LE PALMARÈS DES MODÈLES D'INTELLIGENCE ARTIFICIELLE

Vence le 11 mai 2025

Auteur : Gilbert Brault guidant : Gemini PRO 2.5 et GPT-4.5

Introduction

L'intelligence artificielle (IA) est sur toutes les lèvres, transformant notre façon de travailler, de créer et d'interagir. Au cœur de cette révolution se trouvent des "modèles" complexes, sortes de cerveaux numériques entraînés pour comprendre et générer du texte, des images, et bien plus encore. Ce document présente un aperçu comparatif des modèles d'IA les plus en vue, leurs caractéristiques principales, et une tentative d'évaluation de leurs capacités relatives, notamment à travers le prisme d'autres IA avancées.

Il est important de noter que les évaluations chiffrées présentées ci-dessous (colonnes "Note GPT-4.5" et "Note Gemini 2.5 Pro") sont des estimations générées par les IA 'GPT-4.5' et une version de 'Gemini 2.5 Pro' . Elles reflètent leur 'perception' interprétative des capacités relatives de ces modèles à partir des informations disponibles et ne constituent pas des benchmarks objectifs absolus.

Tableau Comparatif des Modèles d'IA Majeurs

Nom du Modèle	Société Développeuse	Date d'introduction	Description	Technologie	Note GPT-4.5	Note Gemini 2.5 Pro
Claude	Anthropic	Mars 2023	Modèle conversationnel connu pour sa sécurité et son alignement avec les valeurs humaines.	Transformers (Decoder-only), RLHF, Constitutional AI	8.5	8.7
PaLM	Google	Avril 2022	Grand modèle linguistique pour la compréhension et la génération de texte à grande échelle.	Transformers (Encoder-Decoder, puis Pathways)	8.0	8.5
Gemma	Google	Février 2024	Famille de modèles de langage open-source, légers et performants, conçus pour être accessibles et efficaces.	Transformers (Decoder-only, basé sur Gemini), Open-source	7.5	8.2
Gemini	Google DeepMind	Décembre 2023	Modèle multimodal performant capable de gérer texte, image, et interactions complexes.	Transformers multimodaux (Texte, Image, Audio, Vidéo, Code), RLHF, potentiellement MoE (pour Ultra)	9.0	9.6
Watsonx	IBM	Mai 2023	Plateforme d'IA pour l'entreprise avec outils et modèles spécialisés pour diverses applications commerciales.	Plateforme IA (modèles Transformers, outils MLOps, cloud IA)	7.5	7.5
Granite	IBM	Novembre 2023	Famille de modèles open-source conçus pour des applications d'entreprise, incluant génération de code et traitement multimodal.	Transformers (Decoder-only, potentiellement multimodal pour certaines versions), Open-source	8.0	7.8
Llama	Meta AI	Février 2023	Modèle open-source destiné à favoriser la recherche en IA et la génération de texte.	Transformers (Decoder-only), Open-source	8.5	8.6
Mistral	Mistral AI	Septembre 2023	Modèle open-source optimisé pour la génération de texte rapide et efficace.	Transformers (Decoder-only, GQA, SWA), Open-source	8.0	8.3
GPT-3	OpenAI	Juin 2020	Modèle avancé de génération de texte	Transformers (Decoder-only)	8.5	8.0

Nom du Modèle	Société Développeuse	Date d'introduction	Description	Technologie	Note GPT-4.5	Note Gemini 2.5 Pro
			basé sur les transformers, polyvalent et précurseur de nombreux usages actuels.			
GPT-4	OpenAI	Mars 2023	Version améliorée de GPT-3 avec des capacités multimodales (texte et images) et une compréhension contextuelle accrue.	Transformers multimodaux (Texte, Image), RLHF, potentiellement MoE	9.5	9.3
ChatGPT	OpenAI	Novembre 2022	Version conversationnelle du modèle GPT optimisée pour le dialogue humain.	Transformers (Decoder-only, basé sur GPT-3.5/4), RLHF	9.0	9.1
T-NLG	Microsoft	Février 2020	Modèle avancé de génération de texte à grande échelle optimisé par DeepSpeed et ZeRO.	Transformers (optimisé avec DeepSpeed & ZeRO)	7.5	7.4
MT-NLG	Microsoft/NVIDIA	Octobre 2021	Modèle très large, combinant technologie Megatron (NVIDIA) et DeepSpeed (Microsoft), optimisé pour des tâches complexes.	Transformers (optimisé avec Megatron & DeepSpeed)	8.5	8.4
DeepSeek	Équipe de chercheurs	Septembre 2023	Modèle axé sur la génération de texte, souvent utilisé dans des contextes académiques et de recherche.	Transformers (Decoder-only), Open-source (pour certaines versions/recherches)	7.0	7.2

