Introduction aux LLM

Mehdi Ammi
Professeur, Université de Paris 8
mehdi.ammi@univ-paris8.fr

Sources

- **Generative AI and LLMs, Snowflake Special Edition** (2024) <Lien>
- **Foundational Large Language Models & Text Generation, Google DeepMind / Google Cloud** (2025) <Lien>
- ***The Age of Copilots: How Large Language Models are Reshaping Work* , Microsoft (2023).**

Qu’est-ce que la Gen AI ?

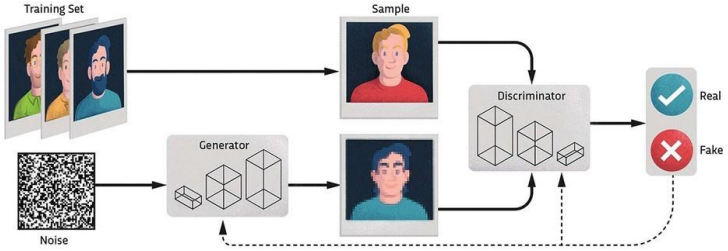
La *Generative AI (Gen AI)* est une **IA créative** capable de produire du contenu **original** à partir de simples instructions (*prompts*). Elle se distingue de l’IA prédictive, qui se limite à analyser ou classer des données existantes.

- **Caractéristiques clés**
  - **Production multimodale** : génération de **texte, images, sons, vidéos, code**.
  - **Créativité contextuelle** : chaque sortie est adaptée à la demande (pas une copie brute).
  - **Polyvalence** : utilisable dans des domaines très variés (santé, droit, éducation, art, industrie).

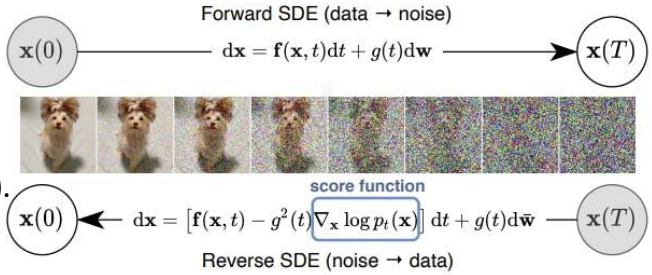
Modèles de l'IA générative

L'IA générative regroupe plusieurs **familles de modèles** capables de créer du contenu nouveau :

- **GANs (Generative Adversarial Networks, 2014)**
  - Fonctionnement : deux réseaux (**générateur + discriminateur**) s'entraînent en compétition.
  - Usages : deepfakes, visages réalistes, génération d'images haute fidélité.
  - *Exemples* : **StyleGAN (2018)**, deepfake apps.



- **Diffusion Models (2015 → popularisés dès 2020)**
  - Fonctionnement : **ajout puis suppression progressive de bruit** pour générer une image.
  - Usages : création d'images, art visuel, design.
  - *Exemples* : **DALL·E (2021)**, **Stable Diffusion (2022)**, **MidJourney (2022)**.



- **LLMs (Large Language Models, 2018–2020)**
  - Fonctionnement : modèles autoregressifs basés sur **Transformers**.
  - Usages : génération de texte, code, multimodalité (texte + image + son).
  - *Exemples* : **GPT-3 (2020)**, **LLaMA 2 (2023)**, **Gemini 1.5 (2024)**.

En résumé :

- **Diffusion Models & GANs** → plutôt orientés **vision / images**.
- **LLMs** → orientés **langage naturel (NLP) et multimodalité**.
- Ensemble, ils constituent le cœur de la **révolution IA générative**.

Les LLMs

Les **Large Language Models (LLMs)** sont une **catégorie de l'IA générative** spécialisée dans le langage.
Ils reposent sur l'architecture **Transformer** et sont entraînés sur des **corpus massifs** de texte.

- Objectif principal : **prédire le mot suivant** dans une séquence.
- Résultat : ils développent des **capacités avancées de langage**.

Exemples de tâches

- **Question simple** :
  "The capital of France is …" → "Paris"
- **Traduction** :
  "Bonjour, comment vas-tu ?" → "Hello, how are you?"
- **Résumé** :
  Texte de 5 pages → Résumé en 5 lignes
- **Code** :
  "Write a Python function to compute factorial" → génération du script complet
- **Conversation** :
  Utilisateur : "Qui a écrit *Les Misérables* ?"
  Modèle : "Victor Hugo, publié en 1862."

Les LLMs transforment une simple prédiction statistique en un **outil polyvalent** :

- compréhension,
- génération,
- traduction,
- raisonnement,
- dialogue interactif.

Dans la frise des **modèles génératifs** (GANs → Diffusion → LLMs), les LLMs apportent la **dimension langage et multimodalité**.

Caractéristiques des LLMs

Les *LLMs* se distinguent par leur taille colossale et leur capacité à apprendre des représentations riches du langage.
Leur puissance vient à la fois du **volume d'entraînement** et de l'**architecture Transformer**

- **Réseaux neuronaux géants** : des centaines de milliards à plus d'un trillion de paramètres
  Exemple : GPT-3 (175B), PaLM (540B), GPT-4 (estimé >1T)
  Attention : **100B+ paramètres → "B" = *billion* en anglais = un milliard en français**

- **Entraînement sur des trillions de tokens** : textes, articles, code, dialogues
  Sources : Wikipédia, forums, dépôts GitHub, articles scientifiques

- **Capacité de généralisation** : apprennent des patterns linguistiques réutilisables dans divers contextes
  Exemple : répondre à une question historique ou générer du code Python

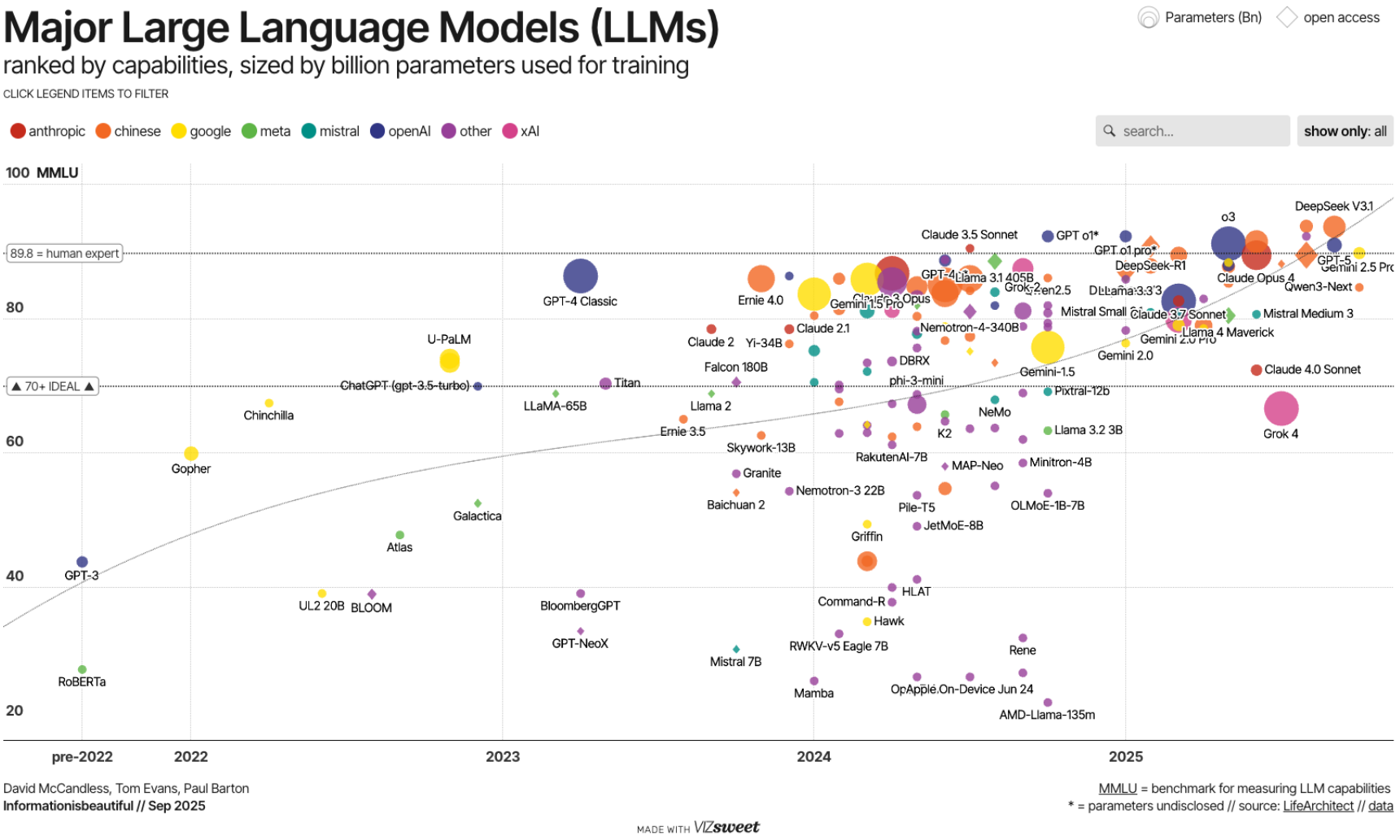
- **Génération fluide et cohérente** : texte naturel, proche du langage humain
  Exemple : écrire un résumé, un poème, un mail professionnel

