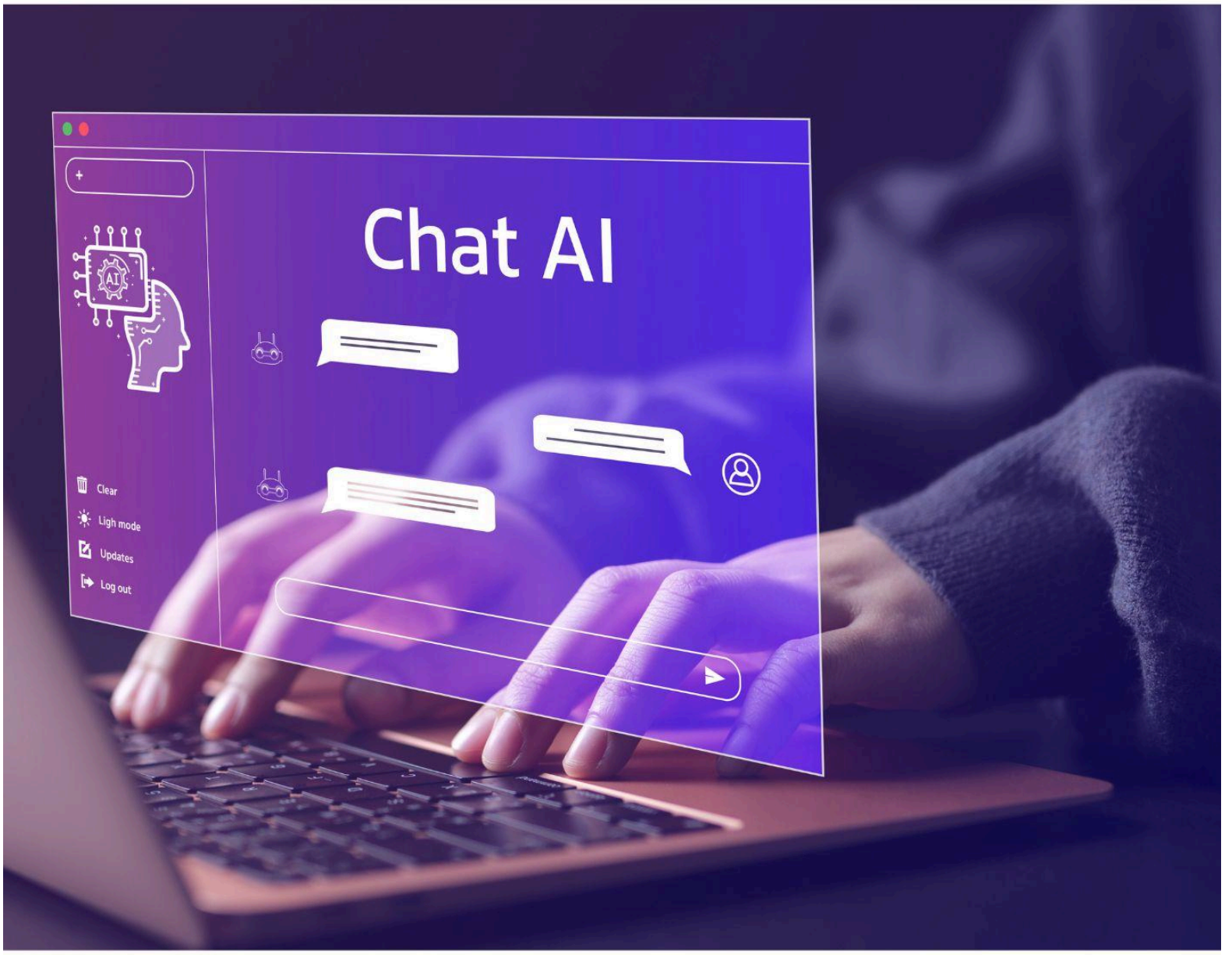
L'intelligence artificielle

Les modèles génératifs de langage



Niveau intermédiaire

EPN
Bibliothèque
de Nivelles

Ce support a été en partie rédigé avec l'aide de l'intelligence artificielle. Toutes les captures d'écran présentes dans ce document sont utilisées à des fins pédagogiques, sans but commercial.

