn luyện trên các nguồn công khai, ưu tiên các nguồn thông tin chính xác nhất. Bài báo về LLaMa1 cung cấp nhiều thông tin hơn, và có thể suy đoán rằng phương pháp huấn luyện của LLaMa2 tương tự. Tổng số lượng token dùng để huấn luyện là 2 nghìn tỷ, ít hơn so với một số mô hình khác.

Arch	Transformer+RoPE+SwiGLU, context 4k, RMSNorm
Data	Trained on “public sources, up-sampling the most factual sources”, LLaMa 1 has more info (next page), total 2T tokens
Train	7B=3e-4, batch size 4M tokens

LLaMa 1 - Training Data

Trong quá trình huấn luyện mô hình LLaMa1, dữ liệu được sử dụng bao gồm Common Crawl, C4, GitHub, Wikipedia, sách, ArXiv và Stack Exchange. Đáng chú ý, nhóm nghiên cứu đã tăng cường (upsample) dữ liệu từ Wikipedia và sách, đồng thời giảm bớt (downsample) dữ liệu từ GitHub so với lượng dữ liệu thực tế có sẵn. Cụ thể, họ đã thực hiện 2.45 vòng lặp (epoch) với dữ liệu Wikipedia, 2.23 vòng với sách, chỉ một vòng với dữ liệu web tiêu chuẩn, ArXiv và Stack Exchange, và 0.64 vòng với dữ liệu GitHub. Điều này cho thấy họ đánh giá cao giá trị của dữ liệu từ Wikipedia và sách, và muốn mô hình học tốt từ những nguồn này. Có thể họ đã thực hiện việc tăng cường dữ liệu thực tế theo cách tương tự.

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Trong quá trình huấn luyện mô hình, không nhất thiết phải lưu các checkpoints sau mỗi Epoch. Thay vào đó, bạn có thể lưu sau mỗi 10.000 bước hoặc một khoảng thời gian nhất định. Việc này giúp tránh cảm giác rằng việc huấn luyện mất quá nhiều thời gian khi dữ liệu huấn luyện lớn. Thực tế, nhiều mô hình ngôn ngữ không huấn luyện trên toàn bộ dữ liệu từ web vì chi phí quá cao, mặc dù họ có sẵn dữ liệu. Do đó, việc lưu checkpoints định kỳ có thể giúp quản lý tài nguyên hiệu quả hơn và khuyến khích sử dụng dữ liệu huấn luyện lớn hơn mà không lo ngại về thời gian huấn luyện kéo dài.

LLaMa2 - Reward Model

Tiếp theo, chúng ta sẽ tìm hiểu về quá trình tinh chỉnh an toàn cho các mô hình LLaMa2. Các nhà phát triển LLaMa2 đã đầu tư nhiều công sức để đảm bảo mô hình hoạt động an toàn, không chỉ để tránh các sự cố PR mà còn để tạo ra một sản phẩm thực sự an toàn cho người dùng.

Đầu tiên, họ thu thập dữ liệu để xây dựng mô hình phần thưởng (mô hình ưu tiên). Quá trình này bao gồm việc xếp hạng các đầu ra của mô hình dựa trên sự ưu tiên. Các bộ dữ liệu như "anthropic helpful and harmless", dữ liệu từ OpenAI, và dữ liệu từ Stack Exchange đã được sử dụng để phân loại các câu trả lời hữu ích và không hữu ích. Ngoài ra, họ còn sử dụng dữ liệu từ Stanford để tìm các bài đăng trên Reddit có nhiều lượt upvote hơn dù được đăng sau.

