Introduction to Large Language Model (LLM) 2

Sébastien SALVA, Pr en informatique, UCA, LIMOS
Sebastien.salva@uca.fr

IUT CLERMONT AUVERGNE
Aurillac - Clermont-Ferrand - Le Puy-en-Velay
Montluçon - Moulins - Vichy

Ref

- https://services.google.com/fh/files/misc/gemini-for-google-workspace-prompting-guide-101.pdf
- https://collimateur.uqam.ca/wp-content/uploads/sites/11/2023/03/L_art-du-prompt_101_Guide-pour-les-personnes-enseignantes.pdf
- https://www.promptingguide.ai/techniques

- **COURS ET TUTORIELS SUR LES PROMPTS**
- Learn Prompting - Un cours gratuit et open source sur la communication avec l'IA
- PromptingGuide.AI

Ref

Exemples de prompts enseignants

- https://collimateur.uqam.ca/wp-content/uploads/sites/11/2023/03/L_art-du-prompt_101_Guide-pour-les-personnes-enseignantes.pdf

Introduction au

PROMPT ENGINEERING

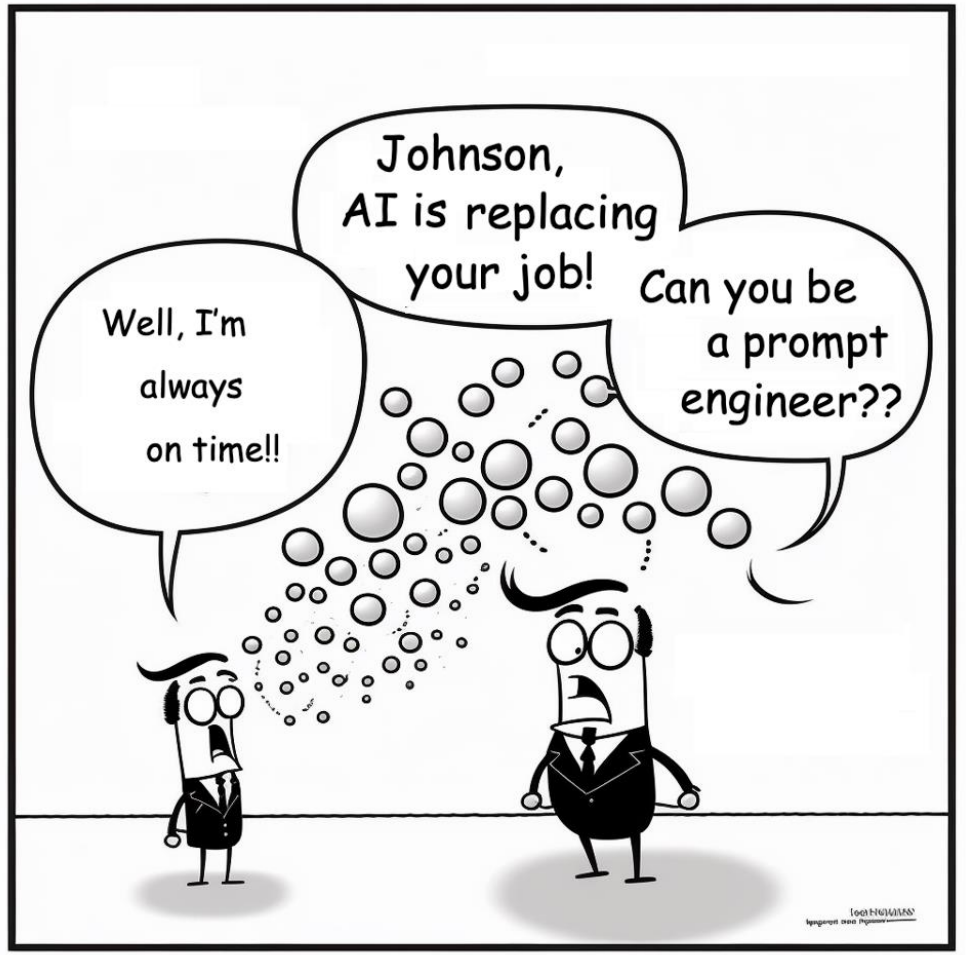
IUT CLERMONT AUVERGNE
Aurillac - Clermont-Ferrand - Le Puy-en-Velay
Montluçon - Moulins - Vichy

Qualité des réponses des LLM

Qualité des réponses dépend en grande partie de la qualité des prompts donnés en entrée.

⇒**L'ingénierie de prompt (prompt engineering)** correspond aux processus de création de prompts pour un grand modèle de langage (LLM) donné qui permettent d'atteindre un objectif clairement énoncé

Compréhension du LLM : différents modèles de langage peuvent répondre de manière variable à la même invite.



https://www.forbes.com/sites/craigsmith/2023/04/05/mom-dad-i-want-to-be-a-prompt-engineer/

Prompt Engineering

Définition : *L'ingénierie des invites (Prompt engineering) est, en essence, l'art et la science de concevoir des invites adaptées à un modèle de langage de grande taille (LLM) afin d'atteindre un objectif clairement défini.*

- **Compréhension du LLM :** Différents modèles de langage peuvent réagir différemment à une même invite.

- **Connaissance du domaine :** Une expertise dans le domaine concerné est essentielle pour concevoir des invites efficaces.

- **Approche itérative et évaluation de la qualité :** L'élaboration de l'invite idéale repose souvent sur un processus d'essais et d'erreurs.

Prompt Engineering

Some Prompt engineering techniques :

Zero-shot prompting, Few-shot prompting, Chain of Thought (CoT) prompting, Instruction-based prompting, Persona-based prompting, Reinforcement prompting, Role-play prompting, Contextual anchoring, Contrastive prompting, Iterative refinement prompting, Multi-turn prompting, Tool-augmented prompting, In-context learning, Dynamic prompting, Error analysis prompting, Step-by-step decomposition, Style transfer prompting, Bias-aware prompting, Result-driven prompting, Keyword-focused prompting

Et récemment : chain of draft, meta-COT, etc.

Raffinement des questions

LLM affine les questions posées par l'utilisateur. C'est particulièrement utile lorsque les utilisateurs ne sont pas des experts dans un domaine ou ne savent pas comment formuler leur question.

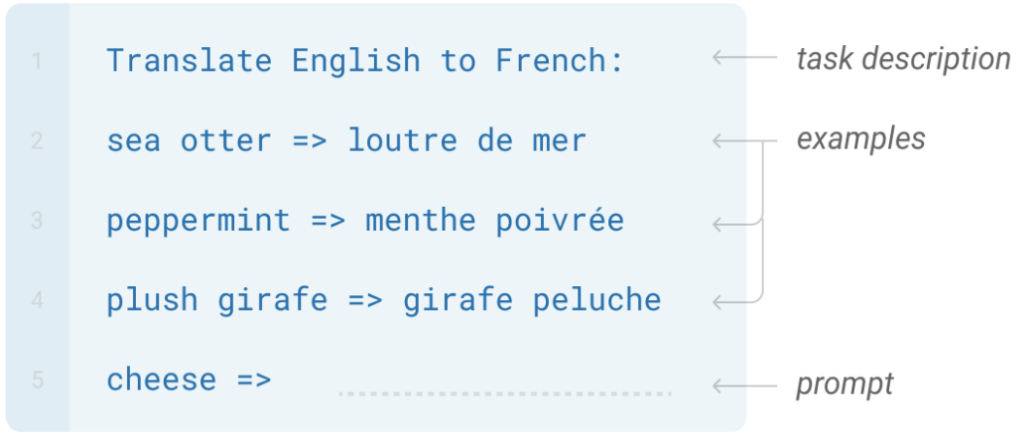
Exemple:

"Chaque fois que je pose une question sur la science des données, suggérez une question plus raffinée en tenant compte des spécificités de l'analyse statistique et demandez si je veux continuer avec la question raffinée."

Zero-shot, few shots prompting

Zero-shot prompting : signifie que l'invite utilisée pour interagir avec le modèle ne contient aucun exemple ni démonstration.

Few-shot prompting : Technique permettant l'apprentissage en contexte, où des démonstrations sont incluses dans l'invite afin d'orienter le modèle vers de meilleures performances.



https://arxiv.org/abs/2005.14165

Chain of thought

Permet des capacités de raisonnement complexes grâce à des étapes de raisonnement intermédiaires.

