**Case Study: LLMs**

What are the LLMs ? (LLMs là gì ?):

Large Language Model (LLM – mô hình ngôn ngữ lớn) là một loại **mô hình trí tuệ nhân tạo** được huấn luyện trên một khối lượng dữ liệu văn bản khổng lồ, với mục tiêu học cách nhận diện, dự đoán và sinh ngôn ngữ tự nhiên.

Cụ thể hơn:

* **Cấu trúc:** LLM thường dựa trên kiến trúc Transformer (ví dụ như GPT, BERT, LLaMA). Transformer giúp mô hình xử lý chuỗi văn bản dài, nắm bắt được mối quan hệ giữa các từ trong ngữ cảnh.
* **Đặc điểm:**
  + Có hàng tỷ (hoặc thậm chí hàng nghìn tỷ) tham số.
  + Học từ lượng dữ liệu văn bản khổng lồ: sách, bài báo, trang web, mã nguồn, v.v.
  + Không chỉ "ghi nhớ" mà còn **khái quát hóa** để sinh ra ngôn ngữ mới, hợp ngữ cảnh.
* **Khả năng:**
  + Hiểu và trả lời câu hỏi.
  + Viết văn bản, dịch thuật, tóm tắt.
  + Sinh mã lập trình, phân tích dữ liệu.
  + Thậm chí có thể suy luận, sáng tạo hoặc đối thoại tự nhiên.
* **Ví dụ nổi bật:** ChatGPT (GPT-4, GPT-5), Google Gemini, Claude, LLaMA, Mistral.

Nhận thức (Percepts) của LLMs là gì ?

Trong **trí tuệ nhân tạo** nói chung, “**percepts**” (nhận thức/đầu vào cảm nhận được) là thông tin mà **agent** (tác tử) tiếp nhận từ môi trường để xử lý. Ví dụ: robot có percepts là hình ảnh từ camera, âm thanh từ micro, hay dữ liệu từ cảm biến.

Với **Large Language Models (LLMs)**:

* **Percepts của LLMs chính là văn bản (text tokens)** mà mô hình nhận làm đầu vào.
* Các ký tự, từ, câu… đều được biến đổi thành **tokens** (mã số biểu diễn từ/ngữ) rồi thành **vector số học** trong không gian nhiều chiều. Đây chính là cách mô hình "cảm nhận" dữ liệu.
* Khác với con người có nhiều giác quan, LLMs **chỉ có một kênh nhận thức chính: ngôn ngữ** (dù hiện nay có xu hướng mở rộng thành Multimodal Models – nhận thêm hình ảnh, âm thanh, video).

Kết luận:

* **Percepts của LLM = chuỗi tokens từ dữ liệu văn bản đầu vào.**
* Nếu là **Multimodal LLMs** thì percepts có thể gồm văn bản, hình ảnh, âm thanh... tất cả đều được chuyển thành dạng vector để xử lý.

Hành động (Actions) của LLMs là gì?

Nếu **percepts của LLMs là văn bản đầu vào (tokens)** thì:

**Actions của LLMs**

* **Action chính của LLM** là sinh ra **chuỗi tokens đầu ra** → tức là **tạo văn bản** (text generation).
* Mỗi action cụ thể chính là việc **dự đoán và chọn ra token tiếp theo** dựa trên ngữ cảnh trước đó.
* Dần dần, các tokens này ghép lại thành câu, đoạn, hoặc toàn bộ văn bản mà Hoàng thấy.

Ví dụ:

* Input (percept): "Hôm nay trời rất..."
* Action (mỗi bước): dự đoán token tiếp theo → "đẹp", "mát", "nóng"…
* Output: "Hôm nay trời rất đẹp."

**Các dạng action mà LLM có thể thể hiện trong ứng dụng thực tế**

* **Sinh ngôn ngữ tự nhiên**: trả lời câu hỏi, viết văn bản, tóm tắt, dịch.
* **Sinh mã lập trình**: viết code, gợi ý sửa lỗi.
* **Suy luận**: giải toán, lập luận logic.
* **Tương tác hệ thống**: nếu LLM được “gắn” vào một agent, action có thể là *gọi API, truy cập web, điều khiển robot, hoặc chạy một lệnh hệ thống*.