Dataset	Num. of Comparisons	Avg. # Turns per Dialogue	Avg. # Tokens per Example	Avg. # Tokens in Prompt	Avg. # Tokens in Response
Anthropic Helpful	122,387	3.0	251.5	17.7	88.4
Anthropic Harmless	43,966	3.0	152.5	15.7	46.4
OpenAI Summarize	176,625	1.0	371.1	336.0	35.1
OpenAI WebGPT	13,333	1.0	237.2	48.3	188.9
StackExchange	1,038,480	1.0	440.2	200.1	240.2
Stanford SHP	74,882	1.0	338.3	199.5	138.8
Synthetic GPT-J	33,139	1.0	123.3	13.0	110.3
Meta (Safety & Helpfulness)	1,418,091	3.9	798.5	31.4	234.1
Total	2,919,326	1.6	595.7	108.2	216.9

Meta cũng thu thập một lượng lớn dữ liệu nội bộ để tinh chỉnh LLaMa qua nhiều phiên bản. Họ cho phép người dùng thử nghiệm mô hình và thu thập dữ liệu từ những người cố gắng "phá" mô hình bằng cách khiến nó đưa ra các câu trả lời không phù hợp. Dữ liệu này sau đó được sử dụng để cải thiện mô hình qua từng phiên bản.

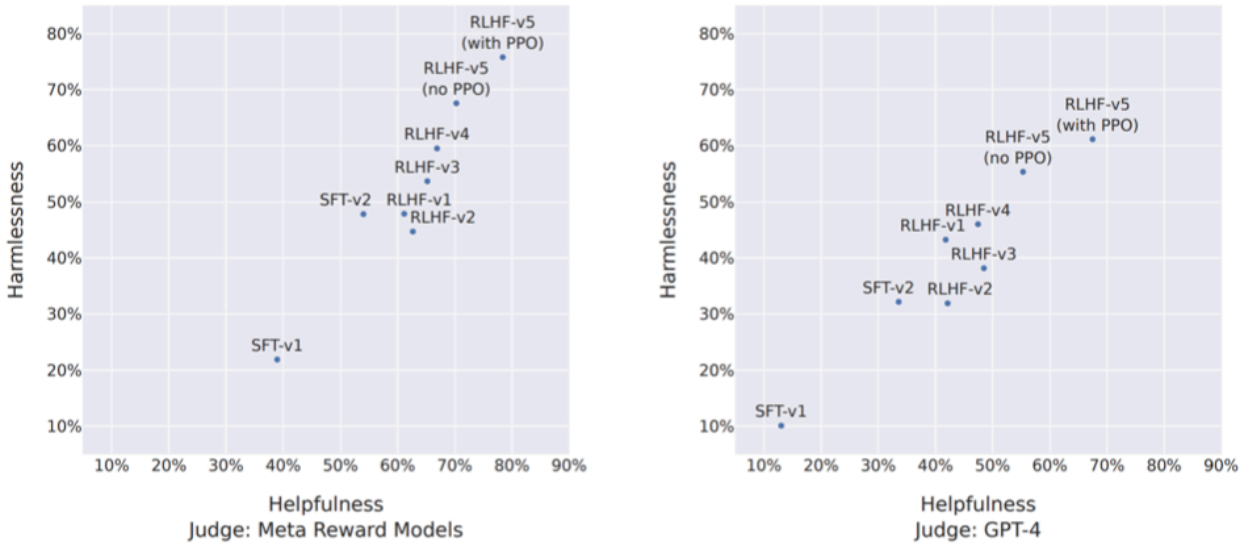
Tiếp theo, họ huấn luyện mô hình để tuân theo các ưu tiên này, bằng cách dự đoán sự ưu tiên của con người giữa hai đầu ra của mô hình ngôn ngữ. Đây là một thách thức lớn vì cả hai đầu ra đều có thể trông rất tự nhiên. Để giải quyết, họ tạo ra hai mô hình phần thưởng riêng biệt: một để phân biệt an toàn và một để phân biệt tính hữu ích. Mô hình phần thưởng cho tính hữu ích hoạt động tốt hơn trong việc phân biệt các đầu ra hữu ích, trong khi mô hình an toàn làm tốt hơn trong việc phân biệt các đầu ra an toàn.