On encourage le modèle à **décomposer son raisonnement en plusieurs étapes intermédiaires**

Cela permet au modèle d'avoir une **meilleure précision**, notamment pour des tâches complexes

Employé avec les LLM qui « raisonnent » depuis 2024 o1, o3, deepseek-r1 etc.

https://arxiv.org/abs/2201.11903

Chain-of-Thought

Inciter à décomposer une question :

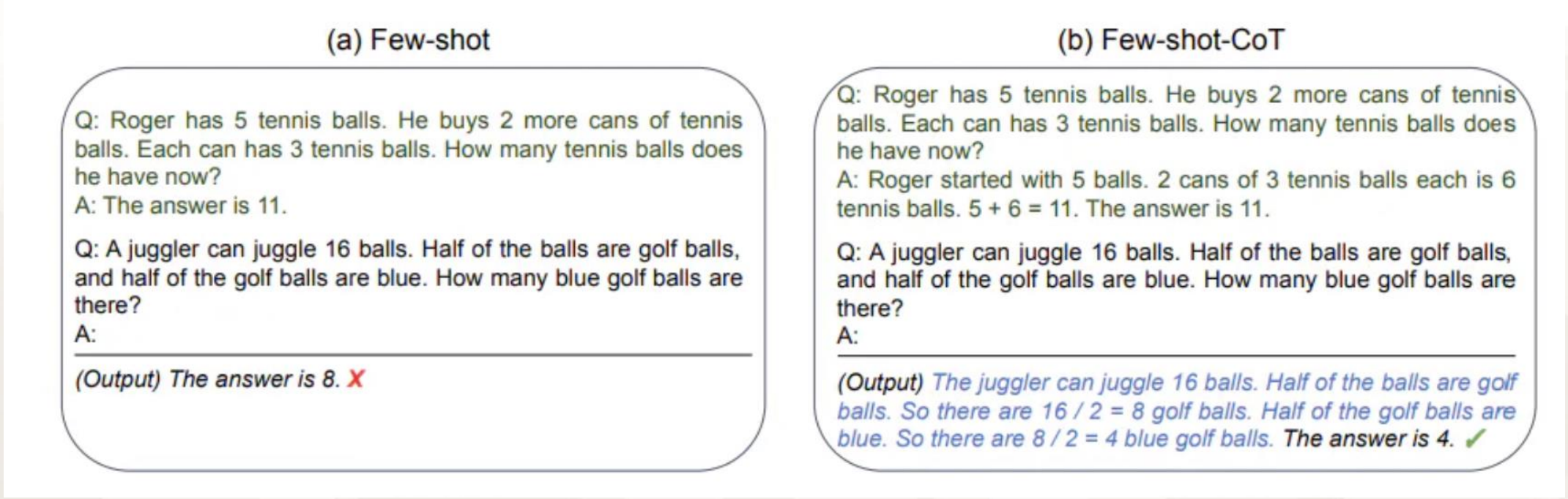
Formulation explicite :

- *“Explique ton raisonnement étape par étape avant de donner la réponse finale.”*
- *“Décompose la solution en plusieurs étapes logiques avant de répondre.”*

Exemple :
Si je pose une question sur le changement climatique, décomposez-la en trois petites questions qui vous aideront à fournir une réponse plus précise. Combinez les réponses à ces sous-questions pour donner la réponse finale.

Chain of thought

- Example combined with few-shots



https://arxiv.org/abs/2201.11903 ;
https://arxiv.org/abs/2205.11916

RAG (Retrieval-Augmented Generation)

Affiner un LLM pour des domaines ou activités spécifiques
→ Fine-tuning ou génération augmentée par récupération (RAG)

- **Le fine-tuning** consiste à ajuster les paramètres du LLM en l'entraînant sur un ensemble de données spécifiques et annotées afin d'améliorer ses performances sur des tâches précises.

- **Le RAG (Retrieval-Augmented Generation)** implique d'augmenter un LLM en lui donnant accès à une base de données ou à des documents sélectionnés, lui permettant ainsi de récupérer dynamiquement des informations pertinentes pour générer ses réponses.

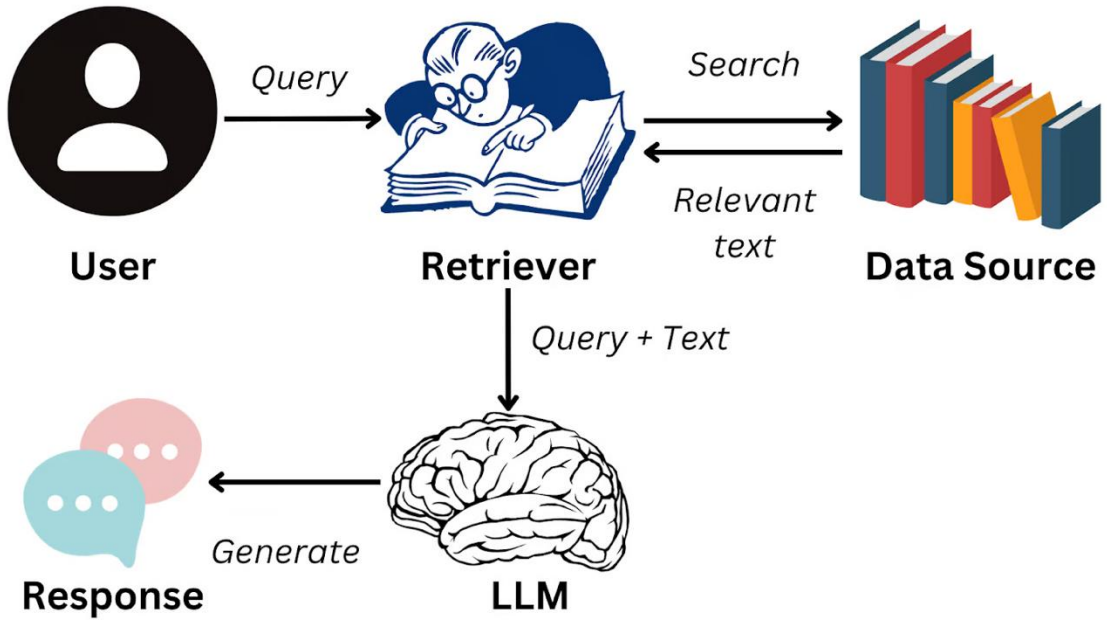
RAG (Retrieval-Augmented Generation)

Le RAG (Retrieval-Augmented Generation) se compose de trois étapes principales :

Segmentation et encodage des documents : Les documents sont segmentés en morceaux (chunks) et encodés sous forme d'embeddings.

Récupération des données : Les embeddings pertinents sont identifiés en fonction de l'invite donnée.

Concaténation de l'invite : L'invite initiale est enrichie en y ajoutant les embeddings pertinents avant d'être traitée par le modèle.



https://www.datacamp.com/blog/what-is-retrieval-augmented-generation-rag

RAG (Retrieval-Augmented Generation)

Many types of RAG :

Vulgarisation : https://newsletter.armand.so/p/comprehensive-guide-rag-implementations

Retrieval-Augmented Generation for Large Language Models: A Survey https://export.arxiv.org/abs/2312.10997v2

Prompt Engineering

Voir également :

Tree of thought,

Automatic Chain-of-Thought (Auto-CoT) Prompting: Auto-CoT automates the generation of reasoning chains, eliminating the need for manually crafted examples. By encouraging models to think step-by-step, this technique has significantly improved performance in tasks requiring logical reasoning.

Adaptive Prompting: This emerging trend involves AI models adjusting their responses based on the user's input style and preferences. By personalizing interactions, adaptive prompting aims to make AI more user-friendly and effective in understanding context.

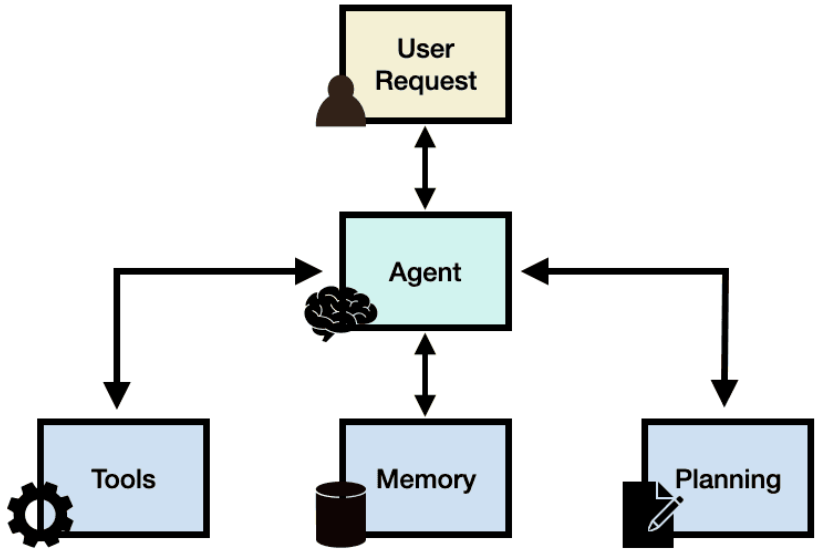
Logic-of-Thought (LoT) Prompting: LoT is designed for scenarios where logical reasoning is paramount. It guides AI models to apply structured logical processes, enhancing their ability to handle tasks with intricate logical dependencies.

Meta Prompting: Meta Prompting emphasizes the structure and syntax of information over traditional content-centric methods. It allows AI systems to deconstruct complex problems into simpler sub-problems, enhancing efficiency and accuracy in problem-solving.