- **Adaptabilité** : modèles réutilisables et ajustables via *prompt engineering*, RAG ou fine-tuning
  Exemple : BioBERT spécialisé en médecine, Legal-BERT en droit

LLMs vs Deep Learning

- **Réseau de Deep Learning "classique" (1M – 100M paramètres)**
  - **Mémoire** : quelques centaines de Mo à quelques Go pour stocker les poids.
  - **GPU nécessaires** : 1 à 4 GPU standards (ex. NVIDIA V100, A100).
  - **Temps d'entraînement** : heures → quelques jours.
  - **Données** : millions d'images ou textes annotés.
  - **Exemple** :
    - YOLOv3 (62M paramètres) → ~240 Mo de poids.
    - ResNet-50 (25M paramètres) → ~100 Mo de poids.
- **Large Language Model (100B – 1T+ paramètres)**
  - **Mémoire** :
    - GPT-3 (175B) → ~700 Go de poids (en FP16).
    - GPT-4 (estimé 1T) → plusieurs **To** de mémoire pour les paramètres seuls.
  - **GPU nécessaires** : milliers de GPU haut de gamme en parallèle (NVIDIA A100/H100).
    - Exemple : GPT-3 a nécessité environ **10 000 GPU V100**.
  - **Temps d'entraînement** : plusieurs semaines → mois.
  - **Coût énergétique** : consommation équivalente à des milliers de foyers/an.
  - **Données** : trillions de tokens (texte du web, code, articles, dialogues).

Tailles des modèles



A visualisation of major large-language models (LLMs), ranked by performance, using MMLU (Massive Multitasks Language Understanding) a benchmark for evaluating the capabilities of large language models.

<Lien>

Tailles des modèles

Taille vs Contexte

- **GPT-2** (2019, OpenAI, NLP) : 1.5B paramètres, contexte ~1 024 tokens
- **GPT-3 (**2020, OpenAI, NLP) : 175B paramètres, contexte 2 048 tokens
- **PaLM** (2022, Google) : 540B paramètres, contexte ~8 000 tokens
- **LLaMA 2** (2023, Meta) : 7B, 13B et 70B paramètres, contexte 4 096 tokens
- **GPT-4** (2023, OpenAI) : >1T paramètres (estimés), contexte 8k à 32k tokens
- **Gemini 1.5** (2024, Google DeepMind) : taille non précisée, contexte jusqu'à 1M tokens
- **Grok-4** (2024, xAI) : taille non publiée, contexte jusqu'à 128k tokens (Heavy / Fast)
- **GPT-5** (2025, OpenAI) : taille non confirmée, contexte ~400k tokens
- **Grok-4** (2025, xAI, version améliorée) : taille non confirmée, contexte ~256k tokens
- **Gemini 2.5 Pro** (2025, Google DeepMind) : taille non précisée, contexte ~1M tokens
- **LLaMA 4 Scout** (2025, Meta) : taille non précisée, contexte ~10M tokens
- **Gemma-3** (2025, Google) : plusieurs tailles, contexte ~131k tokens (32k pour la plus petite)

Synthèse

- De **1.5B (2019)** → à **>1T (2023)** en seulement 4 ans (**x1000**).
- **2019–2023** : la course aux **paramètres** (1.5B → 1T+).
- **2024–2025** : la course au **contexte** (1k → 10M tokens).

L’importance des données

Les LLMs tirent leur puissance des données sur lesquelles ils sont entraînés.

La diversité, la qualité et la quantité des données conditionnent directement la pertinence, la fiabilité et la capacité de généralisation du modèle.

- **Données massives** : les modèles de dernière génération sont entraînés sur des trillions de tokens (texte, code, articles).
- **Données non structurées** : 80–90 % des données disponibles sont sous forme libre (textes, images, vidéos, logs).
- **Sources multiples** : internes (CRM, support client), partenaires (bases sectorielles), externes (web, publications).
- **Corrélation** : plus le volume et la variété des données sont élevés, plus le modèle capte de nuances linguistiques et contextuelles.
- **Qualité > quantité** : des données bruitées ou biaisées se traduisent par des réponses erronées ou discriminatoires.

L’importance des données

- **Taille du LLM**
  - Plus de paramètres → plus grande capacité d’**apprentissage** et de **représentation**.
  - Mais au-delà d’un certain seuil, les gains sont marginaux si les données ne suivent pas.
    - Exemple : GPT-3 (175B) vs GPT-4 (~1T estimé) → gains limités si corpus pas mieux filtré.

- **Taille et qualité du corpus d’entraînement**
  - C’est le **facteur le plus critique**.
  - Sans données diversifiées, riches et propres, un modèle peut être gigantesque mais inutile.

- **Les facteurs clés d’un LLM**
  - Taille du modèle (paramètres)- > Importance : moyenne à forte
  - Qualité + diversité des données -> Importance : maximale

L’importance des données

- **ChatGPT / GPT-3 (OpenAI, 2020)**
  - **Corpus** : Common Crawl (énorme snapshot du web), Wikipedia, livres (BooksCorpus), articles scientifiques (arXiv), code.
  - **Volume** : ≈ **570 Go de texte filtré** → soit environ **300 milliards de tokens**.
  - **Taille modèle** : 175B paramètres.

- **GitHub Copilot (Codex, dérivé de GPT-3)**
  - **Corpus** : dépôts GitHub publics + forums développeurs (StackOverflow) + doc techniques.
  - **Taille** : Codex a été entraîné sur une base comprenant **plusieurs dizaines de milliards de lignes de code**.
  - Taille exacte non publique, mais estimée à **>100 Go de code filtré** en plus du corpus GPT-3.

- **PaLM (Google, 2022)**
  - **Corpus** : mélange de Common Crawl (C4 dataset ~ 750 Go), Wikipedia, livres, code, dialogues.
  - **Volume** : ≈ **780 milliards de tokens**.
  - **Taille modèle** : 540B paramètres.

- **LLaMA 1 (Meta, 2023)**
  - **Paramètres** : 7B → 65B
  - **Corpus** : ~1.4 To de texte nettoyé
  - **Tokens** : ~1 trillion (1 000 milliards)
  - Basé sur données publiques (Common Crawl, Wikipedia, Project Gutenberg, arXiv).