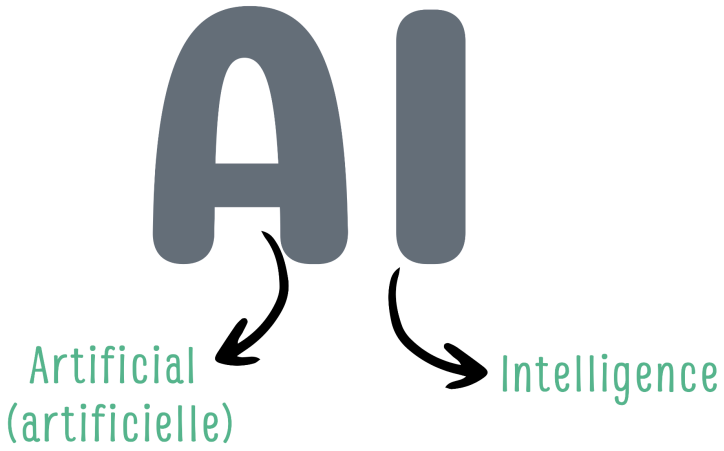
Introduction

Les modèles génératifs de langage, comme ChatGPT, transforment notre manière d'interagir avec la technologie. Capables de produire du texte, de répondre à des questions, de résumer, traduire ou même coder, ils ouvrent de nouvelles perspectives dans de nombreux domaines. Ce support vous permet de découvrir une sélection de modèles de langage et propose une première exploration de leur fonctionnement.

Lexique

- **IA : intelligence artificielle :**

C'est un ensemble de techniques permettant à des machines de simuler l'intelligence humaine (reconnaissance vocale, traduction, analyse de données…).



Exemples : AI génératives, AI prédictives, AI descriptives...

- **IA générative :**

L'intelligence artificielle générative est une branche de l'IA capable de créer du contenu : texte, images, musique, ou même vidéos.

- ## Modèles génératifs de langage :

Ce sont des programmes d'intelligence artificielle capables de comprendre et de produire du texte de manière autonome. Ils sont entraînés sur d'énormes volumes de données et utilisent des algorithmes avancés pour prédire et générer du texte cohérent en fonction d'une requête donnée. Parmi eux, on trouve des modèles comme ChatGPT, qui peuvent répondre à des questions, rédiger des textes ou assister dans diverses tâches. Leur fonctionnement repose sur des réseaux neuronaux et des techniques comme l'apprentissage profond. Il peut être simple ou avancé.

- ## Prompt :

C’est une consigne, une instruction que l’on donne à l’IA pour qu’elle réalise une tâche. C'est comme une commande qui guide l'IA sur ce qu'elle doit faire. Pour bien utiliser ChatGPT, il faut savoir rédiger de bons prompts, cela s’appelle le prompt engineering. Le prompt engineering (ou ingénierie de prompt) consiste à rédiger et ajuster les instructions données à une intelligence artificielle pour obtenir les réponses les plus précises et utiles possibles.

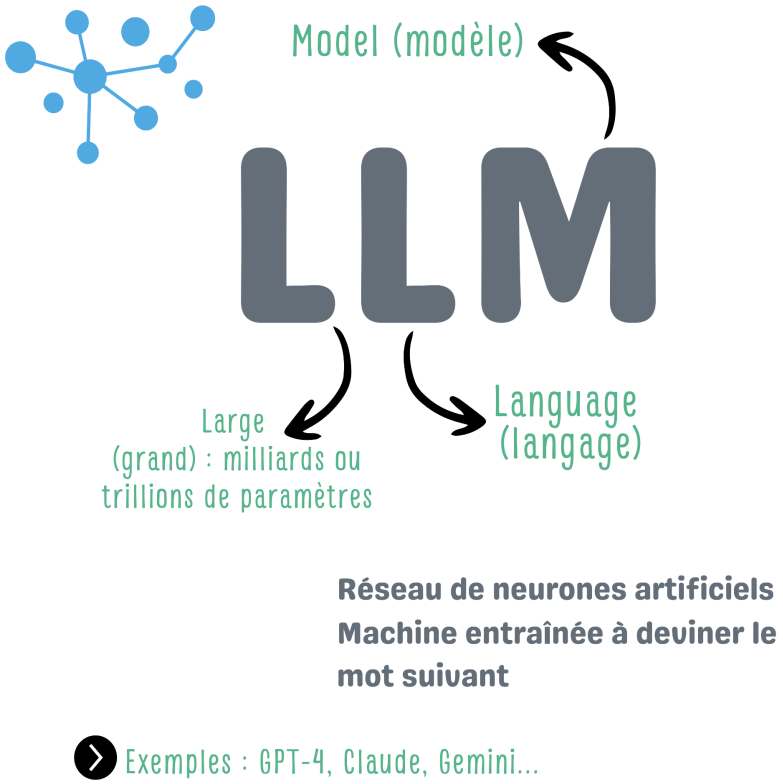
C'est l'art de poser les bonnes questions ou de formuler les bonnes demandes pour bien guider l'IA !