Explications des Technologies mentionnées

Les modèles d'intelligence artificielle peuvent être regroupés selon leurs architectures et les techniques employées. Les modèles **"Decoder-only"** (comme GPT-3, GPT-4, ChatGPT, Llama, Mistral) sont principalement orientés vers la génération continue de texte, prédisant chaque nouveau mot en fonction du contexte précédent. À l'inverse, les modèles **"Encoder-Decoder"** (comme PaLM initialement) utilisent une architecture à deux parties : l'encodeur traite et comprend les informations d'entrée, tandis que le décodeur génère une réponse appropriée. Enfin, les modèles **"multimodaux"** (comme Gemini et GPT-4) sont capables de traiter simultanément différents types d'informations (texte, image, audio, vidéo), les rendant extrêmement polyvalents.

Voici quelques termes clés pour mieux comprendre ces technologies :

Architectures Fondamentales et Types de Modèles :

- **Transformers** : C'est l'architecture de base, le "moteur" derrière la plupart de ces IA, permettant aux machines de comprendre le langage en prêtant attention à l'importance des différents mots dans une phrase.

  « Transformers for Natural Language Processing » (Rothman, 2021)

  > *« Les architectures Transformer surpassent largement les approches antérieures grâce à leur mécanisme d'attention, permettant de traiter efficacement des contextes étendus » (Rothman, 2021).*

- **Decoder-only** : Type d'architecture Transformer, spécialisé pour générer du texte ; il prédit chaque mot successivement pour former des phrases et des paragraphes.
- **Encoder-Decoder** : Type d'architecture Transformer idéal pour des tâches comme la traduction ou le résumé ; il analyse (encode) l'entrée puis génère (decode) une sortie.
- **Multimodal** : Capacité d'un modèle à traiter et comprendre plusieurs types d'informations (texte, image, audio, etc.) simultanément.

Techniques d'Entraînement et d'Alignement :

- **RLHF (Apprentissage par renforcement à partir de retours humains)** : Méthode d'entraînement où des humains évaluent les réponses de l'IA, permettant au modèle d'affiner ses réponses pour être plus utile, honnête et inoffensif. C'est comme éduquer un enfant en lui disant ce qui est bien ou pas.
- **Constitutional AI (IA Constitutionnelle)** : Approche (notamment utilisée par Anthropic pour Claude) où un ensemble de règles ou principes ("constitution") guide le comportement de l'IA pour assurer des réponses sûres et éthiques.

  « GPT-3 and the Future of AI) » (Wolfram, 2021)

  > *« L'apprentissage par renforcement à partir de retours humains (RLHF) permet aux modèles de produire des réponses alignées sur les attentes humaines, réduisant ainsi les biais et les erreurs fréquentes des modèles non supervisés » (Wolfram, 2021).*

Concepts Avancés et Optimisations :

- **MoE (Mélange d'Experts)** : Architecture où, au lieu d'un seul grand réseau neuronal, plusieurs "sous-réseaux" plus petits et spécialisés (les "experts") collaborent. Seuls les experts pertinents sont activés pour une tâche donnée, rendant les très grands modèles plus efficaces.
- **GQA (Grouped-Query Attention) & SWA (Sliding Window Attention)** : Techniques d'optimisation de l'attention dans les Transformers (utilisées par Mistral, par exemple) pour améliorer la vitesse et l'efficacité, notamment pour traiter de longs contextes.