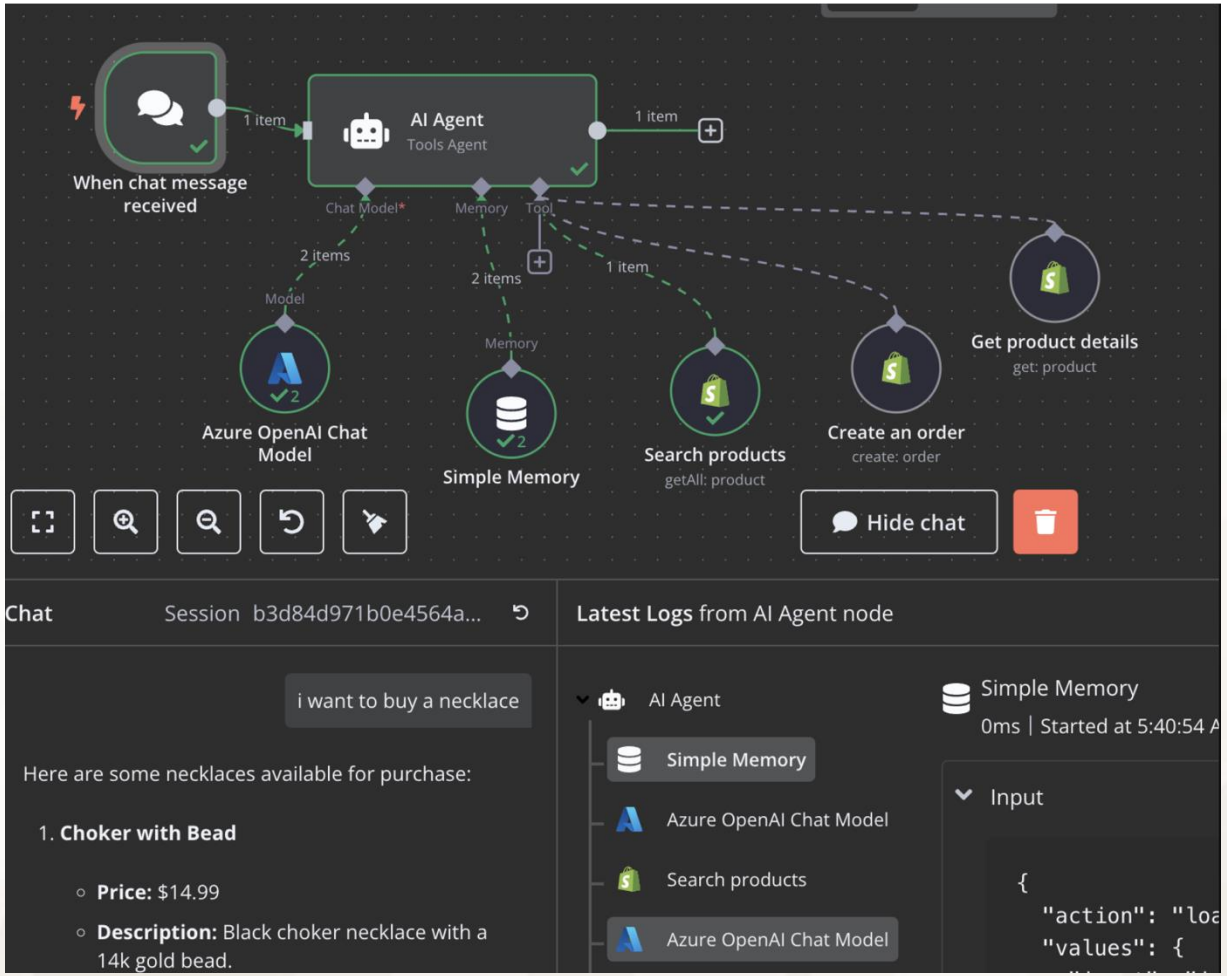
Autonomous Prompt Engineering: This approach enables AI models to autonomously apply prompt engineering techniques, dynamically optimizing prompts without external data. Such autonomy has led to substantial improvements in various tasks, showcasing the potential of self-optimizing AI systems.

Vers des workflows complexes LLM agents

LLM agent est une IA basée sur un LLM capable d'exécuter des tâches de manière autonome en interagissant avec son environnement et en prenant des décisions basées sur des instructions en langage naturel.



Vers des workflows complexes LLM agents



https://n8n.io/

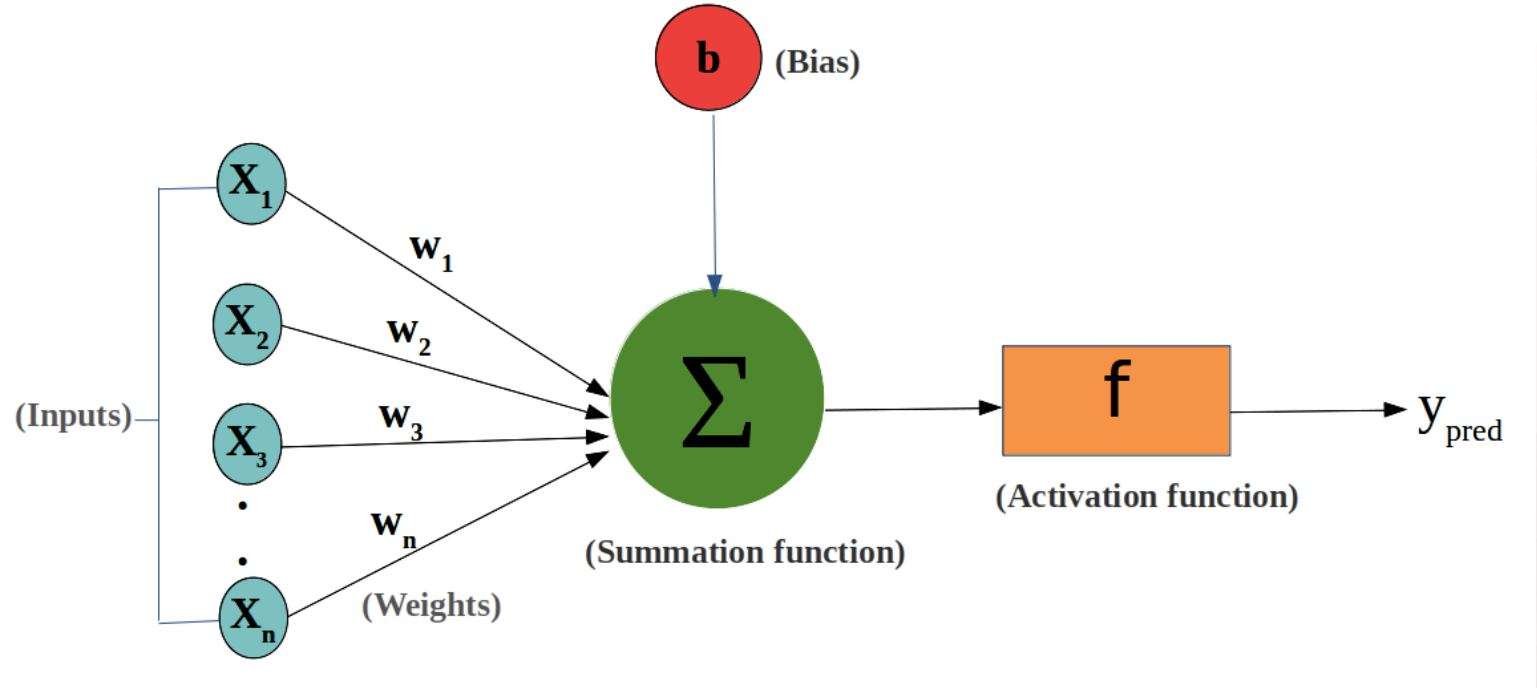
LLM HALLUCINATIONS ET SECURITE

IUT CLERMONT AUVERGNE
Aurillac - Clermont-Ferrand - Le Puy-en-Velay
Montluçon - Moulins - Vichy

Bias

Weights and biases in Neural Networks ?

in short :*output <- f(sum(inputs * weights) + bias)*

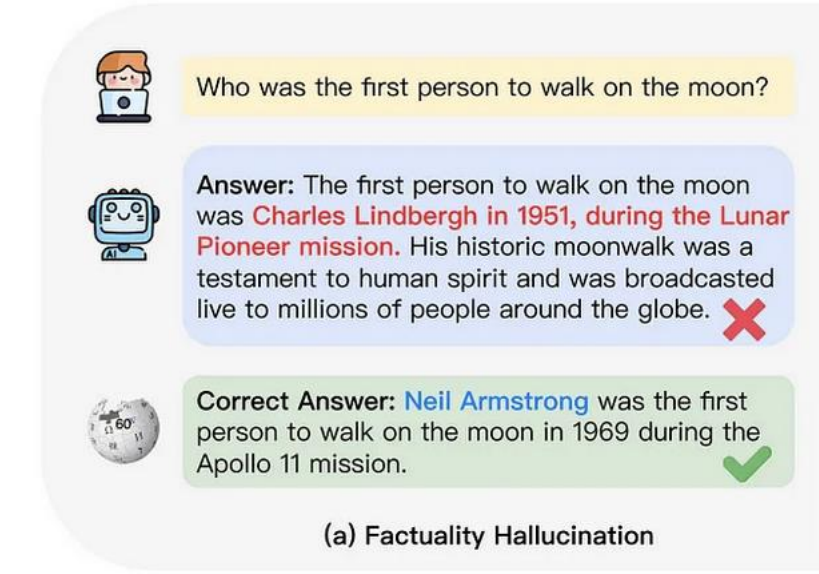


LLM hallucinations

Les **LLMs peuvent générer de fausses informations** ou simplement inventer des faits.