- ## LLM :

Large **L**anguage **M**odel

Modèle de langage capable de comprendre et de générer du texte. Il répond aux prompts en utilisant des milliards de mots sur lesquels il a été entraîné. Il ne s’agit pas d’une intelligence qui pense comme un humain, c’est de la prédiction statistique.

Exemple de LLM : GPT-4



- **Omnimodaux :**

Les modèles génératifs de langage omnimodaux sont des IA capables de comprendre et de générer différents types de contenu : texte, images, audio, vidéo, etc.

Contrairement aux modèles classiques qui traitent un seul type de données, ces modèles peuvent analyser et mélanger plusieurs formats à la fois, comme décrire une image en texte ou générer une image à partir d'une description.

- **Token :**

Brique de texte utilisée par les modèles de langage. Les IA utilisent ces token pour comprendre et générer du texte. Un **token** dans un **modèle de langage** est une petite unité de texte. Ça peut être un mot entier, une partie de mot ou même un signe de ponctuation.

Par exemple :

Le mot **"ordinateur"** pourrait être un seul token.

Une phrase comme **"C'est facile !"** pourrait être divisée en plusieurs tokens : C', est, facile, !.

Les **LLM** lisent et génèrent du texte en manipulant ces tokens, un peu comme des briques pour construire des phrases et des idées. Un token est une "brique" de texte que le modèle comprend et traite.

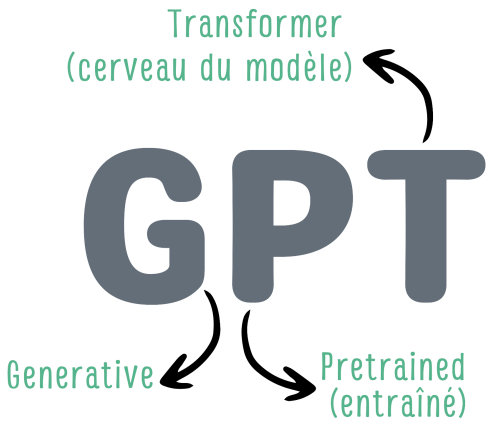
- **La fenêtre de contexte :**

Il s'agit de la quantité d'informations que l'IA peut traiter en une seule fois. Si c'est une conversation trop longue, l'IA peut oublier ce qui a été dit au début. Plus la fenêtre de contexte est grande, plus l'IA peut se souvenir de ce qui a été dit plus tôt dans l'échange et fournir des réponses cohérentes.

- **Les transformers :**

Les transformers sont un type de modèle d'intelligence artificielle conçu pour traiter de grandes quantités de texte. Ils fonctionnent grâce à un mécanisme appelé auto-attention, qui permet au modèle de comprendre le contexte global d'une phrase en tenant compte de chaque mot et de sa relation avec les autres.