	Meta Helpful.	Meta Safety	Anthropic Helpful	Anthropic Harmless	OpenAI Summ.	Stanford SHP	Avg
SteamSHP-XL	52.8	43.8	66.8	34.2	54.7	75.7	55.3
Open Assistant	53.8	53.4	67.7	68.4	71.7	55.0	63.0
GPT4	58.6	58.1	-	-	-	-	-
Safety RM	56.2	64.5	55.4	74.7	71.7	65.2	64.3
Helpfulness RM	63.2	62.8	72.0	71.0	75.5	80.0	70.6

Cuối cùng, họ nhận thấy rằng việc sử dụng mô hình ngôn ngữ lớn và mạnh mẽ là rất quan trọng trong việc xác định phần thưởng. Mô hình 70 tỷ tham số mà họ sử dụng cho kết quả tốt hơn nhiều so với các mô hình nhỏ hơn trong việc dự đoán phần thưởng.

LLaMa2 - RLHF

Nhóm nghiên cứu đã áp dụng một quy trình huấn luyện gia tăng để đảm bảo an toàn. Đầu tiên, họ phát triển một mô hình được tinh chỉnh dưới sự giám sát mà không sử dụng học tăng cường. Mô hình thứ hai được cải thiện đáng kể về mức độ hữu ích. Sau đó, họ tiếp tục áp dụng phương pháp huấn luyện với phản hồi từ con người (RLHF), bắt đầu từ mô hình tinh chỉnh ban đầu, dần dần bổ sung thêm dữ liệu thưởng và huấn luyện với một mô hình thưởng tốt hơn. Cuối cùng, họ đạt được mô hình tối ưu nhất, và đây chính là mô hình được phát hành. Nhóm nghiên cứu đã đầu tư rất nhiều công sức để đảm bảo mô hình này an toàn, đây cũng là một trong những điểm chính của bài báo mà họ công bố.



LLaMa2 - Chat Instruction Following

Một phần thú vị trong bài báo về LLaMa2 là cách họ khiến mô hình tuân theo các hướng dẫn trong hội thoại. Trong lớp học, chúng ta đã thảo luận về việc sử dụng system message, user message, và assistant message để hướng dẫn mô hình ngôn ngữ. System message là những chỉ dẫn cần được tuân thủ trong suốt cuộc hội thoại. Để đạt được điều này, mô hình cần chú ý đặc biệt đến system message.

Ví dụ, nếu bạn yêu cầu mô hình chỉ viết bằng biểu tượng cảm xúc, bạn muốn nó duy trì điều này trong suốt cuộc hội thoại. Mô hình không tự nhiên làm điều này, vì vậy nhóm nghiên cứu đã thực hiện một bước tạo dữ liệu. Họ yêu cầu một mô hình hiện có viết bằng biểu tượng cảm xúc, sau đó tạo ra một tập dữ liệu từ các phản hồi này. Tập dữ liệu này sau đó được dùng để huấn luyện mô hình, giúp nó chú ý hơn đến system message.

Họ thử nghiệm với nhiều quy tắc khác nhau như viết như thể đang giải thích cho trẻ 5 tuổi, viết một cách lịch sự, hoặc viết một cách không trang trọng. Bằng cách tạo ra dữ liệu tổng hợp này, họ có thể huấn luyện mô hình chú ý kỹ lưỡng hơn đến system message để cải thiện hiệu suất.

Data Generation Phase

```
System: Write in only emojis.
User: Write in only emojis. Say hello.
Assistant: [generates]👋
User: Write in only emojis. How are you doing.
Assistant: [generates]😊💗
```

Training Phase

```
System: Write in only emojis.
User: Say hello.
Assistant: 👋
User: How are you doing.
Assistant: 😊💗
```

Tuy nhiên, thông tin chi tiết về dữ liệu huấn luyện của LLaMa2 không được công bố, nên chúng ta chỉ có thể suy luận từ những gì đã biết.

Mistral/Mixtral - Overview

Tiếp theo, chúng ta sẽ tìm hiểu về Mixtral, một mô hình ngôn ngữ mở mạnh mẽ và đa ngôn ngữ được phát triển bởi công ty Mistral AI. Mặc dù thông tin chi tiết về quá trình huấn luyện không được tiết lộ nhiều, Mixtral nổi bật với một số tính năng độc đáo như tối ưu hóa tốc độ, bao gồm "grouped query attention" và "mixture of experts".