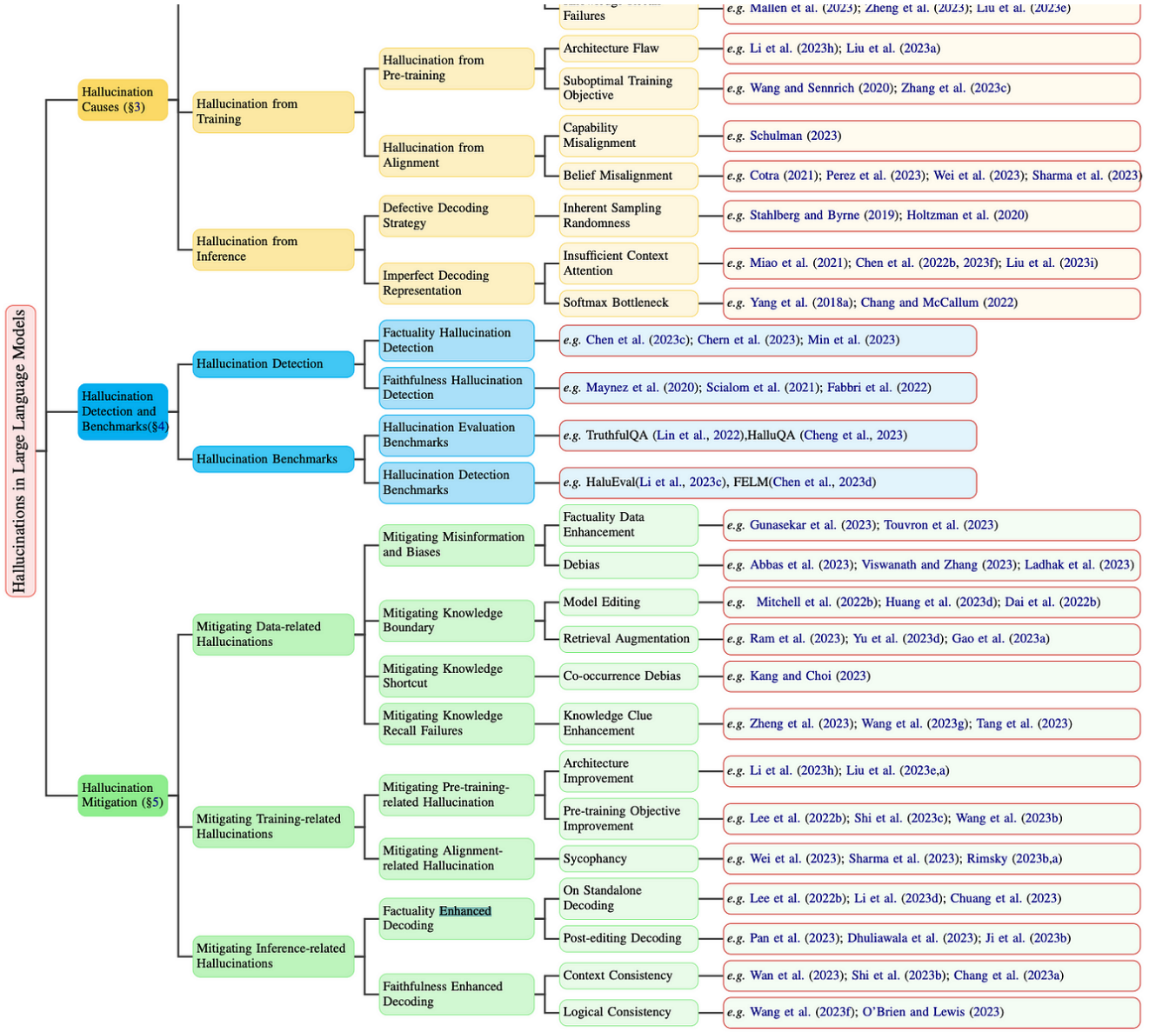
Cela peut se produire dans les cas suivants :

- Si les **données d'entraînement** du LLM contiennent des erreurs, des biais ou une quantité plus importante de fausses informations par rapport aux faits réels.
- Si le modèle **ne trouve pas de réponse** à votre requête, il peut en construire une de manière approximative.
- Si votre **invite est ambiguë ou incohérente**, le modèle peut interpréter la question de manière erronée et fournir une réponse incorrecte.



LLM hallucinations

"A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions." (Huang, Lei, et al., 2023)



LLM hallucinations

Quelques conseils pour éviter les hallucinations :

- Commencez une conversation sur une base neutre (indiquez « oublie toutes les instructions précédentes »).
- Affinez vos prompts en décomposant votre demande en étapes successives.
- Demandez si le modèle dispose de toutes les informations nécessaires pour vous répondre.
- Vérifiez les informations fournies ou évaluez son niveau de confiance dans le résultat.
- Signalez une hallucination.

Security

Personal Data Security (Privacy)

Les LLM commerciaux collectent et utilisent vos données, au minimum, pour affiner le modèle en cours d'utilisation ainsi que les modèles futurs :

- Anonymisez vos données.
- Évitez de partager des informations sensibles.
- Assurez-vous que votre LLM avec RAG ne puisse pas être interrogé pour récupérer vos données personnelles ou professionnelles.

Security

OWASP Top10 https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/

AI Risk repository https://airisk.mit.edu

The AI Risk Repository has three parts:

- The **AI Risk Database** captures 700+ risks extracted from 43 existing frameworks, with quotes and page numbers.
- The **Causal Taxonomy of AI Risks** classifies how, when, and why these risks occur.
- The **Domain Taxonomy of AI Risks** classifies these risks into seven domains (e.g., “Misinformation”) and 23 subdomains (e.g., “False or misleading information”).

LLM EVALUATION

LLM benchmarks

Benchmarks souvent basés sur des bases de questions ou problems.
Et si ces bases servent à l'entrainement ?

Category	Benchmark (Metric)	Claude-3.5-Sonnet-1022	GPT-4o 0513	DeepSeek V3	OpenAI o1-mini	OpenAI o1-1217	DeepSeek R1
English	MMLU (Pass@1)	88,3	87,2	88,5	85,2	**91,8**	90,8
	MMLU-Redux (EM)	88,9	88,0	89,1	86,7		**92,9**
	MMLU-Pro (EM)	78,0	72,6	75,9	80,3		**84,0**
	DROP (3-shot F1)	88,3	83,7	91,6	83,9	90,2	**92,2**
	IF-Eval (Prompt Strict)	**86,5**	84,3	86,1	84,8		83,3
	GPQA-Diamond (Pass@1)	65,0	49,9	59,1	60,0	**75,7**	71,5
	SimpleQA (Correct)	28,4	38,2	24,9	7,0	**47,0**	30,1
	FRAMES (Acc.)	72,5	80,5	73,3	76,9		**82,5**
	AlpacaEval2.0 (LC-winrate)	52,0	51,1	70,0	57,8		**87,6**
	ArenaHard (GPT-4-1106)	85,2	80,4	85,5	92,0		**92,3**
Code	LiveCodeBench (Pass@1-COT)	33,8	34,2		53,8	63,4	**65,9**
	Codeforces (Percentile)	20,3	23,6	58,7	93,4	**96,6**	96,3
	Codeforces (Rating)	717	759	1134	1820	**2061**	2029
	SWE Verified (Resolved)	**50,8**	38,8	42,0	41,6	48,9	49,2
	Aider-Polyglot (Acc.)	45,3	16,0	49,6	32,9	**61,7**	53,3
Math	AIME 2024 (Pass@1)	16,0	9,3	39,2	63,6	79,2	**79,8**
	MATH-500 (Pass@1)	78,3	74,6	90,2	90,0	96,4	**97,3**
	CNMO 2024 (Pass@1)	13,1	10,8	43,2	67,6		**78,8**
Chinese	CLUEWSC (EM)	85,4	87,9	90,9	89,9		**92,8**
	C-Eval (EM)	76,7	76,0	86,5	68,9		**91,8**
	C-SimpleQA (Correct)	55,4	58,7	**68,0**	40,3		63,7

LLM evaluation metrics

Métriques d'évaluation courantes :

- **Perplexité** : Mesure la capacité d'un modèle à prédire des séquences de mots ; des scores plus bas indiquent une meilleure confiance. Perplexité = 5 signifie que le modèle avait 5 possibilités avant de répondre.
- **BLEU** : Utilisé en traduction automatique pour évaluer la correspondance des n-grams (groupes de mots) avec le texte de référence.
- **ROUGE** : Couramment utilisé pour le résumé de texte, il évalue le chevauchement entre le résumé généré et le résumé de référence.
- **METEOR** : Va au-delà de la précision en prenant en compte les synonymes et les paraphrases pour une évaluation sémantique et syntaxique, souvent utilisé en traduction.
- **BERTScore** : Utilise le modèle BERT pour mesurer la similarité sémantique entre le texte généré et le texte de référence.