- **ChatGPT :**

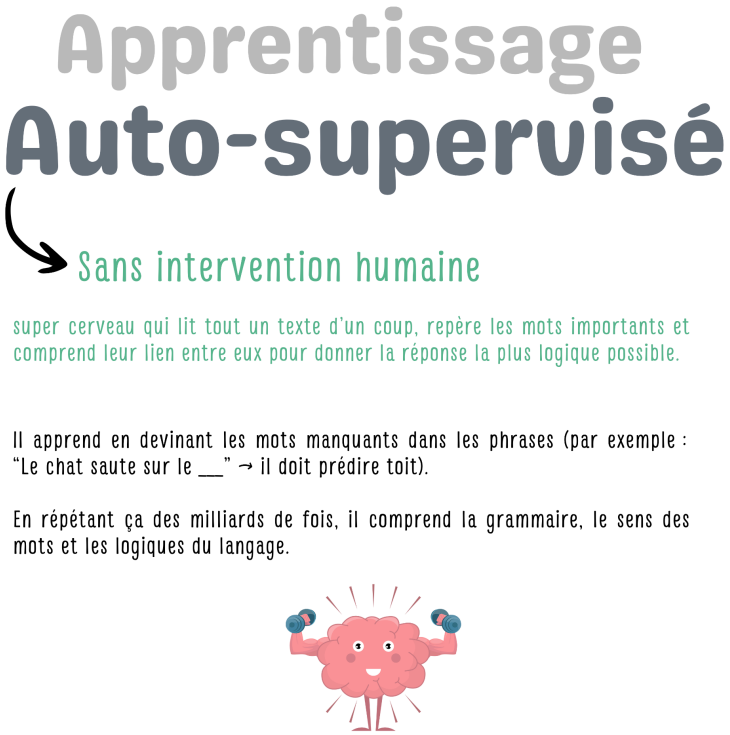


- **Apprentissage auto-supervisé :**

Ces modèles de langage ont été entraînés sur des milliards de phrases, d'images et de données diverses. L'apprentissage auto supervisé est une méthode d'intelligence artificielle où un modèle apprend à partir de données non étiquetées en générant ses propres "questions" et "réponses" pendant l'entraînement.

Par exemple, il peut cacher une partie d'un texte et essayer de la deviner, s'entraînant ainsi sans aide humaine.

Lors de l'entraînement, le modèle est exposé à une immense quantité de données textuelles (livres, articles, Internet…) et cela sans intervention humaine → auto-supervisé. Il est capable de prédire les mots manquants dans un texte incomplet. Et ce processus est répété des milliards de fois, pour affiner la génération de phrases logiques et fluides.

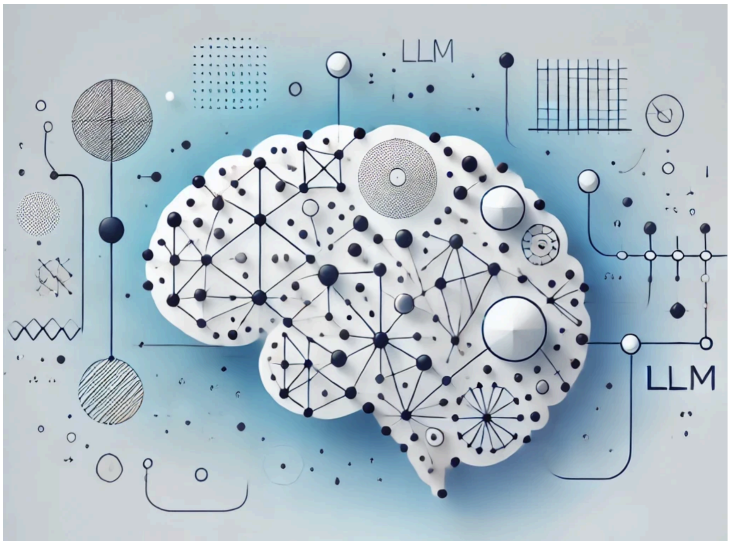


- **Embedding :**

Un embedding, c'est comme un code secret pour des mots ou des objets. Au lieu de traiter des mots sous forme de lettres, l'IA transforme chaque mot en un ensemble de chiffres qui capture sa signification. Ces chiffres permettent à l'IA de mieux comprendre les mots et de voir quels mots sont proches ou liés entre eux, un peu comme si elle pouvait reconnaître des mots qui ont un sens similaire, même s'ils sont écrits différemment.

C'est une façon pour l'IA de travailler plus facilement avec des données, comme des mots, en les transformant en quelque chose qu'elle peut mieux analyser.

Comment fonctionne un LLM ?



Un LLM est un réseau de neurones artificiels (chiffres et paramètres interconnectés) comme notre cerveau qui lui est un ensemble de cellules cérébrales interconnectées.

C’est une machine qui sait deviner le mot suivant.

Un LLM peut avoir des milliards, même des trillions de paramètres, c’est pour cela qu’ils sont appelés **“Large”** language model. Tous les contenus peuvent être transformés en chiffres, puis en texte. C’est l’entraînement des modèles de langage qui permettent de définir les chiffres.

L’entraînement :

Le modèle est alimenté pendant une période d’entraînement avec une très grande quantité de texte. Il va s’entraîner à deviner le mot suivant jusqu’à ce qu’il devienne vraiment excellent.