  « Attention Is All You Need (annoté) » (Vaswani et al., 2017)

  > *« Le mécanisme d'attention multi-têtes permet à un modèle d'encoder simultanément différents types de relations contextuelles, rendant le traitement du langage naturel plus efficace » (Vaswani et al., 2017).*

- **Context (Contexte)** : dans les modèles d'IA basés sur les Transformers, le contexte désigne l'ensemble des informations précédant immédiatement le mot ou la phrase que le modèle tente de générer ou de prédire. Plus le contexte pris en compte par le modèle est large, plus ses prédictions sont précises et adaptées.

Modes de Diffusion et Plateformes :

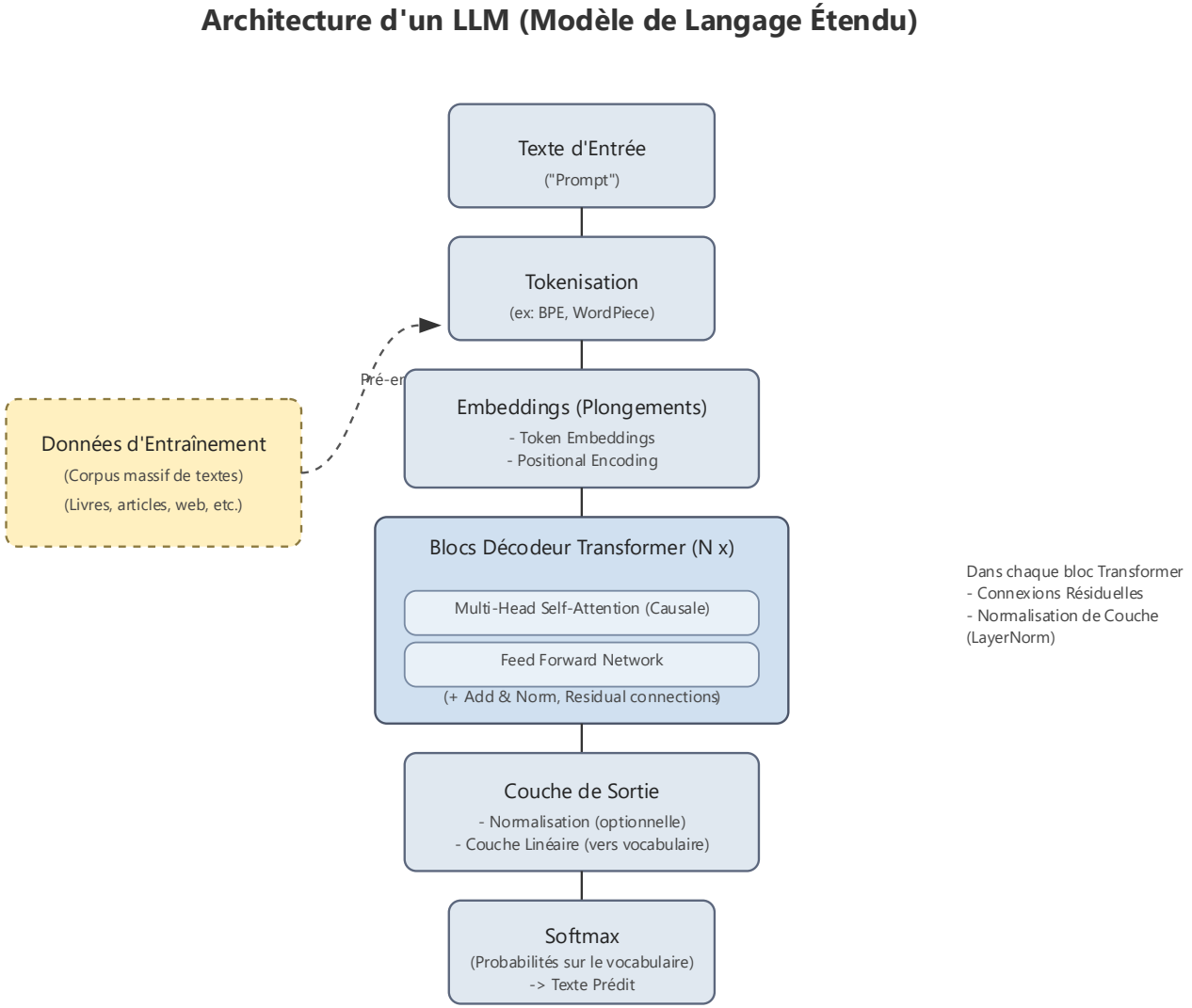
- **Open-source** : Le code source du modèle (ou ses poids) est rendu publiquement accessible, permettant à quiconque de l'utiliser, de le modifier et de le distribuer. Cela favorise l'innovation et la transparence.
- **Plateforme IA** : Un écosystème plus large qui inclut non seulement des modèles d'IA, mais aussi des outils pour le développement, le déploiement, la gestion et la gouvernance d'applications d'IA (souvent orienté entreprise, comme Watsonx).