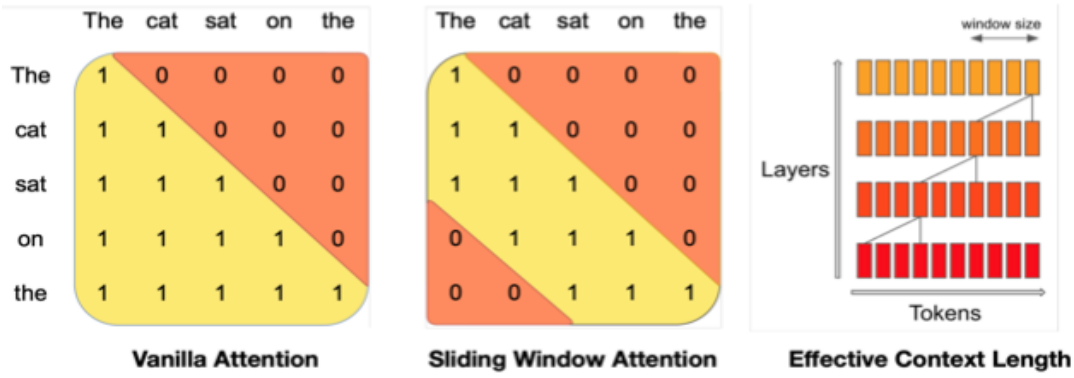
Khác với các mô hình khác, Mistral thực hiện một số thay đổi kiến trúc như "sliding window attention" và "mixture of experts". Dữ liệu huấn luyện của Mistral không được công bố đầy đủ, nhưng đáng chú ý là nó bao gồm tiếng Anh và các ngôn ngữ châu Âu, lý thuyết cho thấy nó có thể vượt trội hơn LLaMa trong việc xử lý các ngôn ngữ này.

Arch	Transformer+RoPE+SwiGLU, context 4k, RMSNorm, sliding window attention. Mixtral has 8x experts in feed-forward layer
Data	Not disclosed? But includes English and European languages
Train	Not disclosed?

Đối với LLaMa, theo tài liệu, mô hình này chủ yếu tập trung vào tiếng Anh với khoảng 85% dữ liệu, 8% là mã lập trình và chỉ 0,3% là các ngôn ngữ khác. Điều này cho thấy LLaMA không thực sự đa ngôn ngữ và chủ yếu nhắm đến việc tạo ra một mô hình tiếng Anh tốt. Thông tin chi tiết về quá trình huấn luyện của LLaMA cũng không được công bố rõ ràng.

Mistral - Sliding Window Attention

Mistral sử dụng cơ chế attention với cửa sổ trượt. Thay vì luôn chú ý đến tất cả các phần tử trước đó trong chuỗi như attention thông thường, Mistral chỉ chú ý đến n phần tử trước đó, với n là 4096. Điều này cho phép mô hình có thể nhìn lại 4096 phần tử trong mỗi lớp, và với nhiều lớp, số lượng phần tử có thể chú ý sẽ tăng lên mà không làm tăng đáng kể chi phí huấn luyện. Độ dài attention cho mỗi token vẫn giữ nguyên, giúp tối ưu hóa hiệu suất.



Mistral, sử dụng mô hình mixture of experts. Mặc dù có ít tham số hơn (45 tỷ tham số so với 70 tỷ của LLaMa), Mistral vẫn mạnh mẽ hơn trong nhiều tác vụ và dễ triển khai hơn. Điều này làm cho Mistral trở thành lựa chọn tốt nếu bạn cần một mô hình mạnh mẽ nhưng nhẹ nhàng hơn.