Les étapes d'entraînement d'un modèle de langage

1. Collecte et préparation des données

- Récupération de vastes ensembles de textes issus de sources variées (livres, articles, sites web, dialogues, etc.).
- Nettoyage des données : suppression des doublons, des erreurs et des informations non pertinentes.

👉 **Objectif :** Constituer un corpus de données représentatif et de qualité.

2. Tokenisation et vectorisation des données

- **Tokenisation** : le texte est découpé en petites unités appelées "tokens" (mots, sous-mots ou caractères).
- **Vectorisation** : chaque token est converti en une représentation numérique sous forme de vecteurs mathématiques.

👉 **Objectif :** Transformer le texte en données exploitables par un modèle d'intelligence artificielle.

3. Choix de l'architecture du modèle

Une fois les données prêtes, il faut définir la structure du modèle en fonction des besoins :

- **Par exemple : Transformers (ex : GPT, BERT)** : modèles avancés utilisant des mécanismes d'attention pour mieux comprendre le contexte et générer du texte plus pertinent.

👉 **Objectif :** Sélectionner une architecture adaptée aux exigences de la tâche.

4. Entraînement du modèle

Le modèle est entraîné en analysant des millions d'exemples pour apprendre à prédire le mot suivant ou compléter des phrases.

- Utilisation d'algorithmes d'optimisation pour ajuster ses paramètres et minimiser les erreurs.
- Apprentissage basé sur des milliards de calculs effectués par des réseaux de neurones.

👉 **Objectif :** Permettre au modèle d'apprendre les structures et les relations linguistiques.

5. Ajustement et optimisation

Une fois l'entraînement terminé, le modèle est testé et amélioré en fonction de ses performances :

- Ajustement des pondérations pour éviter les biais.
- Validation sur des jeux de données indépendants.
- Affinage par apprentissage supervisé (correction manuelle de certaines réponses).

👉 **Objectif :** Améliorer la fiabilité et la précision du modèle.

6. Déploiement et mises à jour

Une fois opérationnel, le modèle est intégré dans des applications (assistants vocaux, chatbots, moteurs de recherche, etc.).

- Il peut être mis à jour régulièrement pour s'adapter aux nouvelles données.
- Il peut aussi s'améliorer grâce aux interactions des utilisateurs.

👉 **Objectif :** Assurer une utilisation concrète et évolutive du modèle dans un cadre applicatif.

Un **vecteur numérique** utilisé dans les modèles de langage est une représentation mathématique d'un mot, d'une phrase ou d'un texte sous forme d'une liste de nombres. Ces nombres capturent des caractéristiques du mot, comme son sens, son contexte ou sa relation avec d'autres mots.

Exemple simple de vecteur numérique

Si l'on prend le mot "chat", il pourrait être représenté par un vecteur de ce type (avec des valeurs arbitraires) :

[0.12, -1.34, 2.45, 0.98, -0.67, 1.22, -0.54, 0.39, -0.88, 1.05]

Chaque nombre dans cette liste représente une **dimension** qui encode une information sur le mot. Par exemple, une dimension peut capturer s'il s'agit d'un animal, une autre peut représenter son lien avec le mot "chien", etc.

Exemple de relations entre mots dans l'espace vectoriel

Les mots ayant un sens proche ont des vecteurs similaires. Par exemple, si on représente "chat" et "chien" dans un espace à 2 dimensions (simplifié) :

- **Chat** → [0.9, 1.2]

- **Chien** → [1.0, 1.3]
- **Voiture** → [5.4, -2.1]

On voit que "chat" et "chien" ont des valeurs proches, tandis que "voiture" est plus éloigné, car son sens est très différent.

Utilisation des vecteurs :

Ces vecteurs permettent au modèle d'effectuer des calculs pour comprendre le sens des mots et leurs relations. Par exemple, des modèles comme **Word2Vec** ou **GPT** utilisent ces représentations pour effectuer des analogies :

Vecteur("Roi") - Vecteur("Homme") + Vecteur("Femme") ≈ Vecteur("Reine")

Cela signifie que le modèle peut comprendre que "roi" est à "homme" ce que "reine" est à "femme".

💡 **En résumé :** Un vecteur numérique est une liste de nombres qui encode le sens et le contexte d'un mot, permettant aux modèles d'intelligence artificielle de traiter et de comprendre le langage.