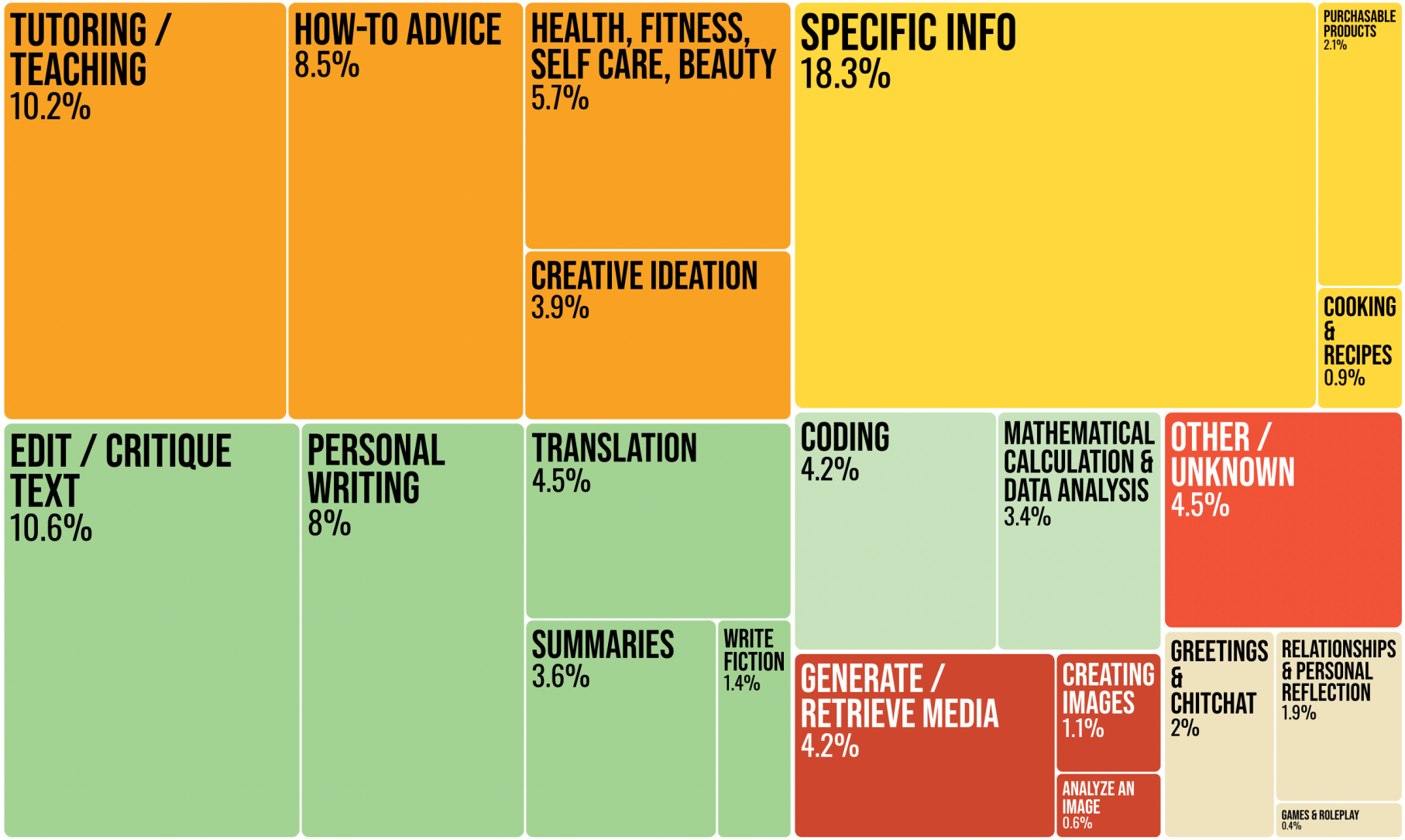
Pourquoi les LLM sont importants ?

Les **LLMs** représentent une **rupture technologique** dans le NLP.
Ils se distinguent par leur **polyvalence**, leur **productivité**, leur rôle dans l'**innovation** et leur **accessibilité**.

- **Polyvalence**
  - Traduction multilingue
  - Résumé de documents
  - Génération de code
  - Dialogue interactif
- **Productivité**
  - Gain de temps considérable
  - Automatisation de tâches
  - Traitement de grands volumes de données
- **Innovation**
  - Nouveaux services (assistants, copilotes, tuteurs virtuels)
  - Nouveaux métiers (Prompt Engineer, expert en LLMOps)
  - Avancés pour la recherche et R&D : Prix Nobel 2024 (Deep Learning + Transformers) pour la prédiction de molécules
- **Accessibilité**
  - Outils utilisables par des non-experts
  - Démocratisation de l'accès au savoir
  - Usage en éducation, recherche et entreprise

What are people using ChatGPT for?

Multimedia | Other | Practical Guidance | Seeking Information | Self Expression | Technical help | Writing



David McCandless **Information is Beautiful**

source: OpenAI

Foundation Models : la base des LLMs

Les *Foundation Models* sont de **très grands modèles entraînés** sur des corpus massifs et variés.
Ils constituent le **socle** sur lequel on peut bâtir des applications spécialisées.

- **Caractéristiques**
  - Entraînés sur du texte, du code, des images, voire de l'audio.
  - Servent de point de départ pour de nombreuses tâches.
  - Peuvent être adaptés via :
    - *Fine-tuning* (réentraînement ciblé).
    - *RAG* (connexion à une base documentaire).
    - *Prompt engineering* (optimisation des instructions).

- **Exemples concrets**
  - **ChatGPT** (OpenAI) → basé sur GPT-3/GPT-4.
  - **Google Bard / Gemini** → basé sur PaLM/Gemini.
  - **GitHub Copilot** → basé sur Codex (dérivé de GPT-3).

Les *Foundation Models* sont comme des **"briques de Lego géantes"** : on ne les recrée pas, on les adapte.

Types de LLMs

À partir des *Foundation Models*, on distingue plusieurs **catégories de LLMs**, selon leur **corpus d'entraînement,** leur **adaptation**, leur **usage** et leur **domaine**.

- **Généralistes**
  - Entraînés *from scratch* sur des corpus massifs et variés (web, livres, articles, code).
  - Polyvalents → rédaction, résumé, traduction, conversation.
  - *Exemples* : GPT (OpenAI), LLaMA (Meta).

- **Spécialisés**
  - Obtenus principalement par **fine-tuning** de modèles généralistes sur un corpus ciblé.
  - Plus performants dans leur domaine spécifique.
  - *Exemples* : **BioBERT** (biomédical), **Legal-BERT** (juridique), **Code Llama** (programmation).

- **Multimodaux**
  - Extension d'architecture : ajout de modules vision, audio, vidéo.
  - Entraînés sur des corpus multimodaux texte + image + son.
  - *Exemples* : GPT-4 multimodal (OpenAI), Gemini (Google DeepMind).

De quoi est composé ChatGPT ?

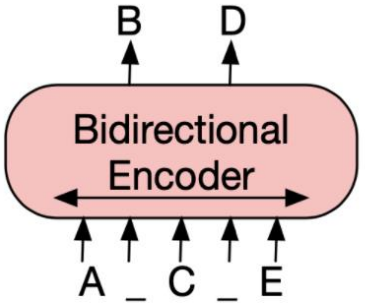
ChatGPT n'est pas qu'un simple modèle de langage (**GPT**).
C'est un **écosystème coordonné** combinant plusieurs briques technologiques.

- **Composants clés**
  - **GPT (LLM)** : cœur du système, basé sur Transformers.
  - **RLHF (Reinforcement Learning with Human Feedback)** : améliore la qualité et l'alignement des réponses.
  - **Filtres de sécurité** : bloquent contenus sensibles, biaisés ou illégaux.
  - **RAG & Plugins** : connexion à des sources externes (bases documentaires, web, calculs).
  - **Autres modèles spécialisés** :
    - **Whisper** → transcription/traduction audio.
    - **CLIP** → texte + image.
    - **DALL·E** → génération d'images.
    - **Codex** → génération de code (GitHub Copilot).

BERT vs GPT : Compréhension vs Génération

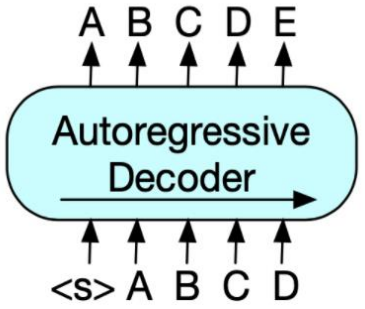
Les deux modèles reposent sur l'architecture Transformer mais poursuivent des objectifs différents. **BERT se concentre sur l'analyse et la compréhension**, alors que **GPT produit du langage et relève de l'IA générative**.

- **BERT : Comprendre le langage**
  - **Lit une phrase dans les deux sens pour analyser le sens global.**
  - **Architecture :** encodeur bidirectionnel → lit la phrase entière.
  - **Entraînement :** *Masked Language Modeling* (mots masqués à prédire).
  - **Forces :** analyse contextuelle, recherche d'information, classification.
  - **Limites :** ne sait pas générer du texte fluide.
  - **Exemples d'usages :**
    - Google Search → comprendre la requête
    - Analyse de sentiments sur avis clients.
    - Extraction d'entités dans un contrat.