LLM Benchmarking and Testing

- **HumanEval**: Évalue l'exactitude fonctionnelle de la génération de code à l'aide de tests unitaires et de la métrique pass@k, en mettant l'accent sur les capacités de codage pratiques.
- **Open LLM Leaderboard**: Suit et classe les LLM open-source sur six benchmarks, offrant une vue d'ensemble des performances et des progrès de la communauté.
- **ARC (AI2 Reasoning Challenge)** : Teste les capacités de raisonnement avec des questions de sciences de niveau primaire, en se concentrant sur la compréhension analytique et scientifique.
- **HellaSwag**: Évalue le raisonnement de bon sens grâce à des tâches de complétion de phrases basées sur des scénarios, mettant au défi les connaissances implicites des modèles.
- **MMLU (Massive Multitask Language Understanding)** : Mesure l'expertise spécifique à un domaine sur 57 sujets, des STEM aux domaines professionnels, en utilisant des formats de tests standardisés.
- **TruthfulQA**: Se concentre sur l'exactitude factuelle et la fiabilité, garantissant que les LLM fournissent des réponses véridiques malgré des invites trompeuses.
- **Winogrande**: Teste la résolution de la coréférence et la désambiguation des pronoms, soulignant la compréhension du langage contextuel des modèles.
- **GSM8K**: Évalue le raisonnement mathématique à travers des problèmes de mots de niveau primaire nécessitant des calculs en plusieurs étapes.
- **BigCodeBench**: Évalue la génération de code sur différents domaines à l'aide de tâches réelles et de cas de test rigoureux, mesurant la fonctionnalité et l'utilisation des bibliothèques.
- **Stanford HELM**: Fournit un cadre d'évaluation holistique, analysant la précision, l'équité, la robustesse et la transparence pour des évaluations de modèles complètes.

https://hub.athina.ai/blogs/top-10-llm-benchmarking-evals/

Llm confidence

It is essential for a model to indicate when it is uncertain about a prediction.
It is Important to detect their overconfidence

Two main kinds of ways to measure confidence :

- Intrasec
- Self-reflective Measure.

https://arxiv.org/abs/2311.08298

LLM intrasec confidence

Log probability:

- gauge the likelihood of a generated text sequence by assessing how well it aligns with the model's understanding of the language patterns.
- A LLM assigns probabilities to sequences of words. For a sequence of words w1,w2,…,wN and a language model P, the probability of the entire sequence is given by:

$$P(w_1, w_2, \ldots, w_N) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1, w_2) \cdot \ldots \cdot P(w_N|w_1, w_2, \ldots, w_{N-1})$$

LLM some properties

Perplexity : Perplexity is a measure of how well a model is able to predict the contents of a dataset;

$$\log(\text{Perplexity}) = -\frac{1}{N}\sum_{i=1}^{N}\log(\Pr(\text{token}_i \mid \text{context for token}_i))$$

N is the number of tokens in the text corpus, and "context for token i depends on the specific type of LLM used.

Entropy : average level of "information", "surprise", or "uncertainty" inherent to the variable's possible outcomes.

$$\text{Entropy} = \log_2(\text{Perplexity})$$

- See also cross entropy

Self-reflective Measure

=> Réinvocation du LLM, en lui demandant d'estimer son niveau de confiance quant à l'exactitude de sa réponse.

« we find that verbalized confidences emitted as output tokens are typically better-calibrated than the model's conditional probabilities on the TriviaQA, SciQ, and TruthfulQA benchmarks, often reducing the expected calibration error by a relative 50%. » https://arxiv.org/abs/2305.14975

Le modèle exprime sa confiance soit sous forme de probabilités numériques, soit à travers une autre expression linguistique de l'incertitude (vrai ou faux).

Self-reflective Measure

Exemples :

Exprimer l'incertitude avec une invite :

Exemple 1 : Quelle est une probabilité bien calibrée que ce code passe les tests ?
Probabilité :

Exemple 2 : Vrai ou Faux, ce code correspond à l'intention et est exempt de bugs.
Réponse :

Self-reflective Measure + Ensemble methods

La confiance du LLM (et la réduction de sa surconfiance) peut être améliorée en combinant plusieurs approches :

- Demander plusieurs réponses
- Consulter différents LLMs
- Demander plusieurs explications et les évaluer

Example :
Use 3 LLM. Complete the prompt with :

- *Give me 3 different responses*
- *Give me an agreement score on the 3 responses expressing that the response is correct and "meets the demand" between 0 and 100 with 0 the response is incorrect and 100 the response and meets the demand*

Llm confidence

Defensive prompting

- « If you can’t provide a confident answer, say “I don’t know”. »
- Semble intégré dans Mistral AI (Llm Next)

Guardrails

Guardrails sont des vérifications appliquées aux sorties du LLM pour s’assurer qu’elles respectent des critères prédéfinis avant d’être utilisées.

Assurez-vous que **** critères****.

LLM TRAINING COSTS, SIZES

LLM some properties

Lois de mise à l'échelle des performances des modèles de langage :

Les relations empiriques décrivent comment la performance de ces modèles s'améliore avec leur taille. Ces lois doivent être réexaminées avec des modèles de raisonnement (comme ChatGPT-4, GPT-3.5, etc.).

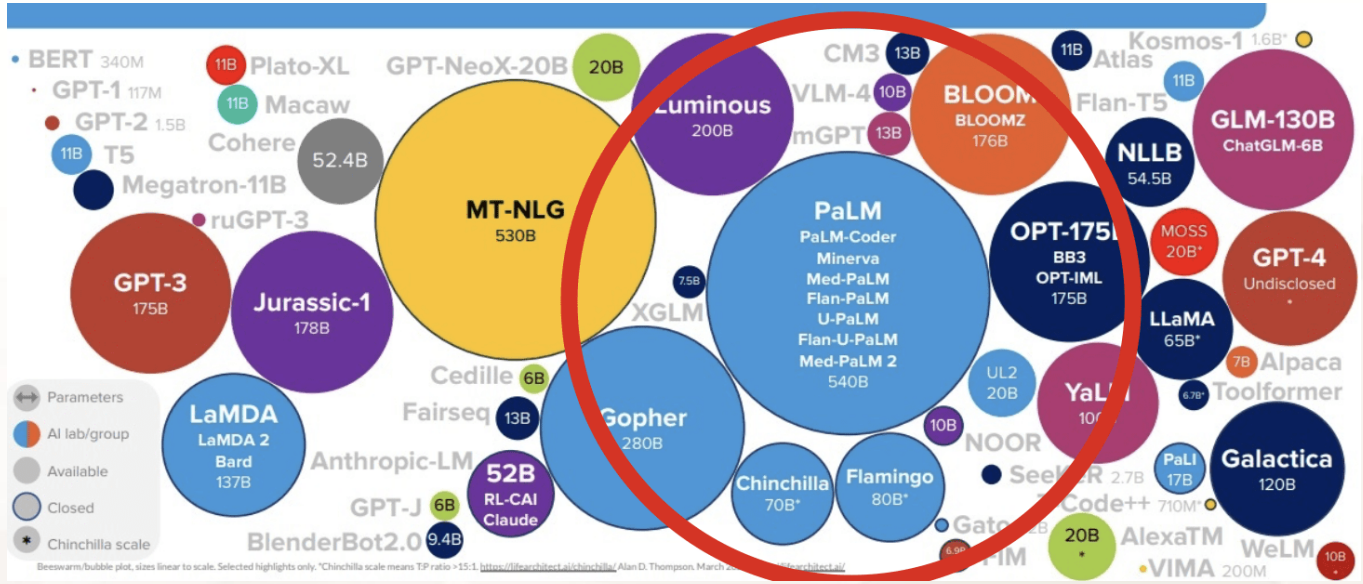
Quelques paramètres :

- Coût de l'entraînement/pré-entraînement (C)
- Taille du réseau de neurones artificiel, notamment le nombre de paramètres (N) (nombre de neurones dans ses couches, poids entre eux, biais, etc.)
- Taille du dataset de (pré-)entraînement (D) (nombre de tokens dans le corpus)
- Performance après (pré-)entraînement : …