Rétropropagation :

La rétropropagation est une méthode utilisée pour entraîner les réseaux de neurones en intelligence artificielle.

Concrètement, après avoir fait une prédiction, le modèle compare son résultat à la réponse attendue, calcule l'erreur, puis ajuste ses connexions en remontant étape par étape pour améliorer ses futures prédictions. C'est un peu comme apprendre de ses erreurs.

Pour être fonctionnelle, le modèle doit subir un entraînement humain. C'est l'apprentissage par renforcement avec feedback (RLMF : reinforcement Learning with Human Feedback). Cet apprentissage est conçu par des humains mais cependant, une fois ces éléments mis en place, l'IA va s'entraîner seule en fonction des feedbacks reçus pendant l'apprentissage.

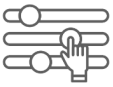
C'est pour cette raison que chatGPT ne nous donnera pas d'information sur un sujet illégal. Il sait très bien quelle est la réponse, mais grâce à l'entraînement, il sait que l'on ne doit pas commettre d'actes illégaux.

Une fois l'entraînement terminé, il est figé. On peut lui apporter quelques ajustements plus tard.

Chat GPT a été entraîné avec différents types de données : livres, articles, journaux, sites Web, bases de données ouvertes, documents libres… La dernière mise à jour des données a été effectuée en octobre 2023, mais on peut lui demander d'aller sur Internet pour identifier des informations plus récentes. Quand on veut une réponse avec des informations récentes, il faut activer la recherche Web.

Entraînement : résumé :

Entraînement

1.  **Collecte des données issues de sources diverses (livres, sites Web, articles…).**
2.  **Prétraitement : nettoyage et structure des données textuelles.**
3.  **Tokenisation : division du texte en petites unités : tokens**
4. [6.90] **Encodage des tokens en vecteurs numérique (embedding)**
5.  **Choix de l'architecture : comme le Transformer (comment les données doivent être préparées)**
6.  **Apprentissage : le modèle est entraîné à analyser les relations entre les mots et à faire des prédictions**
7.  **Ajustements (fine-tuning) : par exemple avec un apprentissage supervisé**
8.  **Modèle prêt à l'emploi, déploiement.**
9.  **Renforcement par feedback humain (RLHF) : ajuster selon les préférences humaines.**

L’architecture des LLM :

Leur architecture spécifique est appelée Transformer. Elle excelle dans la compréhension des relations entre les mots dans un texte. Chaque mot est analysé dans son contexte global, c’est de l’auto-attention. Cela permet au modèle de se concentrer sur les parties les plus pertinentes et de générer des réponses plus cohérentes.

Auto-attention :