BERT

- **GPT : Générer du langage**
  - **Lit une phrase mot par mot, de gauche à droite.**
  - **Architecture :** décodeur unidirectionnel → prédit le mot suivant.
  - **Entraînement :** *Causal Language Modeling* (compléter une séquence).
  - **Forces :** génération cohérente et fluide, créativité, polyvalence.
  - **Limites :** risque d'hallucinations, dépendance au prompt.
  - **Exemples d'usages :**
    - ChatGPT → conversation et rédaction d'articles.
    - GitHub Copilot → génération de code.
    - Résumé de documents et traduction multilingue.



GPT

Historique rapide de l'IA

L'intelligence artificielle s'est construite par vagues successives.
Chaque période a apporté de nouvelles approches, des ruptures technologiques et des usages élargis.
L'arrivée des Transformers en 2017 a ouvert la voie aux LLMs modernes.

- **Années 1950 : Systèmes experts basés sur des règles**
  → Logique conditionnelle (« si… alors… »), utilisées dans la médecine ou la finance.
  Exemple : MYCIN (1972) pour l'aide au diagnostic médical.

- **1958–1970 : Premiers réseaux de neurones (Perceptron)**
  → Frank Rosenblatt invente le Perceptron, capable d'apprendre à reconnaître des motifs simples.
  Exemple : Perceptron pour la reconnaissance d'images simples.

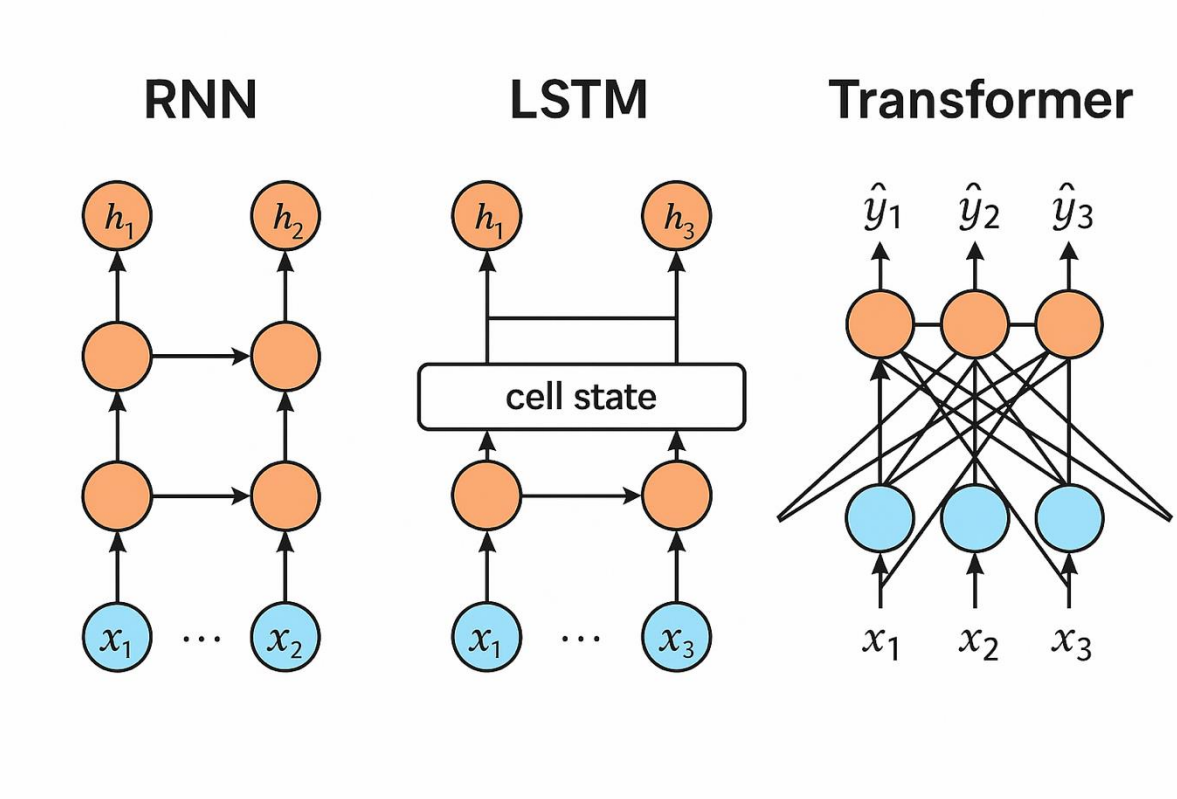
- **Années 1980 : Machine Learning (statistiques et algorithmes)**
  → Utilisation d'arbres de décision, **régressions**, SVM. Nécessite des données annotées.
  Exemple : algorithmes de détection de fraude bancaire.

- **Années 2010 : Deep Learning (CNN, RNN, LSTM)**
  → Révolution dans la vision par ordinateur et le NLP, grâce à la puissance des GPU.
  Exemple : reconnaissance faciale sur smartphones, assistants vocaux Siri/Alexa.

- **2017 : Transformers et modèles séquentiels parallélisés**
  → Introduction de l'auto-attention, entraînement sur des corpus massifs.
  Exemple : BERT (Google, 2018) pour la recherche contextuelle → précurseur des LLMs type GPT.

De RNN à Transformer

Avant 2017, les modèles séquentiels (RNN, LSTM, GRU) dominaient le NLP mais étaient limités.



De RNN à Transformer

Modèle	Principe	Par rapport aux autres	Limites
RNN (1980)	Séquence traitée pas à pas, mémoire via état caché.	Première génération, ouvre la voie au séquentiel.	Perte d'info longue (vanishing gradient), lent (pas parallélisable).
LSTM (1997)	Ajoute une mémoire longue $c_t$et des portes (oublier/ajouter/sortir).	Corrige les faiblesses des RNN, plus robuste pour le NLP.	Toujours séquentiel, lourd à entraîner.
Transformer (2017)	Auto-attention : chaque mot voit tous les autres en parallèle.	Rupture → parallélisme et scalabilité, base des LLMs.	Coût élevé (attention quadratique), gros besoins en ressources.

De RNN à Transformer

Phrase : *“Le tigre a bondi sur sa proie. Il était affamé.”*

RNN

- La phrase est lue mot par mot.
- L’état caché transporte un résumé du passé.
- Quand on arrive à *« Il »*, l’information sur *« tigre »* est lointaine et a probablement été **oubliée**.
- Le RNN risque donc d’associer *« Il »* au mot le plus proche (*« proie »*).
  Résultat : mauvaise interprétation → *« la proie était affamée »*.

LSTM

- La mémoire à long terme $c_t$permet de mieux garder en mémoire des informations anciennes.
- Grâce aux portes, l’info importante (*« tigre »*) peut être conservée et utilisée plus tard.
- Quand on lit *« Il »*, le modèle a plus de chances de relier le pronom à *« tigre »*.
  Résultat : meilleure interprétation → *« le tigre était affamé »*.
  Mais si le texte est très long et complexe, l’info peut quand même se diluer.

Transformer

- Avec l’auto-attention, *« Il »* peut directement “regarder” **tous les autres mots de la phrase** en parallèle.
- L’attention attribue un poids plus fort à *« tigre »* qu’à *« proie »*.
- Pas besoin de transporter l’info pas à pas → dépendances longues gérées sans perte.
  Résultat : interprétation correcte et robuste → *« le tigre était affamé »*.

Résumé

- **RNN** : oublie le contexte lointain → confusion (*« Il »* → *« proie »*).
- **LSTM** : mémoire améliorée → souvent correct (*« Il »* → *« tigre »*).
- **Transformer** : dépendances directes → toujours correct (*« Il »* → *« tigre »*).