Conclusion

Le monde des modèles d'IA est en ébullition constante, et ce palmarès n'est qu'un instantané d'un domaine qui évolue à une vitesse fulgurante. Les dates d'introduction indiquées sont importantes, mais de nouvelles versions et des modèles concurrents émergent régulièrement, redéfinissant les capacités et les performances. Il est donc crucial de rester informé des derniers développements.

Une chose est sûre : la compétition pour créer l'IA la plus intelligente, la plus utile, la plus efficace et la plus responsable est loin d'être terminée, promettant un avenir où l'intelligence artificielle jouera un rôle encore plus central et transformateur dans nos vies.

Architecture d’un LLM



1. **Données d'Entraînement** (Training Data)
   Ce bloc représente la fondation sur laquelle le LLM est "pré-entraîné". Il s'agit d'un corpus textuel absolument colossal (des milliards voire des trillions de mots) provenant de diverses sources comme des livres, des articles, des sites web, du code, etc. Durant cette phase d'apprentissage non supervisé, le modèle apprend les patrons linguistiques, la grammaire, les connaissances factuelles, les styles d'écriture et les relations sémantiques entre les mots. L'objectif est typiquement de prédire le mot suivant dans une phrase ou de combler des mots masqués, ce qui force le modèle à internaliser la structure et le sens du langage.

2. **Texte d'Entrée** ("Prompt")
   Une fois le LLM pré-entraîné, le "Texte d'Entrée", souvent appelé "prompt", est la séquence de mots que l'utilisateur fournit au modèle pour initier une interaction ou demander une tâche spécifique. Cela peut être une question ("Quelle est la capitale de la France ?"), une instruction ("Écris un poème sur l'océan"), une phrase à compléter ("Le meilleur moyen de voyager est..."), ou tout autre fragment de texte. C'est le point de départ à partir duquel le modèle va générer une réponse ou **une continuation.**

3. **Tokenisation**
   La tokenisation est le processus de conversion du texte d'entrée brut en une séquence de "tokens" (jetons). Ces tokens peuvent être des mots entiers, des sous-mots (par exemple, "indécomposable" pourrait devenir "in-décomp-osable") ou même des caractères, selon la méthode utilisée (comme BPE - Byte Pair Encoding, ou WordPiece). Chaque token est ensuite mappé à un identifiant numérique unique. Cette étape est cruciale car les modèles ne traitent pas directement le texte, mais des représentations numériques de ces tokens. Elle permet de gérer un vocabulaire vaste et de traiter des mots rares ou inconnus en les décomposant.

4. **Embeddings** (Plongements)
   Après la tokenisation, chaque token numérique est transformé en un vecteur dense de nombres réels, appelé "embedding" ou "plongement lexical". Ces vecteurs capturent le sens sémantique des tokens : des tokens ayant des significations similaires auront des vecteurs proches dans cet espace vectoriel. En plus des Token Embeddings, on ajoute généralement un Positional Encoding (Encodage Positionnel). C'est un vecteur qui fournit au modèle des informations sur la position de chaque token dans la séquence, car l'architecture Transformer elle-même ne traite pas l'ordre séquentiel des tokens de manière inhérente. La somme de ces deux types de plongements est ensuite fournie au bloc Transformer.