Một điểm đáng chú ý khác là cách Mistral xử lý độ dài ngữ cảnh. Với độ dài ngữ cảnh là 4096, mô hình thực sự có thể xử lý một khối dữ liệu gấp đôi kích thước đó, tức là 8192 token. Điều này cho phép mỗi token có thể chú ý đến tất cả các token trước đó trong khối, giúp mô hình có khả năng xử lý thông tin từ xa một cách hiệu quả.

Qwen - Overview

Tiếp theo, tôi muốn giới thiệu về mô hình đa ngôn ngữ mạnh mẽ có tên là Qwen, được phát triển bởi Alibaba. Mặc dù ở Mỹ, mô hình này có thể chưa được chú ý nhiều, nhưng nó nổi bật với khả năng xử lý đa ngữ, đặc biệt là tiếng Anh và tiếng Trung, cùng với nhiều ngôn ngữ khác.

Một trong những đặc điểm nổi bật của Qwen là từ vựng lớn, hỗ trợ tốt cho việc xử lý đa ngữ. Mô hình này có nhiều phiên bản kích thước khác nhau, bao gồm phiên bản 7B và phiên bản lớn hơn như 70B hoặc 72B. Qwen sử dụng kiến trúc tiêu chuẩn, với một điểm khác biệt nhỏ là có thêm bias trong lớp attention, điều mà LLaMa không có.

Arch	Transformer+RoPE+SwiGLU, context 4k, RMSNorm, bias in attention layer
Data	Trained on multilingual data + instruction data at pre-training time, 2-3T tokens
Train	3e-4, batch size 4M tokens

Qwen được huấn luyện trên dữ liệu đa ngữ với từ vựng lên đến 150k, so với 32k của LLaMa, giúp nó xử lý dữ liệu đa ngữ hiệu quả hơn. Mặc dù có nhiều điểm tương đồng với LLaMa về chế độ huấn luyện, nhưng Qwen có thể mạnh hơn nhờ vào kỹ thuật xử lý dữ liệu khác biệt.

Qwen - Multilinguality

Một điểm thú vị từ nghiên cứu là cách Qwen xử lý token hóa từ con. Trong các subword model, dữ liệu đầu vào được chia nhỏ, với các token thường xuyên có đầu ra dài hơn và ngược lại. Điều này gây ra vấn đề khi xử lý đa ngữ, vì dữ liệu ít sẽ bị chia nhỏ thành ký tự hoặc byte, làm tăng chi phí và giảm hiệu quả.

Khi so sánh với các mô hình khác như XLM-R, Qwen cho thấy khả năng token hóa tương đương, thậm chí ít chia nhỏ hơn khi xử lý mã nguồn. Điều này chứng tỏ Qwen có khả năng xử lý đa ngữ mạnh mẽ.

Cuối cùng, Quen không chỉ nổi bật về khả năng đa ngữ mà còn có hiệu suất mạnh mẽ, đứng đầu trong nhiều bảng xếp hạng mô hình ngôn ngữ mở. Một ví dụ là mô hình được tinh chỉnh bởi Abus AI, dựa trên Qwen, đã đạt được thành công lớn.

Other Models

Code Models

Trong phần tiếp theo, tôi muốn giới thiệu về các mô hình mục đích đặc biệt, cụ thể là các mô hình tạo mã. Hiện nay, hầu hết các mô hình ngôn ngữ lớn đều được huấn luyện trên mã nguồn, vì việc tạo mã là một ứng dụng quan trọng. Ngoài ra, nghiên cứu cho thấy rằng huấn luyện trên mã nguồn có thể cải thiện khả năng suy luận của các mô hình ngôn ngữ.

Một số mô hình nổi bật bao gồm StarCoder 2, một mô hình mã nguồn mở hoàn toàn với dữ liệu và chi tiết huấn luyện được công khai. Đây là một mô hình mạnh mẽ và đáng chú ý. Tiếp theo là CodeLlama của Meta, một phiên bản thích ứng từ LLaMa, cũng đạt hiệu suất tốt. Cuối cùng là DeepSeek Coder, một mô hình khác đang dẫn đầu trên nhiều bảng xếp hạng.