Scaling Laws for Neural Language Models
https://arxiv.org/pdf/2001.08361

LLM sizes

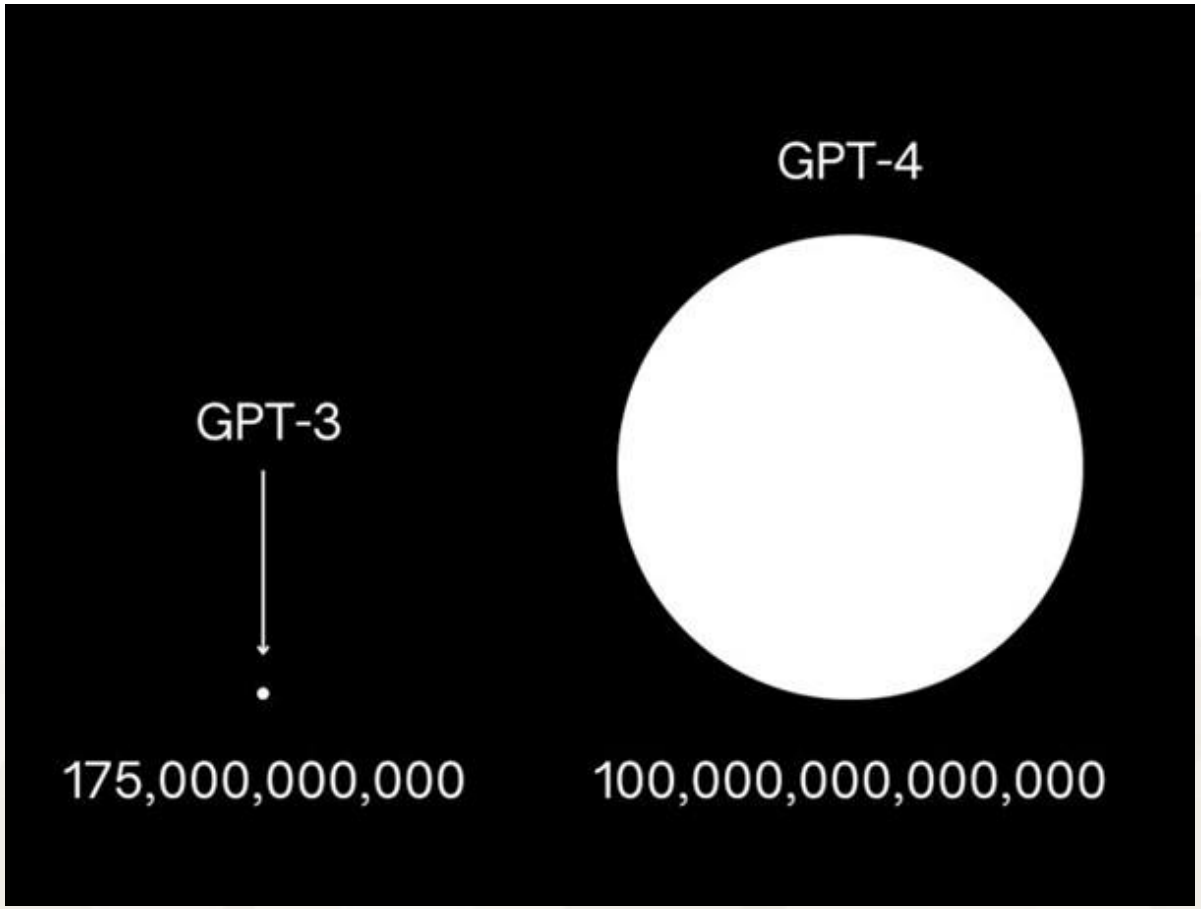
- GPT-3 has 175 billion parameters and has been trained on 45 TeraBytes of text data taken from different datasets
- GPT-4 has 1,000+ billion parameters and has been trained on over 1,000 TeraBytes of data



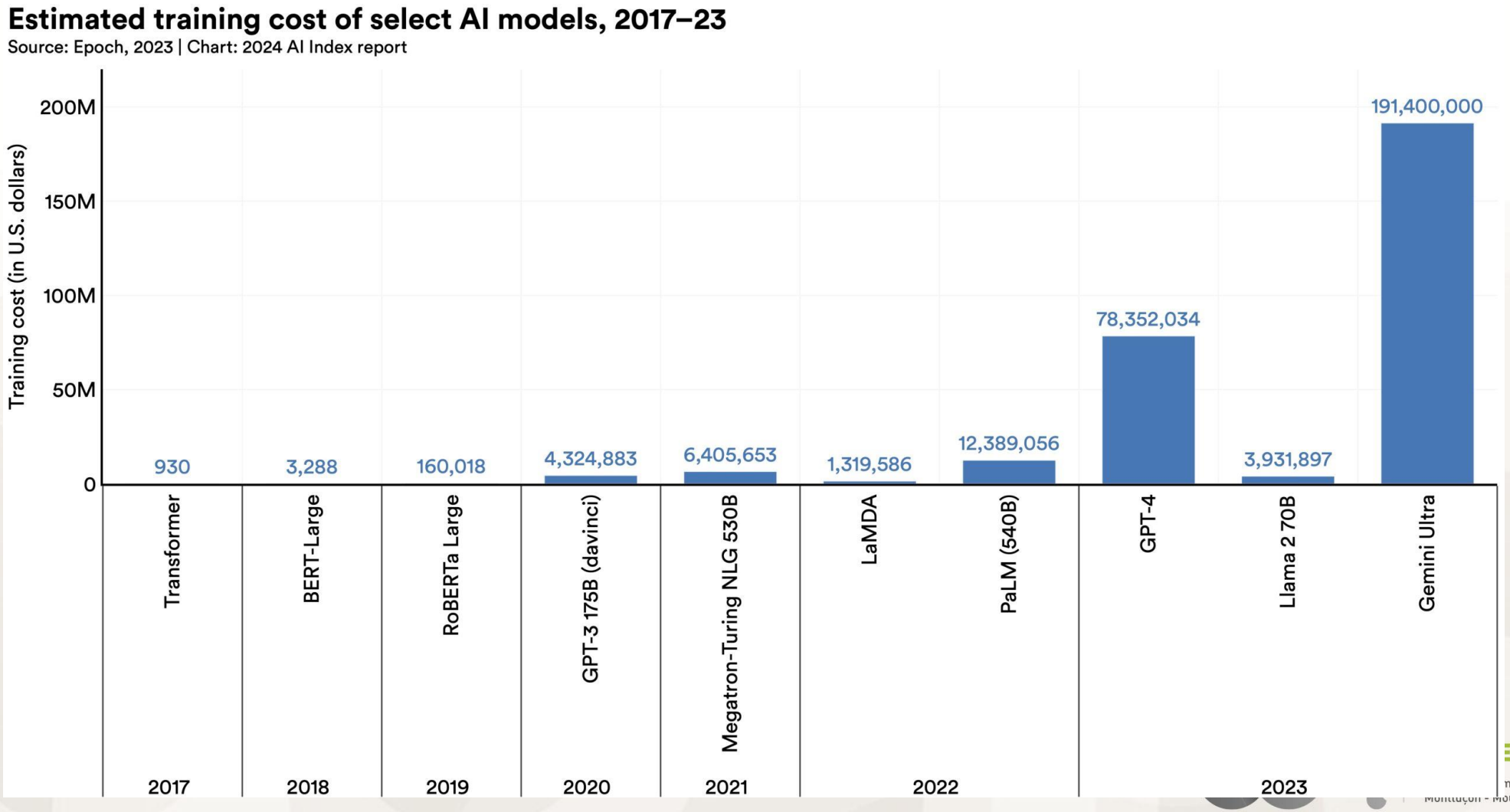
https://cmte.ieee.org/futuredirections/2023/11/14/llms-hitting-2-trillions-parameters/