De RNN à Transformer

- **RNN**
  - Contexte utile : ~ **50–100 tokens**
  - ≈ **40–70 mots**
  - ≈ **150–400 caractères**

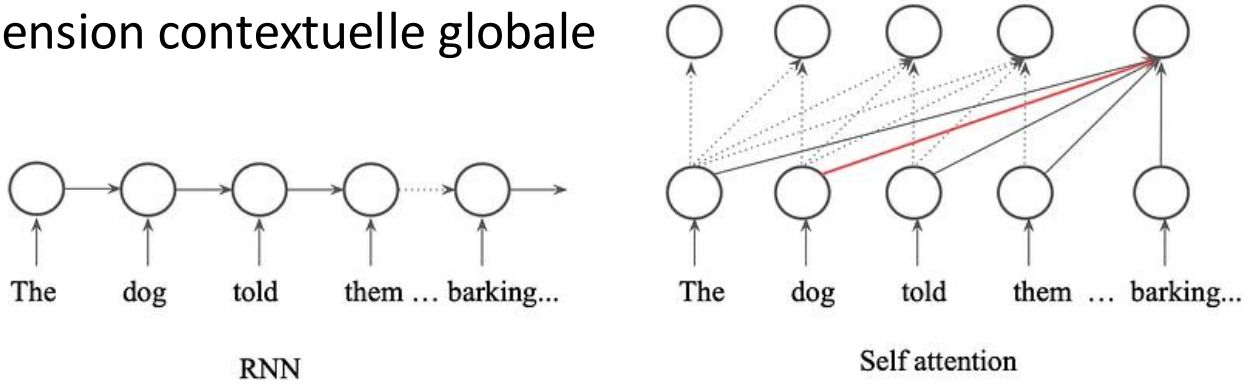
- **LSTM**
  - Contexte utile : ~ **200–500 tokens**
  - ≈ **150–350 mots**
  - ≈ **600–2 000 caractères**

- **Transformers**
  - **BERT (2018)** : 512 tokens ≈ 350 mots ≈ 1 500–2 000 caractères
  - **GPT-2 (2019)** : 1 024 tokens ≈ 750 mots ≈ 3 000–4 000 caractères
  - **GPT-3 (2020)** : 2 048 tokens ≈ 1 500 mots ≈ 6 000–8 000 caractères
  - **GPT-4 / Claude 2 (2023)** : 128k tokens ≈ 90 000 mots ≈ 400 000–500 000 caractères
  - **LLaMA 4 (2025, Scout)** : annoncé jusqu'à **10M tokens** (≈ 7,5M mots, ≈ 30–40M caractères)

Le rôle de l'auto-attention

L'auto-attention est le cœur du Transformer. Elle pondère l'importance des mots.

- Compare chaque mot avec tous les autres
- Capture des dépendances longues et complexes
- Compréhension contextuelle globale



Exemples :

- Phrase : *"Le patient a pris un médicament car il avait de la fièvre."*
  → "il" est relié au **patient**, pas au "médicament".
- Phrase : *"La voiture que Pierre a achetée hier est rouge."*
  → Le mot "rouge" se rattache à **voiture** même s'ils sont éloignés.
- Traduction :
  "The cat sat on the mat." → "Le chat s'est assis sur le tapis."
  → L'auto-attention permet de garder le lien correct entre *sat* → *s'est assis*.

Multi-Head Self-Attention

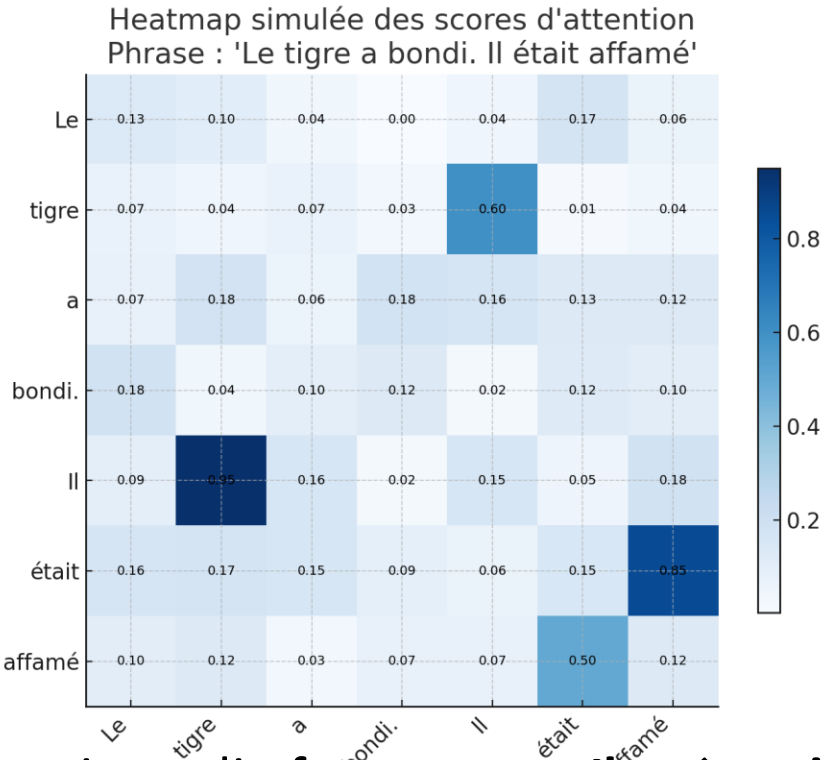
L’auto-attention n’est pas appliquée une seule fois mais en **plusieurs têtes parallèles**.

Chaque tête apprend à capturer une **relation différente** (grammaire, sémantique, dépendances à longue distance, etc.).

Les résultats sont ensuite concaténés et fusionnés pour donner une vision plus riche du contexte.

Multi-Head Self-Attention

Exemple : Dans *« Le tigre a bondi. Il était affamé »* :



- Une tête d'attention relie fortement **« Il » → « tigre »** (résolution correcte de la coréférence).
- Une autre relie **« était » → « affamé »** (association verbe–attribut).
- Les autres relations reçoivent des poids plus faibles.

Parallélisme et efficacité

Les Transformers traitent une séquence entière en parallèle, alors que les RNN/LSTM fonctionnent de façon séquentielle (mot après mot).

Cette différence fondamentale explique pourquoi les LLMs modernes ont pu émerger.

Parallélisme

- Chaque mot de la séquence peut être comparé simultanément à tous les autres grâce à l'auto-attention.
- Exploite pleinement la puissance des GPU/TPU, conçus pour exécuter des milliers d'opérations en parallèle.
- Entraînement beaucoup plus rapide : ce qui prenait des semaines avec un RNN peut être fait en quelques jours avec un Transformer.

Scalabilité

- Les Transformers se généralisent bien lorsqu'on augmente leur taille (nombre de couches, têtes d'attention, paramètres).
- Passage possible de millions de paramètres (BERT) à centaines de milliards (GPT-3, PaLM), voire plus de 1T (GPT-4 estimé).
- Plus un modèle est grand, plus il capte de nuances linguistiques et de connaissances.

Efficacité

- Permettent d'entraîner des modèles sur des corpus gigantesques : trillions de tokens (textes web, articles scientifiques, code, dialogues).
- Optimisés pour tirer parti du big data sans être bloqués par des limites techniques des RNN/LSTM.
- Résultat : une IA capable de généraliser et d'être utilisée dans des contextes très différents.

Capacités générales : traitement de texte

Les LLMs excellent dans les tâches NLP classiques.

+ Traduction multilingue
+ Résumé de documents longs
+ Réécriture et correction grammaticale
+ Recherche d’informations

Capacités générales : génération

Les LLMs ne se contentent pas de comprendre, ils créent du contenu.

+ Génération de code (Python, SQL, JavaScript)
+ Création littéraire (histoires, poèmes)
+ Production de mails ou d'articles
+ Dialogue interactif (chatbots)

Exemple : Générer un script Python pour calculer la factorielle.

Capacités générales : classification et analyse

Les LLMs (GPT, LLaMA, Gemini…) sont conçus pour **générer du texte fluide et cohérent**.
Mais ils peuvent aussi accomplir des tâches classiques de **NLP (Natural Language Processing)**, historiquement dominées par des modèles comme **BERT**.

Exemples de tâches possibles :

+ Détection de spam ou phishing
+ Analyse de sentiments
+ Extraction d'entités (noms, dates, lieux)
+ Résumé statistique d'avis

Exemple : 200 avis clients → 80 % positifs, 15 % neutres, 5 % négatifs.

Attention :

- LLMs peuvent **se tromper dans les comptages**
- Pour des données massives, il vaut mieux les **coupler avec des outils analytiques** (SQL, Pandas, etc.)

Raisonnement étape par étape

Les **LLMs** peuvent résoudre un problème en **séquences d'actions simples**, ce qui réduit les erreurs et rend la solution vérifiable.
L'idée : **décomposer**, calculer/interpréter **étape par étape**, puis **synthétiser** le résultat.