Nói cách khác, **hành động gốc của LLM là “sinh token”, nhưng khi tích hợp vào hệ thống rộng hơn, hành động đó có thể được diễn dịch thành rất nhiều dạng hành động đa dạng.**

Mục tiêu (Objectives) của LLMs là gì?

Khi nói về **mục tiêu**, có 2 góc nhìn:

**1. Ở mức huấn luyện (training objective)**

* **Mục tiêu cốt lõi của LLMs** là:
  + **Dự đoán token tiếp theo** trong chuỗi văn bản.
  + Hay còn gọi là **Next Token Prediction / Language Modeling Objective**.
* Ví dụ:
  + Văn bản gốc: "Hôm nay trời rất đẹp."
  + Mô hình thấy: "Hôm nay trời rất" → phải dự đoán "đẹp".
* Quá trình này lặp đi lặp lại hàng tỷ lần, giúp mô hình học được quy luật ngôn ngữ, cú pháp, ngữ nghĩa, và một phần tri thức về thế giới.

Ngoài ra, sau huấn luyện gốc, mô hình còn có thể được **tinh chỉnh (fine-tuning)** với các mục tiêu khác:

* **Instruction tuning**: học cách làm theo hướng dẫn.
* **Reinforcement Learning from Human Feedback (RLHF)**: tối ưu để phản hồi phù hợp với mong đợi của con người (hữu ích, an toàn, ít sai lệch).

**2. Ở mức ứng dụng (task objective)**

Khi được sử dụng, mục tiêu của LLM phụ thuộc vào tác vụ cụ thể:

* Viết văn bản: tạo ra đoạn viết mạch lạc, tự nhiên.
* Trả lời câu hỏi: đưa thông tin đúng và rõ ràng.
* Tóm tắt: rút gọn mà vẫn giữ ý chính.
* Dịch thuật: chuyển ngữ chính xác, trôi chảy.
* Sinh code: tạo đoạn mã đúng cú pháp, đúng logic.

Kết luận:

* **Mục tiêu huấn luyện:** dự đoán token tiếp theo (language modeling).
* **Mục tiêu sử dụng:** phụ thuộc vào tác vụ (viết, dịch, giải thích, tạo code…).

AGI (Artificial General Intelligence) là gì và sự khác biệt giữa AGI và LLMs ?

**LLM**: là mô hình xử lý ngôn ngữ, có khả năng làm nhiều tác vụ (viết, dịch, tóm tắt, phân loại…).

Khi thấy cụm “**general-purpose**” (đa mục đích), nhiều người tự hỏi:

* “Nếu một LLM đã có thể làm được nhiều việc như vậy, liệu nó có phải là **AGI** không?”

**AGI (Artificial General Intelligence) là gì?**

* **AGI** là trí tuệ nhân tạo có khả năng:
  + **Học và thích nghi** với nhiều lĩnh vực, không giới hạn ở một tác vụ cụ thể.
  + **Suy luận, sáng tạo, giải quyết vấn đề** trong các tình huống mới, không cần huấn luyện lại từ đầu.
  + Hoạt động linh hoạt như trí tuệ con người.

**Khác biệt giữa LLM và AGI**

* **LLM hiện nay**:
  + Chủ yếu xử lý ngôn ngữ (text).
  + Chưa có hiểu biết thực sự, chưa có ý thức, chưa có khả năng tự định hướng mục tiêu.
  + Khả năng “đa mục đích” vẫn **giới hạn trong phạm vi ngôn ngữ**.
* **AGI**:
  + Có thể kết hợp ngôn ngữ, hình ảnh, âm thanh, vận động…
  + Biết học hỏi và hành động trong thế giới thực.
  + Có mức độ linh hoạt, tự chủ gần như con người.