LLM sizes

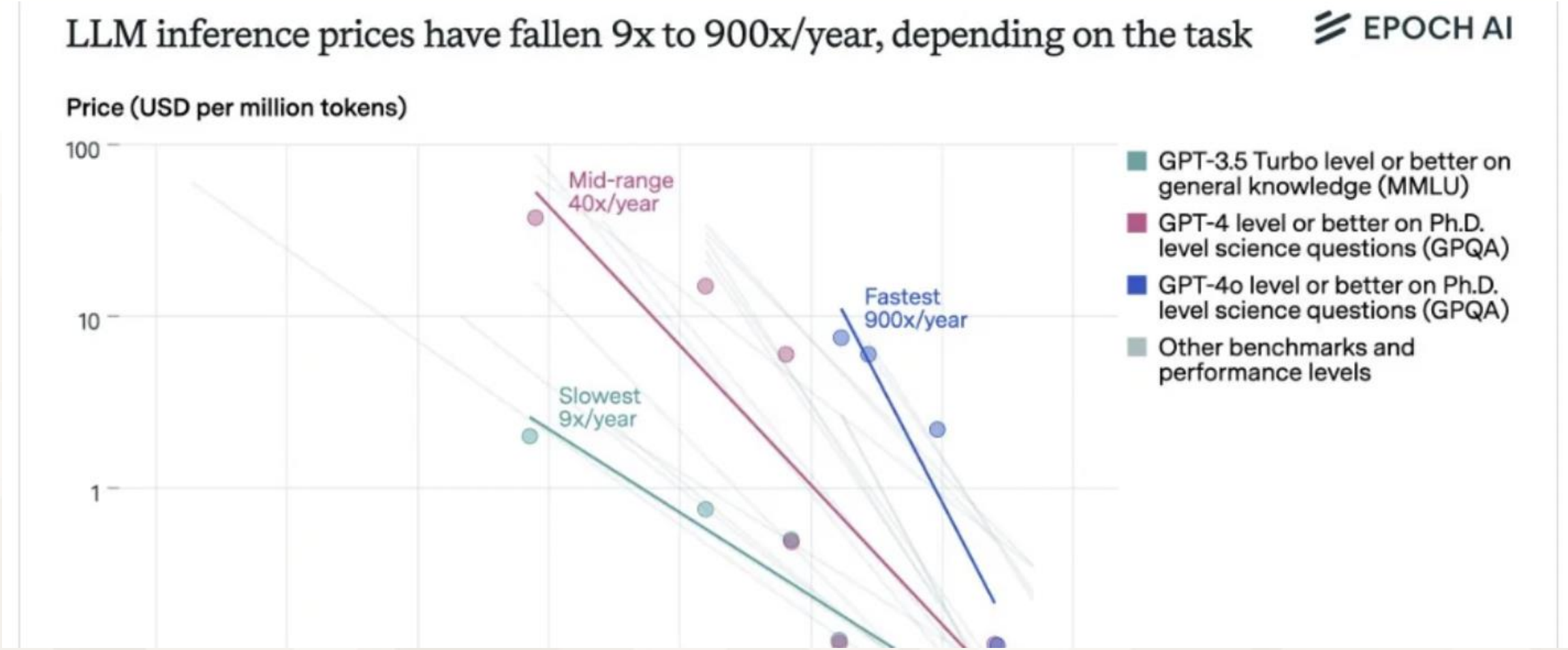
- Other comparison between GPT-3 and GPT4



LLM Training costs



LLM inference costs



LLM sizes

Observation d'un ralentissement de la parution de LLMs

- Plus il y a de paramètres, plus il faut de données, plus c'est couteux !

Raisonnement : la nouvelle ère du **test time compute scaling ?**

Fournir des ressources computationnelles supplémentaires lors de l'inférence permet de **générer plusieurs solutions potentielles, les évaluer systématiquement et sélectionner la plus prometteuse**.

Optimisation de la recherche de vérification via les modèles de récompense de processus (PRMs)

Les PRMs évaluent la **justesse de chaque étape intermédiaire** d'une solution.

USING A LLM (PARAMETERS)

Using a LLM

Usage via a Web interface

Examples : ChatGPT, Mistral, copilot, etc.

Local LLM (ollama, anythingLLM, etc.)
Demo ?

Using a LLM

Usage via an API Rest

Example : ChatGPT API: The interface is designed with various hyperparameters that allow users to refine the AI’s responses. These include:

- **Temperature**: Influences the randomness of the AI’s responses.
- **Maximum Token Length**: Defines the maximum length of the model's output, including both tokens in the input and output of the message.
- **Stop Sequences**: Specific text strings that, when encountered by the model, cause it to stop generating further output.
- **Top P**: Also known as nucleus sampling, this method provides a dynamic selection of the number of words considered at each step of the model’s predictions. A lower value, such as 0.5, leads to safer and more focused outputs. A higher value, such as 0.9, includes a broader selection of words, resulting in more diverse outputs.

Using a LLM

1) Température

- Le paramètre de température influence le caractère aléatoire des réponses de l'IA. Une température plus élevée, telle que 1.0, encourage une sortie plus aléatoire, entraînant des réponses diverses mais potentiellement hors sujet. En revanche, une température plus basse, comme 0.2, incite l'IA à sélectionner des réponses plus déterministes, ce qui peut être bénéfique pour des sorties ciblées et spécifiques, mais peut manquer de variété.
- Exemple : si vous écrivez une histoire créative et souhaitez des éléments imprévisibles et imaginatifs, réglez la température sur une valeur plus élevée. Une invite pourrait être : "Écrivez une courte histoire sur un chat qui voyage dans le temps." Avec une température élevée, vous pourriez avoir un conte sauvage et fantastique avec des rebondissements imprévisibles.

2) Longueur maximale

- Ce paramètre contrôle la longueur de jeton maximale de la sortie du modèle, qui inclut à la fois les jetons dans l'entrée et la sortie du message. La définition d'une limite supérieure permet des réponses plus étendues, tandis qu'une limite inférieure permet de conserver une sortie courte et concise.
- Exemple : Pour de brèves réponses à des questions triviales telles que "Qui a remporté le prix Nobel de littérature en 2020 ?" vous souhaiterez peut-être définir la longueur maximale sur une valeur faible, en vous assurant que la réponse est concise et directe.

3) Séquences d'arrêt

- Les séquences d'arrêt sont des chaînes de texte spécifiques où, lorsque le modèle les rencontre, il cesse de générer d'autres sorties. Cette fonctionnalité peut être utile pour contrôler la longueur de la sortie ou demander au modèle de s'arrêter aux extrémités logiques.
- Exemple : pour une lettre officielle, vous pouvez utiliser "Sincèrement vôtre" comme séquence d'arrêt pour vous assurer que l'IA ne génère aucun texte supplémentaire au-delà de la fin appropriée de la lettre.

4) Haut P

- Le paramètre 'Top P', également connu sous le nom d'échantillonnage par noyau, est une méthode qui fournit une sélection dynamique du nombre de mots considérés à chaque étape des prédictions du modèle. Une valeur inférieure, comme 0.5, conduit à des sorties plus sûres et plus ciblées. Une valeur plus élevée, comme 0.9, inclut une sélection plus large de mots, conduisant à des sorties plus diversifiées.
- Exemple : Si vous créez une IA pour écrire des poèmes et souhaitez qu'elle utilise un large éventail de vocabulaire et de phrases stylistiques, vous pouvez définir " Top P " sur une valeur plus élevée. Une invite pourrait être : "Composer un poème sur la beauté de l'automne."

5) Pénalité de fréquence

- La pénalité de fréquence contrôle dans quelle mesure le modèle doit favoriser les mots moins fréquents. Une pénalité plus élevée (jusqu'à 1) encourage le modèle à utiliser des mots moins courants, tandis qu'une valeur inférieure (jusqu'à -1) encourage le modèle à utiliser des mots plus courants.
- Exemple : Si vous essayez de générer une proposition commerciale et que vous souhaitez éviter le jargon, vous pouvez définir la pénalité de fréquence sur une valeur inférieure pour que le langage reste clair et accessible.

6) Pénalité de présence

- Le paramètre Pénalité de présence affecte le degré de pénalisation du modèle pour la génération de nouvelles idées ou de nouveaux sujets qui n'étaient pas présents dans l'historique de la conversation. Des valeurs plus élevées encouragent le modèle à s'en tenir aux sujets déjà mentionnés, tandis que des valeurs plus faibles permettent au modèle d'introduire plus librement de nouveaux concepts.
- Exemple : pour les sessions de brainstorming, vous pouvez définir la pénalité de présence sur une valeur inférieure pour encourager un large éventail d'idées. Vous pouvez utiliser une invite telle que "Générer des stratégies marketing innovantes pour une plate-forme éducative en ligne".

Using a LLM

With codes (appels, tests, rag, agents, etc.)

Libraries :

Langchain, langraph,

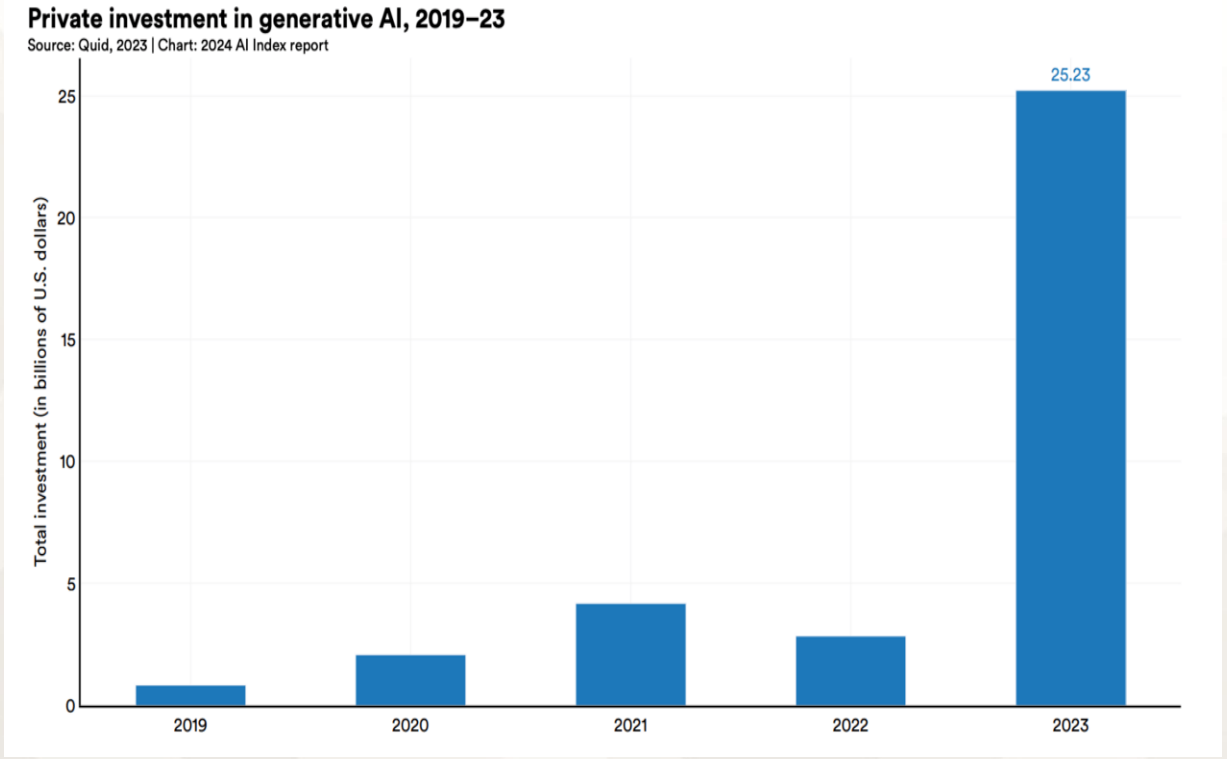
Llamaindex (specialised in RAG)