Principes

- **Décomposition** : identifier données → inconnues → formules/règles → plan d'action.
- **Formats utiles** : liste numérotée, tableau "entrée → opération → sortie", pseudo-code court.
- **Cas d'usage** : arithmétique, conversions d'unités, logique conditionnelle, requêtes SQL, procédures métier.
- **Bonnes pratiques de prompt** : "Résous en **étapes numérotées**", "Calcule **d'abord**…, **puis**…", "Donne **les hypothèses** puis **le résultat**".
- **Vérification** : recalcule la dernière étape, impose un **format de sortie** (ex. nombre/JSON), précise **l'arrondi**.
- **Limites** : sensibilité au prompt, risques d'erreurs cumulées → **contrôle final** nécessaire.

Exemples

- **Calcul simple** : *Paul a 3 pommes, il en donne 1* → Étapes : 3 − 1 = **2**.
- **TVA** : *HT = 120 €, TVA 20 %* → 120 × 0,20 = **24** → TTC = 120 + 24 = **144 €**.
- **Conversion** : *5 km/h en m/s* → 5 × 1000 / 3600 = **1,39 m/s**.

Raisonnement avancé

Avec des **prompts adaptés**, les LLMs peuvent résoudre des **problèmes complexes** en décomposant le raisonnement étape par étape et en vérifiant les résultats.

Domaines clés

- **Mathématiques** : équations, géométrie, optimisation
- **Sciences** : physique (mécanique, circuits), biologie (analyse de données)
- **Logique juridique** : analyse structurée de cas (IRAC : Issue, Rule, Application, Conclusion)

Méthodes

- Formaliser données et hypothèses
- Écrire formules / règles avant calcul
- Résoudre symboliquement puis numériquement
- Vérifier unités, ordre de grandeur, cas limites

Limites

- Risque d'erreurs cumulées
- Sensibles au prompt → nécessité de guider la réponse
- Non substitut à des outils spécialisés (calcul formel, expertise humaine)

Raisonnement avancé

Exemple : Résolution détaillée d'un problème de géométrie.

- Un LLM **ne raisonne pas "vraiment" comme un humain** : il prédit la suite la plus probable dans une séquence de texte.
- Si on lui pose directement la question *« Quelle est l'aire de ce triangle ? »*, il peut :
  - soit donner **la bonne réponse** par chance (il a "vu" ce problème ou similaire dans ses données d'entraînement),
  - soit donner une réponse **fausse** ou incomplète (hallucination, erreur de calcul).
- **Comment ça marche avec un prompt guidé ?**
  - L'humain peut demander explicitement au LLM de **raisonner étape par étape**.
  - Exemple de prompt efficace : *"Résous ce problème en détaillant les étapes intermédiaires avant de donner le résultat final."*
  - Le LLM va alors **imiter une logique humaine** :
    - Identifier les données
    - Appliquer une formule
    - Calculer
    - Vérifier
- C'est ce qu'on appelle la méthode **"Chain-of-Thought" (raisonnement en chaîne)**.

Raisonnement avancé

- **Peut-il se débrouiller seul ?**
  - Parfois oui : s'il a déjà vu beaucoup de problèmes similaires dans son corpus, il reproduit une "solution type".
  - Mais la **décomposition claire** est souvent déclenchée par le **prompt** → c'est **l'utilisateur qui force le raisonnement explicite**.
  - Dans certains cas, les LLMs entraînés avec des techniques comme **"Chain-of-Thought fine-tuning"** ou **"ReAct" (reason + act)** sont plus capables d'appliquer cette décomposition par défaut.

- **Donc :**
  - **L'humain (via le prompt)** joue un rôle crucial pour guider le LLM vers un raisonnement étape par étape.
  - Sans guidage, le LLM peut donner une bonne réponse, mais sans justification fiable.
  - Avec guidage, on augmente les chances d'obtenir une solution correcte et vérifiable.

Limites du raisonnement

Les LLMs ne raisonnent pas comme des humains :

- Leur raisonnement repose sur des **corrélations statistiques** apprises dans les données, et non sur une logique stricte ou des règles formelles. Cela entraîne plusieurs limites.

- **Erreurs de calcul fréquentes**
  - Les LLMs ne manipulent pas réellement les nombres, mais des **séquences de tokens**.
  - Ils peuvent donner des résultats **plausibles mais faux**.
  - Exemple : *123 × 456* → réponse attendue : **56 088**, mais un LLM peut produire un autre nombre “semblant correct”.

- **Raisonnements incohérents**
  - Les étapes d’une réponse peuvent sembler logiques individuellement, mais **ne s’enchaînent pas correctement**.
  - Exemple :
    - Étape 1 correcte : “Paul a 3 pommes, il en donne 1 → il reste 2.”
    - Étape 2 erronée : “Puis il en achète 2 → résultat final = 2 (au lieu de 4).”

Limites du raisonnement

- **Dépendance au prompt**
  - La qualité du raisonnement dépend fortement de la **formulation de la question**.
  - Une mauvaise consigne → réponse approximative ou erronée.
  - Exemple :
    - **Prompt direct** : « Calcule le TTC pour 120 € HT à 20%. »
    - **Prompt guidé** : « Calcule **étape par étape** le TTC pour 120 € HT avec **TVA=20%**. Donne **HT, TVA, TTC** séparés et vérifie le total. »
      Attendu (décomposé) : TVA = 120 × 0,20 = **24** → TTC = **144**.

- **Conséquences pratiques**
  - Risque de résultats faux si on prend la réponse “brute” sans vérification.
  - Obligation de mettre en place des **garde-fous** :
    - Vérification avec des outils externes (calculatrice, moteur de règles).
    - Demander au modèle de **montrer ses étapes** pour détecter les erreurs.
    - Utiliser des **prompts structurés** (“raisonne étape par étape”, “vérifie le résultat”).

Multimodalité : texte + image

Les **LLMs classiques** traitent uniquement du texte.
Pour intégrer les images, ils sont **couplés à des modèles de vision par ordinateur** (CNN, ViT…).
Cela donne naissance aux **VLMs (Vision-Language Models)** ou **VLLMs (Vision Large Language Models)**.

- **Articulation avec les modèles de vision**
  - **Étape 1 : Vision** → un modèle de Computer Vision (ex. CNN ou Vision Transformer) encode l'image en **vecteurs (embeddings visuels)**.
  - **Étape 2 : Langage** → un LLM (ex. GPT-4, Gemini) reçoit ces vecteurs et les combine avec du texte.
  - **Étape 3 : Multimodalité** → le modèle peut générer une description, répondre à une question sur l'image, ou comparer texte ↔ image.

- **Exemples concrets de VLLMs**
  - **CLIP (OpenAI)** : associe images + textes → utilisé pour la recherche multimodale.
  - **Flamingo (DeepMind)** : modèle multimodal texte + image.
  - **GPT-4 multimodal (OpenAI)** : capable de lire une image (ex. photo, graphique) et de générer une réponse textuelle.
  - **Gemini (Google DeepMind)** : multimodal avancé (texte, image, code, audio).

- **Cas d'usage**
  - **Médical** : analyser une radiographie et générer une description diagnostique.
  - **Éducation** : expliquer un schéma scientifique ou une carte géographique.
  - **Industrie** : lire un plan technique ou une image satellite et donner une interprétation.
  - **E-commerce** : retrouver un produit à partir d'une photo + générer une fiche descriptive.

Multimodalité : texte + audio/vidéo

LLMs multimodaux : texte, image, son et vidéo

Les LLMs de nouvelle génération (ex. GPT-4o, Gemini, Claude 3) traitent plusieurs types de données.
Ils combinent **texte, image, audio et vidéo** pour offrir des usages encore plus riches.

- **Articulation multimodale**
  - **Audio encoder** → transforme la parole en vecteurs.
  - **Vision encoder** → encode images ou vidéos.
  - **LLM** → fusionne les représentations texte + image + son + vidéo.
  - **Sortie multimodale** → transcription, résumé, traduction, génération.

- **Exemples de modèles**
  - **Whisper + GPT-4o (OpenAI)** → transcription et traduction audio.
  - **Gemini (Google DeepMind)** → multimodal (texte, image, audio, vidéo).
  - **Claude 3 (Anthropic)** → analyse multimodale avec contexte long.

LLMs multimodaux : génération de vidéos 2D et 3D

- Les modèles d'IA générative ne se limitent plus aux textes et aux images fixes. Ils peuvent désormais **générer des vidéos** réalistes ou des environnements **3D immersifs** à partir de simples instructions textuelles.