Tất cả ba mô hình này đều rất cạnh tranh và có thể được coi là tốt nhất trong lĩnh vực tạo mã. Tuy nhiên, tôi sẽ không đi sâu vào chi tiết vì chúng ta sẽ có một bài học riêng về tạo mã và các vấn đề liên quan sau này.

Math Models

Đối với mô hình ngôn ngữ lớn, khả năng xử lý toán học của chúng thường không được đánh giá cao. Tuy nhiên, đã có một số mô hình được huấn luyện đặc biệt cho toán học. Một trong số đó là mô hình LLeMa, được phát triển bởi EleutherAI. Đây là một mô hình hoàn toàn mở, với tất cả dữ liệu và thông tin đều công khai.

Ngoài ra, DeepSeek cũng đã tạo ra một mô hình toán học mạnh mẽ DeepSeek Math, có khả năng cạnh tranh với GPT-4 trong nhiều bài toán. Họ đã huấn luyện một bộ phân loại để xác định và thu thập dữ liệu liên quan đến toán học từ web, sau đó tinh chỉnh mô hình dựa trên dữ liệu này. Nguồn dữ liệu chất lượng cao được lấy từ các nguồn như Proof Pile và nhiều nguồn khác.

Mặc dù có thể nghĩ đến việc sử dụng các phương pháp như học tăng cường dựa trên việc mô hình có đưa ra đáp án đúng hay không, nhưng hiện tại, điều này chưa phải là thành phần chính trong các mô hình toán học này.

Science Model: Galactica

Meta đã phát triển một mô hình ngôn ngữ có tên Galactica, được thiết kế để hỗ trợ trong lĩnh vực khoa học. Mô hình này ra mắt cách đây khoảng hai năm và đã gặp phải một số chỉ trích do khả năng "hallucination" và đưa ra thông tin khoa học sai lệch, điều khá phổ biến với các mô hình ngôn ngữ thời điểm đó. Mặc dù bị chỉ trích, Galactica thực sự có nhiều điểm đáng chú ý. Nó được thiết kế như một mô hình đa năng cho khoa học, có khả năng hiểu không chỉ văn bản mà còn nhiều loại dữ liệu khoa học khác nhau, bao gồm cả latex, mã nguồn, cấu trúc phân tử, collagen, DNA, và nhiều hơn nữa. Mặc dù gặp phải sự cố PR, nhưng công việc và ý tưởng đằng sau Galactica rất đáng được ghi nhận. Hy vọng rằng trong tương lai, sẽ có nhiều mô hình ngôn ngữ dành cho khoa học được phát triển hơn nữa.

Closed Models

GPT-4 - Overview

Phần cuối cùng, chúng ta sẽ thảo luận về các mô hình ngôn ngữ đóng và một số khả năng nổi bật của chúng. Mặc dù thông tin chi tiết về cách xây dựng các mô hình này thường không được công khai, nhưng chúng ta có thể tìm hiểu về khả năng của chúng thông qua các bài viết và báo cáo đánh giá.

Một ví dụ điển hình là mô hình GPT-4, được coi là tiêu chuẩn mạnh mẽ trong lĩnh vực mô hình ngôn ngữ. GPT-4 không chỉ nổi bật trong việc xử lý ngôn ngữ mà còn hỗ trợ đầu vào hình ảnh và có khả năng gọi các công cụ bên ngoài thông qua “function calling interface”. Điều này cho phép GPT-4 thực hiện các tác vụ đa phương tiện, chẳng hạn như chuyển đổi dữ liệu thành định dạng JSON hoặc tạo hình ảnh từ mô tả văn bản.

Ngoài ra, GPT-4 còn có khả năng tạo mã và hiển thị kết quả, như tạo biểu đồ từ dữ liệu số. Các nỗ lực hiện tại đang hướng tới việc phát triển các mô hình ngôn ngữ mã nguồn mở có thể thực hiện những tác vụ tương tự, yêu cầu sự kết hợp giữa đa phương tiện và khả năng sử dụng công cụ.