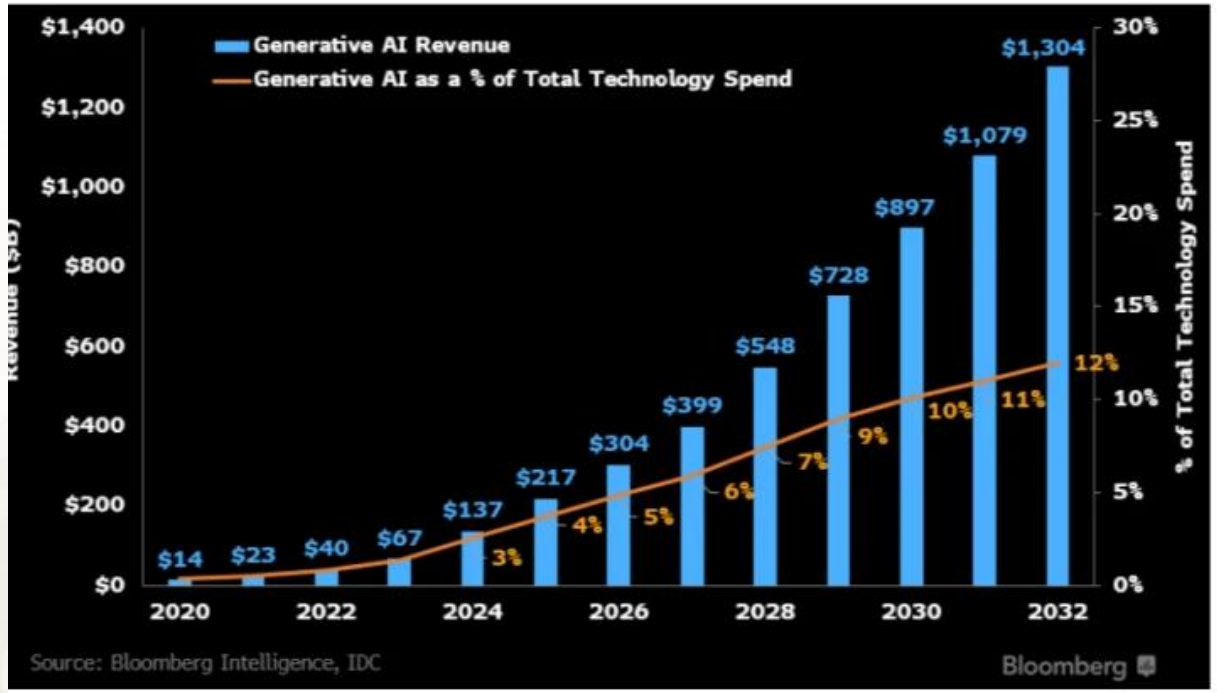
Python, langchain :

```
from langchain_ollama import ChatOllama
llm = ChatOllama( model = "llama3", temperature = 0.8, num_predict = 256, # other params ... )

messages = [ ("system", "You are a helpful translator. Translate the user sentence to French."), ("human", "I love programming."), ]

llm.invoke(messages)
```

(QUELQUES) PERSPECTIVES

LLM financial perspectives ?





LLM perspectives ?

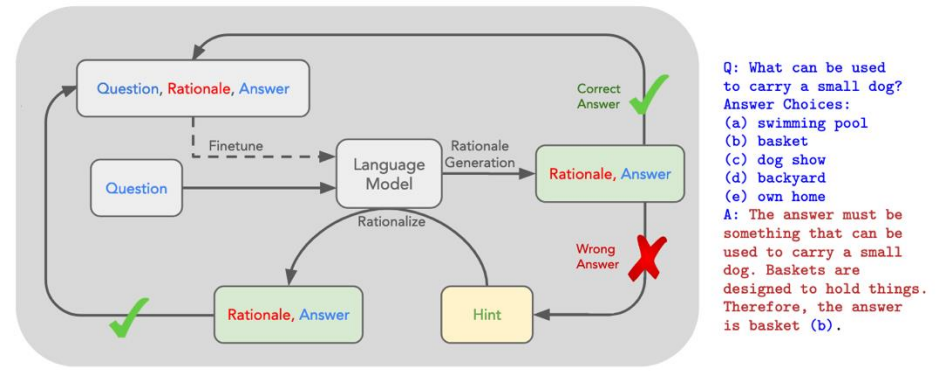
La croissance de la taille des LLMs est-elle durable ? Non.

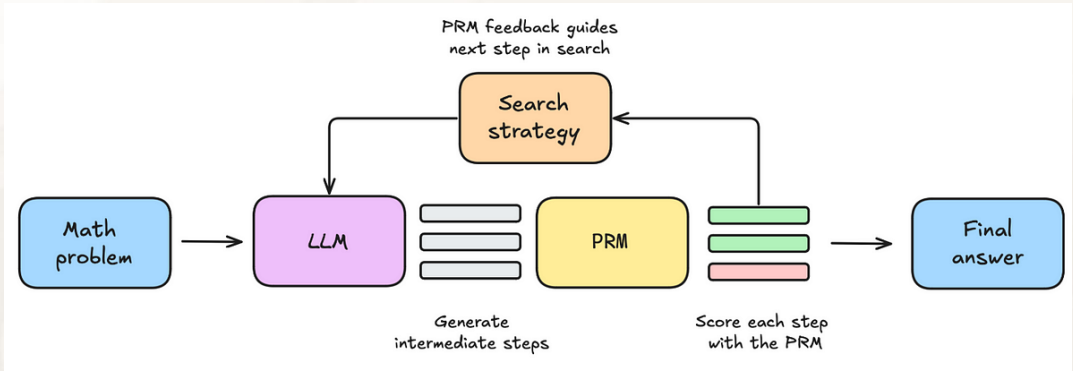
Les Small Language Models (SLMs) utilisables localement , tels que qwen2.5, Llama3.3, mistral-small, dont la performance serait presque équivalente à celle de GPT-3.5 (lancé en novembre 2022), offrent une approche prometteuse pour développer des modèles plus petits mais puissants.

LLM perspectives ?

Is the growth in the size of LLMs sustainable? no

- Networks of SLMs (see Langroid https://github.com/langroid/langroid)
- Smaller models with reasonning capabilities
  - Chain of Thoughts, STaR, Tree of thoughts) and test time compute scaling





STaR: Self-Taught Reasoner https://arxiv.org/abs/2203.14465

LLM perspectives ?

- *There is no question that AI will eventually reach and surpass human intelligence in all domains. But it won't happen next year. And it won't happen with the kind of Auto-Regressive LLMs currently in fashion (although they may constitute a component of Yann LeCun (@ylecun)*

- « Le futur de l'IA n'est pas dans les *LLM*, mais dans l'IA guidée par les objectifs » *Yann LeCun* Objective-Driven AI. https://www.college-de-france.fr/sites/default/files/media/document/2024-02/Sagot-2023-2024_Seminaire_Yann-LeCun.pdf

1. Learning representations and predictive models of the world
   - Using Self-supervised learning from video and other sensory inputs
   - learning to represent the world in a non task-specific way
   - Learning predictive world models for planning and control

2. Learning to reason, like Daniel Kahneman's "System 2"
   - Beyond feed-forward, System 1 subconscious computation.
   - Making reasoning compatible with learning.
   - Reasoning and planning as energy minimization.

3. Learning to plan complex actions to satisfy objectives

LLM perspectives ?

- Weaknesses of LLMs in their interactions with the physical real world
- Mesh of agents (specialized SLM with functions + LLMs)
- Ethical aspects ?
- Sustainability (green IT, green AI ?)

Merci