- **Articulation multimodale**
  - **Texte → Embeddings** : le prompt est transformé en représentations numériques.
  - **Vision/Video decoder** : le modèle génère une séquence d'images (frames).
  - **Moteur 3D** : possibilité d'ajouter profondeur et mouvements pour une scène 3D.
  - **Sortie multimodale** : vidéo classique ou rendu 3D interactif.

- **Exemples de modèles**
  - **Runway Gen-2** → génération vidéo à partir de texte ou d'image.
  - **Pika Labs** → création de clips animés courts.
  - **Stable Video Diffusion (Stability AI)** → extension vidéo de Stable Diffusion.
  - **DreamFusion (Google Research)** → génération d'objets et scènes **3D** à partir de texte.

- **Cas d'usage**
  - **Création de contenu** : publicités, courts-métrages, effets visuels.
  - **Éducation** : visualiser un phénomène scientifique en 3D (molécule, système solaire).
  - **Architecture / Design** : générer un prototype d'espace 3D à partir d'une description textuelle.
  - **Gaming** : création d'assets 3D à partir d'un prompt pour accélérer le développement.

Impacts des LLMs

Impacts positifs

Les **LLMs** apportent des **bénéfices majeurs** dans de nombreux secteurs en augmentant la **productivité**, en facilitant l'**apprentissage**, et en accélérant la **recherche**.

- **Gains de productivité**
  - Rédaction automatisée de **mails, rapports, contrats, comptes rendus**.
  - Génération de **code** et assistance au développement logiciel.
  - Automatisation de tâches répétitives (résumés, notes de réunion, FAQ).
    Permet de libérer du temps pour des tâches à plus forte valeur ajoutée.

- **Soutien éducatif et formation**
  - **Tuteurs virtuels** personnalisés pour étudiants.
  - Explications adaptées au niveau de l'apprenant (primaire → universitaire).
  - Correction et **feedback instantané** sur des exercices.
    Démocratisation de l'accès au savoir, apprentissage plus interactif.

- **Accélération de la recherche et innovation**
  - Exploration et synthèse rapide de **littérature scientifique**.
  - Génération d'**hypothèses** et assistance dans la rédaction d'articles.
  - Support à l'**analyse de données complexes** (sciences, médecine, droit).
    Réduction drastique du temps entre la **question** et la **connaissance exploitable**.

Impacts sur les métiers

Les **LLMs** transforment profondément le monde du travail, en redéfinissant la **valeur ajoutée** des métiers existants et en créant de **nouvelles opportunités professionnelles**.

- **Automatisation des tâches répétitives**
  - Rédaction de mails, notes de synthèse, réponses standards au support client.
  - Analyse de documents longs (contrats, rapports, données brutes).
  - Génération de code ou de scripts pour des tâches simples.
    Réduction de la charge de travail sur les missions à faible valeur.

- **Déplacement de la valeur ajoutée**
  - Les professionnels passent moins de temps sur l'exécution, plus sur l'**analyse stratégique** et la **prise de décision**.
  - Exemples :
    - Juristes → se concentrent sur l'argumentation plutôt que sur la recherche documentaire.
    - Développeurs → se focalisent sur l'architecture et la sécurité plutôt que sur le code basique.
      L'humain garde un rôle central, mais sur des tâches à **fort impact**.

- **Nouveaux métiers émergents**
  - **Prompt Engineer** : spécialiste de la conception de requêtes efficaces pour LLMs.
  - **LLMOps Engineer** : déploiement, monitoring et optimisation des modèles.
  - **AI Trainer** : supervision de données et affinement des modèles.
    Une nouvelle économie des compétences se développe autour de l'IA générative.

Impacts sociétaux et culturels

Les **LLMs** transforment notre rapport au savoir, à la créativité et à l'information.
Ils offrent de nouvelles opportunités mais posent aussi des défis majeurs pour nos sociétés.

- **Nouvelles formes de créativité**
  - Génération de textes littéraires, poèmes, scénarios, chansons.
  - Outils pour designers, écrivains, journalistes, musiciens.
  - Collaboration homme–IA → co-création de contenus innovants.
    Favorise l'**exploration artistique** et l'**innovation culturelle**.

- **Accès démocratisé à la connaissance**
  - Savoirs complexes rendus accessibles via des explications en langage naturel.
  - Soutien scolaire personnalisé (tuteurs virtuels).
  - Traduction multilingue en temps réel → réduction des barrières linguistiques.
    Une **démocratisation sans précédent** de l'éducation et de la formation.

- **Questions éthiques et risques**
  - **Désinformation** : génération de fake news ou de contenus trompeurs.
  - **Biais** : reproduction et amplification de stéréotypes présents dans les données d'entraînement.
  - **Dépendance cognitive** : risque d'appauvrissement des compétences critiques humaines.
    Nécessité de mettre en place des **règles d'usage et une régulation**.

Limites techniques : hallucinations

Les **LLMs** peuvent générer des réponses **fausses mais convaincantes**.
C'est ce qu'on appelle des **hallucinations** : le modèle invente des faits ou des liens inexistants, tout en gardant un style crédible.

- **Nature du problème**
  - **Faits inventés** : le modèle génère une information qui n'existe pas dans la réalité.
  - **Erreurs factuelles** : il mélange des données correctes avec des approximations fausses.
  - **Persuasion trompeuse** : le texte est fluide, bien formulé → l'utilisateur croit que c'est vrai.

- **Exemples concrets**
  - Attribuer l'invention d'Internet à **Elon Musk** (au lieu de chercheurs des années 1960).
  - Inventer une **citation** d'un auteur qui n'a jamais été écrite.
  - Générer une **référence scientifique** (titre + auteur) qui semble authentique, mais qui n'existe pas.
  - Confondre deux personnalités aux noms proches (par ex. attribuer un prix Nobel au mauvais scientifique).

- **Causes**
  - Les LLMs ne "savent" pas → ils prédisent la suite de mots la plus probable.
  - Absence de **vérification factuelle interne**.
  - Biais liés au corpus d'entraînement (incomplet, bruité, contradictoire).

- **Conséquences**
  - Risque de **désinformation** dans des domaines sensibles (santé, droit, éducation).
  - Difficulté pour l'utilisateur de distinguer le vrai du faux.
  - Besoin de **méthodes de validation externe** (bases de données fiables, outils de fact-checking).

Limites techniques : hallucinations

Gérer les hallucinations : Détecter – Corriger – Prévenir

1. Détecter

- Demander au modèle de **justifier ses réponses** (*"cite tes sources"*).
- Poser la question de **plusieurs manières** pour vérifier la cohérence.
- Utiliser des outils de **fact-checking automatique** (Google Fact Check, CrossRef, etc.).

2. Corriger

- Confronter la sortie du LLM à des **bases de données fiables** (Wikipédia, PubMed, BOFiP, jurisprudence).
- Faire valider les réponses par un **expert humain** dans les domaines critiques (santé, droit, finance).
- Reformuler le prompt : *"Ne réponds que si tu es sûr, sinon dis que tu ne sais pas."*

3. Prévenir

- Coupler le LLM à une **base de connaissances externe** (RAG – Retrieval Augmented Generation).
- Limiter l'usage dans les **domaines sensibles** sans supervision.
- Former les utilisateurs à **critiquer et valider** les réponses.

Limites techniques : raisonnement et calcul

Le raisonnement des LLMs reste **instable et dépendant du contexte**.

- **Mathématiques complexes** : résultats approximatifs, surtout sur grands calculs.
- **Chaînes logiques** : étapes parfois contradictoires, cohérence globale fragile.
- **Dépendance au prompt** : formulation influence fortement la qualité de la réponse.
- Les LLMs **imitent** un raisonnement, mais ne garantissent pas une logique stricte.

Limites techniques : mémoire et contexte

Les LLMs ne possèdent pas de **mémoire humaine** : ils traitent uniquement le texte contenu dans leur **fenêtre de contexte**.
Au-delà de cette limite, ils "oublient" ou mélangent les informations.

- **Fenêtre de contexte**
  - Taille variable selon le modèle : **4K → 10M tokens** (ex. GPT-3.5 : 4K, GPT-4 Turbo : 128K, Gemini 1.5 : 1M).
  - Chaque token ≈ un mot ou un fragment de mot.
  - Plus la fenêtre est grande, plus le modèle peut gérer de longues conversations ou documents.