Dấu hỏi “**AGI?**” là cách nhấn mạnh sự tranh luận: **LLMs đã tiến gần tới AGI chưa, hay vẫn chỉ là công cụ xử lý ngôn ngữ mạnh mẽ?**

**What do LLMs do? (LLMs làm gì?)**

Các Mô hình ngôn ngữ lớn (LLMs) là hệ thống trí tuệ nhân tạo được thiết kế để hiểu và sinh ra ngôn ngữ tự nhiên của con người. Chúng được huấn luyện trên một lượng dữ liệu văn bản khổng lồ và học được các mẫu thống kê về từ ngữ, câu và khái niệm. Nhờ đó, chúng có thể thực hiện nhiều nhiệm vụ khác nhau:

1. Sinh văn bản: Viết luận, truyện, tóm tắt hoặc nội dung sáng tạo.
2. Trả lời câu hỏi: Giải đáp các câu hỏi thực tế, khái niệm hoặc ứng dụng.
3. Dịch thuật: Chuyển đổi văn bản giữa các ngôn ngữ.
4. Tóm tắt: Rút gọn các đoạn văn dài thành nội dung ngắn gọn, rõ ràng.
5. Tìm kiếm và giải thích thông tin: Giải thích các khái niệm ở mức cơ bản hoặc nâng cao.
6. Hội thoại và trò chuyện: Thực hiện các cuộc đối thoại tự nhiên với người dùng.
7. Hỗ trợ học tập và công việc: Giúp viết code, giải quyết vấn đề và nghiên cứu.

LLMs phân tích văn bản đầu vào, dự đoán các từ tiếp theo hợp lý nhất, và sinh ra câu trả lời mạch lạc, phù hợp ngữ cảnh, khiến chúng trở thành công cụ đa năng cho giao tiếp, sáng tạo và giải quyết vấn đề.

**Do LLMs act rationally? (Liệu LLMs có hành động một cách hợp lý không?)**

Theo nghĩa truyền thống trong trí tuệ nhân tạo, một tác nhân “hợp lý” (rational agent) là tác nhân luôn chọn hành động tối ưu để đạt được mục tiêu của mình dựa trên thông tin có sẵn.

LLMs thì khác:

1. Không có mục tiêu nội tại – LLMs không tự đặt ra mục đích. Chúng chỉ dự đoán từ kế tiếp trong văn bản dựa trên xác suất thống kê được học từ dữ liệu.
2. Không đánh giá hậu quả dài hạn – Khi tạo ra câu trả lời, chúng không suy nghĩ xem kết quả đó có giúp “đạt mục tiêu” hay không, mà chỉ tập trung vào việc tạo ra chuỗi từ hợp lý trong ngữ cảnh.
3. Hành vi có vẻ hợp lý – Vì LLMs được huấn luyện trên kho dữ liệu khổng lồ từ con người, nên đầu ra thường trông hợp lý, logic và thuyết phục. Tuy nhiên, đây là sự bắt chước hợp lý chứ không phải sự suy luận thực sự.

LLMs không thật sự hành động một cách hợp lý như con người hoặc một tác nhân AI có mục tiêu rõ ràng, mà chỉ tái hiện lại sự hợp lý dựa trên mẫu dữ liệu chúng đã học.

**CONCLUSION:**

How do LLMs reason and what are the limits? (LLMs lý luận như thế nào và

giới hạn của chúng là gì?)

Các mô hình ngôn ngữ lớn (LLMs) “lý luận” thông qua **nhận diện mẫu thống kê**, chứ không phải suy luận ký hiệu hay nhân quả thực sự. Chúng dự đoán token tiếp theo dựa trên các mối tương quan học được từ khối dữ liệu khổng lồ, nhờ đó có thể “mô phỏng” các dạng suy luận như diễn dịch, so sánh tương tự hay giải quyết vấn đề theo từng bước.