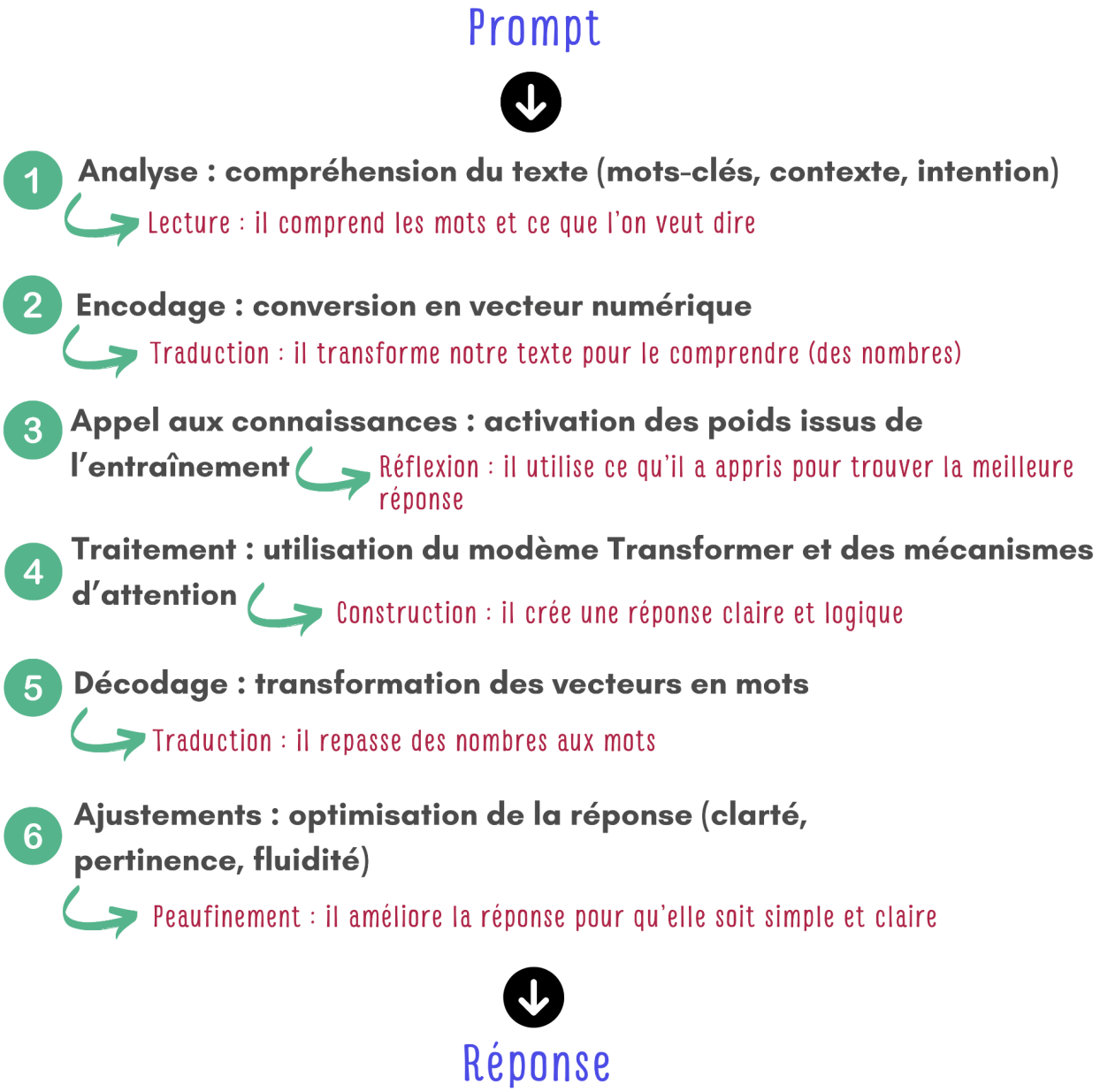
L'auto-attention est un mécanisme utilisé dans les modèles comme les transformers pour analyser chaque élément d'une séquence de données (comme les mots d'une phrase) en tenant compte de son contexte global.

Autrement dit, l'auto-attention permet au modèle de regarder toutes les autres parties de la séquence pour comprendre l'importance de chaque mot par rapport aux autres, ce qui l'aide à saisir les relations entre les mots même s'ils sont éloignés dans la phrase. Cela permet une meilleure compréhension et génération du texte.

Les réponses des LLM :

Quand le modèle reçoit un prompt, il prédit les mots qui suivent. Pour cela, il s’appuie sur sa vaste base de connaissances acquises durant l’entraînement. Il utilise les poids attribués à chaque mot pour formuler des phrases qui respectent la logique et le contexte donné. Quand il génère une réponse, il attribue un poids plus ou moins élevé aux différents mots en fonction de leur pertinence par rapport au contexte du prompt.

Par exemple : dans une phrase comme "Je vais au parc", le mot "parc" aura un poids important pour comprendre de quoi on parle, par rapport à des mots comme "je" ou "au", qui ont moins de poids car ils sont plus courants et servent à structurer la phrase. Les modèles d'IA, comme ceux basés sur l'auto-attention, utilisent ces poids pour déterminer quelles parties de la phrase sont les plus significatives et doivent être prises en compte pour générer une réponse cohérente.



Les hallucinations :

Dans le contexte de ChatGPT, les "hallucinations" désignent des situations où le modèle génère des informations fausses, inventées ou sans fondement réel, tout en les présentant avec une grande confiance et une apparente cohérence. Ces hallucinations ne sont pas le fruit d'une conscience ou d'une intention de tromper, mais plutôt le résultat de la manière dont le modèle a été entraîné. ChatGPT apprend à partir de vastes ensembles de données textuelles, et son objectif est de produire des réponses qui semblent plausibles et pertinentes en

fonction des motifs qu'il a identifiés. Parfois, cela conduit à la création de "faits" ou de "références" qui n'existent pas dans la réalité, mais qui sont construits de manière convaincante à partir des informations disponibles dans son entraînement.