- **Perte d'informations**
  - Dans une **longue discussion**, les premières informations sortent de la fenêtre → oubli.
  - Le modèle peut alors :
    - Répéter une question déjà posée.
    - Donner une réponse contradictoire.
    - "Halluciner" pour combler un trou de mémoire.

- **Pas de mémoire persistante**
  - Une fois la conversation terminée, le modèle **n'a aucun souvenir**.
  - Pas de continuité naturelle d'une session à l'autre.
  - Toute mémoire longue durée doit être **externalisée** (base de données, systèmes RAG, embeddings).

- **Exemple concret**
  - Début de conversation : *"J'ai 3 enfants : Anna, Léa et Paul."*
  - 2000 tokens plus tard : *"Comment s'appelle mon fils ?"*
  - Le modèle peut oublier "Paul" si l'info est sortie de la fenêtre contextuelle.

Limites éthiques et sociales

Les **LLMs** ne sont pas neutres : ils reproduisent les biais et posent des questions majeures en matière de **justice, confidentialité et gouvernance**.

- **Biais culturels, raciaux, de genre**
  - Les modèles apprennent sur des données issues du web, contenant stéréotypes et discriminations.
  - Risque de produire des réponses **sexistes, racistes ou culturellement biaisées**.
  - Exemple : associer automatiquement certains métiers à un genre (“infirmière = femme”, “ingénieur = homme”).

- **Confidentialité des données**
  - Les corpus d’entraînement incluent parfois des données sensibles (mails, codes, forums).
  - Risque d’**exfiltration accidentelle** : le modèle peut régénérer du contenu confidentiel.
  - En entreprise : exposition de secrets industriels si les données ne sont pas filtrées.

- **Plagiat et droits d’auteur**
  - Les LLMs génèrent parfois des passages très proches de leurs données d’entraînement.
  - Problème de **propriété intellectuelle** : qui est l’auteur du texte produit ?
  - Cas sensibles dans la musique, le code ou les œuvres littéraires.

Exemple : Réponse discriminatoire due aux biais d’entraînement.

Limites économiques et écologiques

Les **LLMs** demandent des ressources massives pour leur **entraînement** et leur **utilisation**, avec des impacts financiers, environnementaux et sociaux.

- **Coûts financiers très élevés**
  - Entraîner un modèle de plusieurs centaines de milliards de paramètres = **plusieurs millions de dollars**.
  - Exemple : GPT-3 → coût estimé entre **5 et 12 millions $** pour l'entraînement.
  - Maintenance et déploiement (inférence en production) représentent aussi un coût continu.
    👉 Les petits laboratoires ou startups ne peuvent pas rivaliser.

- **Impact énergétique important**
  - Entraînement = consommation colossale de GPU/TPU → **émissions carbone élevées**.
  - Une seule session d'entraînement peut consommer autant d'énergie que **plusieurs centaines de foyers/an**.
  - L'utilisation quotidienne (milliards de requêtes) amplifie l'empreinte écologique.

- **Inégalités d'accès entre acteurs**
  - Seules les grandes entreprises (OpenAI, Google, Meta, Anthropic) disposent des ressources nécessaires.
  - Risque de **concentration du pouvoir technologique**.
  - Pays, PME, universités → dépendance aux géants du numérique.

- **Exemple**
  - **GPT-3** : plusieurs millions $ pour l'entraînement, sur des supercalculateurs avec **10 000+ GPU**.
  - Coût de déploiement continu pour maintenir un service comme **ChatGPT** utilisé par des centaines de millions d'utilisateurs.

Vers une IA plus efficiente

Après les LLMs géants, la tendance sera à l'efficacité. Les futurs modèles chercheront à consommer moins de ressources tout en restant performants.

- **IA frugale** : réduction de l'empreinte carbone et optimisation énergétique
- **Modèles spécialisés** : entraînés pour un domaine précis (médecine, droit, industrie)
- **Accessibilité** : déploiement possible sur ordinateurs portables, smartphones et IoT

Exemple : **Mistral 7B**, un modèle léger qui fonctionne sur un simple laptop avec GPU et rivalise avec des modèles plus grands.

L’IA multimodale

L’avenir est à des modèles capables de traiter simultanément plusieurs types de données pour mieux comprendre le monde.

- **Fusion de modalités** : texte + image + audio + vidéo + signaux capteurs
- **Applications** : diagnostic médical, robots d’assistance, outils pédagogiques interactifs
- **Avantage** : meilleure compréhension contextuelle et capacités de raisonnement plus proches de l’humain

Exemple : **GPT-4o (OpenAI)** analyse une photo, répond à une question orale et génère du texte en temps réel.

L’IA agentique et autonome

Les modèles deviendront des **agents intelligents**, capables non seulement de générer du contenu, mais aussi d’agir et d’interagir dans des environnements complexes.

- **Planification autonome** : découpage d’objectifs en sous-tâches
- **Connexion à des systèmes externes** : bases de données, API, applications métiers
- **Collaboration multi-agents** : plusieurs IA travaillant ensemble sur un projet

Exemple : **AutoGPT** ou **LangChain agents**, capables de lancer une recherche web, analyser les résultats et rédiger un rapport automatiquement.

L’IA intégrée et personnelle

Demain, chacun pourra avoir son **assistant IA personnel**, intégré dans son quotidien et adapté à ses besoins.

- **Personnalisation extrême** : mémoire longue durée, préférences et style de communication
- **Applications pratiques** : coaching santé, aide à l’apprentissage, support créatif
- **Confidentialité renforcée** : exécution locale (edge computing) et clouds souverains

Exemple : **Lunettes intelligentes avec IA** qui traduisent en direct une conversation et affichent la traduction dans le champ de vision.

Vers l'IA Générale ?

À plus long terme, l'objectif est l'**IA Générale** (AGI), une intelligence artificielle polyvalente, flexible et autonome.

- **Capacité à raisonner** comme un humain, dans des domaines variés
- **Apprentissage rapide** : adaptation à de nouvelles tâches sans réentraînement massif
- **Autonomie** : définition et poursuite d'objectifs de manière indépendante

Exemple prospectif : un **copilote scientifique** qui propose une hypothèse, conçoit une expérience, en analyse les résultats et en tire de nouvelles connaissances.

Atelier

- Mettre en pratique les notions vues dans le cours.
- Analyser des cas concrets d’usage des LLMs.
- Développer un regard critique et responsable.
- Travailler en groupe pour confronter les points de vue.

Méthodologie de l'atelier

- **Groupes** : 3–4 étudiants
- **Mission** : Analyser un cas d'usage attribué (santé, droit, éducation, industrie)
- **Dimensions à traiter** :
  - **Approche technique** : type de modèle, données, adaptation (fine-tuning, RAG), déploiement, ressources nécessaires.
  - **Impacts** : pour les utilisateurs finaux, les organisations, la société (métiers, économie, accessibilité).
  - **Limites & risques** : hallucinations, biais, dépendance, coûts, empreinte écologique.
  - **Bonnes pratiques** : comment l'outil doit être utilisé, rôle de l'humain.
  - **Précautions** : vérifications, garde-fous, fact-checking, supervision.
  - **Formations nécessaires** : compétences techniques (prompting, RAG), esprit critique, sensibilisation aux enjeux éthiques.
- **Restitution** : 5 min par groupe (présentation synthétique + discussion collective).

Cas d’usage à analyser

- **Journalisme automatisé et fact-checking**
  Un LLM multimodal génère automatiquement des articles et vérifie les faits en temps réel pour la presse en ligne.

- **Compagnons émotionnels et psychologiques**
  Des assistants IA empathiques accompagnent des personnes isolées via le texte et la voix pour offrir un soutien quotidien.

- **Éducation personnalisée assistée par IA**
  Un tuteur virtuel sur ordinateur guide les élèves dans leurs exercices, corrige leurs réponses et propose des explications adaptées.

- **Analyse automatisée de dossiers médicaux**
  Un hôpital utilise un LLM pour résumer et croiser les informations contenues dans des dossiers patients volumineux.

- **Conseiller fiscal ou bancaire intelligent**
  Un assistant IA aide particuliers et entreprises à remplir leurs déclarations fiscales et à gérer leurs finances courantes.

- **Recrutement et gestion RH automatisés**
  Une IA analyse CV, lettres de motivation et entretiens pour assister les recruteurs dans la présélection des candidats.