**Giới hạn của khả năng suy luận trong LLMs gồm:**

* Thiếu **sự hiểu biết thật sự**: không có mô hình thế giới hay ý thức.
* Dễ sinh ra **ảo giác (hallucination)**: tạo ra thông tin nghe có vẻ hợp lý nhưng sai.
* Yếu trong **lý luận nhân quả**: khó khăn với các vấn đề đòi hỏi hiểu quan hệ nguyên nhân – kết quả.
* Hạn chế về **ghi nhớ dài hạn và lập kế hoạch**: chỉ hoạt động trong phạm vi cửa sổ ngữ cảnh và độ dài token.
* **Phụ thuộc dữ liệu huấn luyện**: khả năng tổng quát hóa còn hạn chế ngoài những gì đã “thấy” trong dữ liệu.

How do we make sure that LLMs generate factually correct output? (Làm sao chúng ta có thể chắc rằng LLMs tạo ra kết quả chính xác về mặt thực tế?)

Việc đảm bảo LLM **sinh ra thông tin chính xác** là một bài toán hệ thống — cần kết hợp thiết kế mô hình, nguồn dữ liệu ngoài, lớp kiểm chứng và quy trình vận hành. Dưới đây là giải thích cụ thể các phương pháp và cách áp dụng trong thực tế.

**1) Nguyên tắc: *Có nền tảng (grounding) — đừng chỉ tin vào trọng số***

Trọng số của LLM lưu trữ mẫu ngôn ngữ từ dữ liệu huấn luyện; chúng có thể lỗi thời hoặc sai. Để có câu trả lời đúng, phải **gắn (ground) phản hồi vào nguồn dữ liệu có thể kiểm chứng** (tài liệu, API, cơ sở dữ liệu, máy tính) và trả lại chứng cứ (provenance) cho từng tuyên bố.

**2) Các phương pháp kỹ thuật chính**

**A. Retrieval-Augmented Generation (RAG)**

* **Là gì:** Trước khi sinh, truy xuất các đoạn văn bản liên quan từ chỉ mục bên ngoài và cung cấp chúng làm ngữ cảnh cho mô hình.
* **Cách làm (thực tế):**
  1. Mã hóa truy vấn thành embedding.
  2. Tìm kiếm trong index vector (FAISS, Milvus) hoặc kết hợp BM25 + dense để lấy top-k đoạn liên quan.
  3. Rerank nếu cần (cross-encoder).
  4. Dùng template: Context: [đoạn1]...[đoạnK]\nQ: [câu hỏi]\nA: rồi để mô hình trả lời.
  5. Yêu cầu mô hình trích dẫn đoạn nào hỗ trợ từng tuyên bố.
* **Lợi ích:** Mô hình có thể trích dẫn nguồn, giảm hallucination.
* **Lưu ý:** Truy xuất kém → kết quả sai; cần curate chỉ mục.

**B. Sử dụng công cụ & thực thi ngoài**

* **Là gì:** Cho phép LLM gọi công cụ để lấy thông tin trực tiếp hoặc thực hiện tính toán đáng tin cậy.
* **Ví dụ:** API tìm kiếm web (tin tức cập nhật), truy vấn Graph/DB (Wikidata), máy tính cho số học, sandbox chạy code/SQL.
* **Quy trình:** Mô hình đề xuất call; hệ thống thực hiện; trả kết quả cho mô hình để hoàn thiện câu trả lời.

**C. Mô hình kiểm chứng (verifier)**

* **Là gì:** Chạy mô hình thứ hai hoặc pipeline để kiểm tra từng tuyên bố của generator.
* **Cách:** Chuyển tuyên bố thành truy vấn kiểm chứng; truy xuất bằng retriever; dùng mô hình entailment/QAt để đánh giá *Supported / Contradicted / Not supported*.

**D. Tự kiểm tra & tự phê bình (self-verification)**

* **Là gì:** Yêu cầu mô hình tạo chuỗi suy luận rồi tự kiểm tra lại từng bước với nguồn.
* **Chiến lược:**
  + *Self-consistency*: tạo nhiều chain-of-thought và lấy kết quả phổ biến.
  + *Ask-to-check*: “Liệt kê từng tuyên bố, trích nguồn; nếu không, nói I don’t know.”