Một điểm thú vị là cách GPT-4 xử lý đầu vào hình ảnh và gọi công cụ tạo hình ảnh DALLE-3. Khi yêu cầu tạo hình ảnh, GPT-4 sẽ cung cấp mô tả cho DALLE-3 thông qua API để tạo ra hình ảnh mong muốn. Điều này cho thấy sự linh hoạt và khả năng tích hợp của GPT-4 trong việc xử lý các tác vụ phức tạp.

Gemini

Gemini và Claude được xem là hai mô hình có khả năng cạnh tranh với GPT-4 về độ chính xác. Gemini, một mô hình mới của Google, có hai phiên bản: Gemini Pro và Gemini Ultra. Một điểm nổi bật của Gemini Pro là khả năng xử lý đầu vào rất dài, từ 1 đến 10 triệu token, và hỗ trợ cả đầu vào hình ảnh, video cũng như đầu ra hình ảnh. Tôi đã thử nghiệm đưa một video vào và nhận thấy khả năng nhận diện video của nó khá ấn tượng. Bạn có thể thử nghiệm để trải nghiệm tính năng này.

Claude 3 - Overview

Claude, một mô hình ngôn ngữ mới với khả năng xử lý ngữ cảnh lên đến 200k và hỗ trợ xử lý hình ảnh. Claude cho thấy kết quả mạnh mẽ, cạnh tranh với GPT-4. Nếu bạn đang tìm kiếm các mô hình đóng để thử nghiệm, Claude là một lựa chọn đáng cân nhắc.

Một điểm thú vị khác là làm thế nào để các mô hình mở có thể thể hiện những khả năng đặc biệt mà chúng ta thấy ở các mô hình đóng, nhằm mang lại lợi ích cho mọi người và chia sẻ công thức phát triển các mô hình này.

Cách đánh giá mô hình ngôn ngữ

Đầu tiên, bạn có thể xem kết quả Benchmark đã được công bố. Tuy nhiên, cần lưu ý rằng các kết quả này có thể không hoàn toàn chính xác do các phương pháp đánh giá khác nhau. Ví dụ, trong bài báo về mô hình Gemini, họ so sánh Gemini Pro và Gemini Ultra với GPT-4 và GPT-3.5. Mặc dù kết quả cho thấy Gemini vượt trội, nhưng phương pháp prompting mô hình lại khác nhau, và các API liên tục được cải thiện.

Chúng tôi đã thực hiện một nghiên cứu so sánh Gemini Pro và GPT-3.5 Turbo và nhận thấy rằng GPT-3.5 Turbo thực sự hoạt động tốt hơn trong nhiều trường hợp. Điều này cho thấy rằng không nên hoàn toàn tin tưởng vào các kết quả công bố mà không kiểm chứng.

Để đánh giá mô hình một cách có hệ thống, bạn có thể sử dụng các công cụ như LM Evaluation Harness (EleutherAI), giúp dễ dàng đánh giá các mô hình trên nhiều nhiệm vụ khác nhau. Công cụ này đặc biệt hữu ích cho các mô hình mã nguồn mở.

Resources

1. https://phontron.com/class/anlp2024/lectures/#tour-of-modern-large-language-models-mar-12
2. Levels of Release in LMs (Liang et al. 2022)
3. Pythia (Biderman et al. 2023)
4. The Pile (Gao et al. 2021)
5. OLMo (Groeneveld et al. 2024)
6. LLaMa 2 (Touvron et al. 2023)
7. Context Distillation (Askell et al. 2021)

8. Mistral (Jiang et al. 2023)
9. Mixtral (Jiang et al. 2023)
10. Qwen (Bai et al. 2023)
11. Starcoder (Li et al. 2023)
12. Code LLaMA (Rozière et al. 2023)
13. Llema (Azerbayev et al. 2023)
14. Galactica (Taylor et al. 2022)
15. GPT-4 (OpenAI 2023)
16. Gemini (Gemini Team 2023)
17. Claude (Anthropic 2023)