   « Deep Learning avec Python » (Chollet, 2018)

   > « Les plongements (embeddings) transforment les données symboliques en représentations vectorielles continues, facilitant ainsi la capture de relations sémantiques fines par les réseaux neuronaux » (Chollet, 2018).

5. **Blocs Décodeur Transformer** (N x)
   C'est le cœur computationnel du LLM, composé de multiples couches (N blocs) identiques empilées. La plupart des LLM modernes utilisent une architecture de type "décodeur-seulement" (decoder-only) de Transformer. Chaque bloc décodeur contient typiquement deux sous-couches principales :

   a. **Multi-Head Self-Attention** (Causale/Masquée) : Permet à chaque token de "regarder" et de pondérer l'importance des autres tokens précédents dans la séquence (d'où "causale" ou "masquée", pour ne pas voir les tokens futurs lors de la génération). Cela aide le modèle à comprendre le contexte. "Multi-Head" signifie que ce processus d'attention est effectué plusieurs fois en parallèle avec différentes projections des vecteurs, capturant ainsi différents types de relations contextuelles.

   b. **Feed Forward Network** (Réseau de Neurones à Propagation Avant) : Un réseau de neurones simple (typiquement deux couches linéaires avec une fonction d'activation non linéaire) qui traite la sortie de la couche d'attention pour chaque token indépendamment.
      Autour de ces deux sous-couches, on trouve des connexions résiduelles (Add) et des couches de normalisation (Norm, souvent Layer Normalization) pour stabiliser l'entraînement et permettre la construction de modèles très profonds.

6. **Couche de Sortie** (Output Layer)
   Après le passage à travers tous les blocs Transformer, la représentation vectorielle du dernier token (ou de tous les tokens, selon l'objectif) est traitée par une couche de sortie. Cette couche consiste généralement en une normalisation

optionnelle suivie d'une couche linéaire (une transformation affine). Le rôle principal de cette couche linéaire est de projeter le vecteur de sortie du dernier bloc Transformer (qui a une dimension interne au modèle, par exemple 4096) vers un vecteur dont la dimension correspond à la taille du vocabulaire du modèle (par exemple, 50000 tokens). Les valeurs de ce vecteur sont appelées "logits".

7. **Softmax / Prédiction**
   La fonction Softmax est appliquée aux logits produits par la couche de sortie. Elle transforme ces scores bruts en une distribution de probabilités sur l'ensemble du vocabulaire. Chaque token du vocabulaire se voit ainsi attribuer une probabilité d'être le prochain token de la séquence. Le "Texte Prédit" est ensuite généré en sélectionnant un token basé sur ces probabilités. La méthode la plus simple est de choisir le token avec la plus haute probabilité (décodage "greedy"), mais des techniques d'échantillonnage plus sophistiquées (comme le top-k sampling, nucleus sampling) sont souvent utilisées pour introduire de la diversité et de la créativité dans la génération. Ce processus est répété de manière auto-régressive : le token prédit est ajouté à la séquence d'entrée, et le modèle prédit le token suivant, et ainsi de suite, jusqu'à ce qu'un token spécial de fin de séquence soit généré ou qu'une longueur maximale soit atteinte.

Acronymes

Voici la liste des acronymes et leur définition :

1. **IA** : Intelligence Artificielle
2. **GPT** : Generative Pre-trained Transformer (modèle pré-entrainé génératif basé sur les Transformers)
3. **RLHF** : Reinforcement Learning with Human Feedback (Apprentissage par renforcement à partir de retours humains)
4. **MoE** : Mixture of Experts (Mélange d'experts)
5. **GQA** : Grouped-Query Attention (Attention par requêtes groupées)
6. **SWA** : Sliding Window Attention (Attention par fenêtre glissante)
7. **MLOps** : Machine Learning Operations (Opérations d’apprentissage automatique)

Les noms de modèles et technologies connus par leur abréviation :

- **PaLM** : Pathways Language Model

- **Gemini** : Modèle d'intelligence artificielle multimodal (nom propre non décomposé explicitement)
- **T-NLG** : Turing Natural Language Generation
- **MT-NLG** : Megatron-Turing Natural Language Generation