**E. Giới hạn giải mã và prompt an toàn**

* **Ý:** Viết prompt bắt buộc nêu nguồn, hoặc bắt buộc từ chối nếu không xác minh được. Dùng decoding hạn chế (giảm sampling sáng tạo) để tránh sinh thêm thông tin tưởng tượng.

**F. Kết hợp mô-đun tượng trưng / xác định**

* **Ít lỗi hơn cho toán học/logic:** Đưa phần toán cho solver deterministic (calculator, CAS), không để mô hình đoán số thủ công.

**G. Fine-tuning và RLHF cho factuality**

* **Làm:** Fine-tune mô hình trên dữ liệu Q/A có nguồn; train reward model phạt hallucination; dùng RL để tối ưu.
* **Giới hạn:** Giảm hallucination nhưng không xóa hoàn toàn, có thể làm mô hình thận trọng quá mức.

**3) Đánh giá & giám sát**

* **Chỉ số tự động:** benchmark factuality (kiểu FEVER), kiểm tra nhất quán bằng QA/entailment.
* **Human eval:** chuyên gia đánh đánh giá chính xác.
* **Giám sát sản xuất:** log output + nguồn, chạy canary/kiểm thử tấn công, thu feedback người dùng.
* **Regression tests:** tập câu hỏi chuẩn có đáp án để phát hiện suy giảm.

**4) Giao diện sản phẩm & thực hành giảm tác hại**

* Luôn hiển thị **nguồn** (snippet + URL).
* Hiển thị **mức độ tin cậy** hoặc cho phép I don’t know.
* Cơ chế **feedback** cho người dùng để sửa lỗi.
* Ở tác vụ quan trọng (y tế, pháp lý) bắt buộc **con người kiểm duyệt**.

**5) Các lỗi thường gặp & cách khắc phục**

* **Dữ liệu lỗi thời:** cập nhật index thường xuyên, dùng API trực tiếp cho tin mới.
* **Retriever trả docs sai:** dùng reranker, lọc chất lượng nguồn.
* **Mô hình hiểu sai ngữ cảnh:** yêu cầu quote verbatim rồi chạy entailment.
* **Tự tin quá mức:** ép abstention, hiệu chỉnh confidence.
* **Prompt tấn công:** guardrails, lọc input, monitoring.

**6) Checklist triển khai (cho kĩ sư)**

1. Tạo index tài liệu đáng tin (kèm timestamp).
2. Áp dụng RAG: retrieve → rank → include → generate.
3. Thêm tool calls cho dữ liệu trực tuyến và tính toán.
4. Kiểm chứng từng tuyên bố bằng verifier.
5. Bắt buộc trích dẫn trong output và hiển thị provenance.
6. Hiệu chỉnh confidence, bắt từ chối khi không xác minh được.
7. Kiểm thử liên tục: tự động + người đánh giá.
8. Lưu log và vòng phản hồi để cập nhật chỉ mục / tinh chỉnh.

**7) Giới hạn & đánh đổi**

* Grounding tăng tính chính xác nhưng **tăng chi phí, độ trễ**.
* Dựa vào nguồn ngoài có thể đưa vào **thông tin sai** nếu nguồn không đáng tin.
* Không có hệ thống nào hoàn hảo — ở các bài toán nhiều bước phức tạp hoặc thiếu chứng cứ, mô hình vẫn có nguy cơ bịa (hallucinate). Con người vẫn cần tham gia ở bước kiểm duyệt khi rủi ro cao.

How do we fairly compensate the people who create the data that is used to train LLMs? (Làm thế nào để chúng ta trả công xứng đáng cho những người tạo ra dữ liệu được sử dụng để đào tạo LLM?)

Việc **đền bù công bằng cho những người tạo ra dữ liệu** dùng để huấn luyện LLMs là một thách thức phức tạp cả về đạo đức lẫn kinh tế. Dữ liệu thường đến từ nhiều nguồn khác nhau — sách, báo, trang web, mạng xã hội, diễn đàn,... Những cá nhân tạo ra nội dung này ban đầu không được trả công với mục đích phục vụ cho việc huấn luyện AI. Để giải quyết vấn đề này, có một số hướng tiếp cận:

1. **Cấp phép và hợp tác**: Các công ty có thể ký thỏa thuận với nhà xuất bản, nền tảng hoặc tác giả để sử dụng dữ liệu, đảm bảo người sáng tạo được trả công khi tác phẩm của họ đóng góp cho việc huấn luyện AI.
2. **Mô hình chia sẻ doanh thu**: Tương tự như cách nhạc sĩ được trả tiền qua nền tảng streaming, người sáng tạo có thể nhận một phần doanh thu do các hệ thống AI tạo ra từ dữ liệu huấn luyện.
3. **Cơ chế tham gia/từ chối (opt-in/opt-out)**: Cho phép người sáng tạo lựa chọn có cho phép dữ liệu của họ được sử dụng hay không, kèm theo cơ chế trả công nếu họ đồng ý.
4. **Tài trợ công và dữ liệu mở**: Với dữ liệu có nguồn gốc từ quỹ công (ví dụ: nghiên cứu học thuật), hình thức bù đắp có thể là gián tiếp, thông qua việc tăng đầu tư cho các viện nghiên cứu và hạ tầng tri thức.
5. **Ghi nhận và minh bạch**: Dù việc truy ngược chính xác dữ liệu nào dẫn đến đầu ra nào là khó, các hệ thống vẫn có thể cung cấp sự minh bạch về nguồn dữ liệu và ghi nhận đóng góp của người sáng tạo.

How do we use LLMs in learning, so human learning is not compromised? (Chúng ta sử dụng LLM trong học tập như thế nào để việc học tập của con người không bị ảnh hưởng?)

Việc sử dụng LLMs trong học tập mà **không làm tổn hại đến quá trình học của con người** đòi hỏi sự cân bằng giữa **hỗ trợ** và **không phụ thuộc**. LLMs có thể là công cụ mạnh mẽ để nâng cao giáo dục, nhưng nếu dùng sai cách, chúng có thể khiến người học mất đi tư duy phản biện và khả năng giải quyết vấn đề. Một số cách tiếp cận chính:

1. **Hỗ trợ, không thay thế**: LLMs nên được dùng để giải thích khái niệm, đưa ví dụ hoặc phản hồi, chứ không phải để cung cấp luôn đáp án cuối cùng. Điều này giúp người học vẫn phải tự xử lý kiến thức.
2. **Khuyến khích tư duy phản biện**: Người học nên được hướng dẫn đặt câu hỏi, kiểm chứng, và mở rộng từ thông tin LLM đưa ra. Ví dụ: so sánh câu trả lời của LLM với sách giáo khoa hoặc tự đánh giá độ chính xác.
3. **Học tập thích ứng**: LLMs có thể cá nhân hóa trải nghiệm học, đưa ra lời giải thích phù hợp với trình độ từng người. Điều này hỗ trợ hiểu sâu hơn nhưng vẫn đòi hỏi sự tham gia chủ động.
4. **Minh bạch và định hướng**: Giáo viên cần giải thích rõ cho học sinh về điểm mạnh và hạn chế của LLMs, từ đó dạy cách sử dụng có trách nhiệm, tránh lệ thuộc.
5. **Học tập chủ động**: Sử dụng LLMs để tạo bài tập tương tác — như đóng vai hội thoại, giải bài tập, hoặc mô phỏng tình huống — để kích thích sự tham gia thay vì tiếp nhận thụ động.

Tóm lại, LLMs nên đóng vai trò **người hướng dẫn và cộng tác viên**, chứ không phải công cụ thay thế nỗ lực học tập của con người. Khi được tích hợp hợp lý, chúng có thể **tăng động lực, mở rộng khả năng tiếp cận tri thức và nuôi dưỡng sự tò mò**, đồng thời vẫn giữ được cốt lõi của việc học: **nỗ lực, suy ngẫm và tư duy**.