Les biais :

Les modèles de langage comme ChatGPT sont entraînés sur d'immenses quantités de données textuelles provenant d'Internet, ce qui peut entraîner l'intégration de biais présents dans ces données. Ces biais peuvent se manifester de diverses manières, par exemple en favorisant certains groupes démographiques, en perpétuant des stéréotypes ou en reflétant des opinions biaisées. Il est crucial de reconnaître ces biais et de les prendre en compte lors de l'utilisation de ChatGPT, en évaluant de manière critique les informations fournies et en évitant de les considérer comme des vérités absolues.

Quel modèle génératif de langage utiliser ?

Il existe plusieurs modèles génératifs de langage. Un modèle de langage n’a pas d’interface, il a besoin d’une interface pour pouvoir être utilisé.

Exemple : GPT-4 est un modèle de langage, pour l’utiliser, vous pouvez utiliser l’application ChatGPT.

→ Une même application peut utiliser plusieurs modèles différents (comme ChatGPT avec GPT-3.5 et GPT-4o).

Et un même modèle (comme GPT-4) peut être utilisé dans plusieurs applications (ChatGPT, Copilot…).

Ces applications sont appelées “interfaces conversationnelles”. Ce sont des programmes qui permettent aux utilisateurs d’interagir avec un modèle de langage via du texte ou de la voix.

Voici une sélection :

ChatGPT

Concepteur : OpenAI
Entreprise : OpenAI (entreprise américaine)

Infos :

Modèles : GPT-3 (2020) → GPT-4 (2023) → GPT-4.5 / GPT-4o (2024)
Utilisé dans ChatGPT, Copilot (ex-Bing Chat), diverses apps tierces
Accès gratuit (GPT-3.5) et payant (GPT-4/4o via ChatGPT Plus)
Performant pour le langage, le code, l'image, le raisonnement

Lien : https://chatgpt.com/?model=auto

Claude

Concepteur : Anthropic
Entreprise : Anthropic (entreprise américaine)

Lien : https://claude.ai/login?returnTo=%2F%3F

Infos :
Modèle : Claude 1 (2023) → Claude 2 → Claude 3 (2024)
Modèles : Claude 3 Haiku (rapide), Sonnet (équilibré), Opus (le plus avancé)
Disponible via claude.ai (gratuit et payant)
Forte capacité de lecture de gros documents, ton « bienveillant »

Copilot

Concepteur : OpenAI + Microsoft
Entreprise : Microsoft (entreprise américaine)

Infos de base :
Intégration de GPT dans Word, Excel, PowerPoint, Edge et Bing
Version gratuite et premium (Microsoft 365 Copilot)
Axé productivité bureautique
Modèle : Utilise GPT-4 et GPT-4o

Lien : https://copilot.microsoft.com/

DeepSeek

Concepteur : DeepSeek AI
Entreprise : DeepSeek (startup chinoise indépendante)

Infos de base :
Modèle : DeepSeek-Coder (2023) → DeepSeek-V2 et DeepSeek-MoE (2024)

Modèles open source axés sur le langage général et le code (DeepSeek-Coder)
Utilisables via Hugging Face, GitHub, et API ; licences permissives (Apache 2.0)
Très performants en génération de code, traduction et raisonnement logique
Multilingue (anglais, chinois), avec versions dense et Mixture of Experts (MoE)
Lien : https://www.deepseek.com/

Gemini (ex-Bard)

Concepteur : Google DeepMind
Entreprise : Google (Alphabet) (entreprise américaine)

Infos :
Modèle : Gemini 1 (fin 2023) ⟶ Gemini 1.5 (2024)
Intégré dans les produits Google (Gmail, Docs, Search, etc.)
Remplaçant de Bard
Gratuit via gemini.google.com
Très bon sur le multimodal (texte, image, audio, vidéo)

Lien : https://gemini.google.com/?hl=fr

Mistral

Concepteur : Mistral.ai
Entreprise : Mistral ai (entreprise française)

Infos de base :
Modèles de langage open source (LLMs) développés en Europe
Spécialisé dans la performance, la légèreté et l'open access
Modèles : Mistral 7B, Mistral 8x7B, etc.
Pas d'interface conversationnelle officielle (utilisé via API ou intégrations tierces)
Atout : modèles performants, open source et adaptés à l'intégration dans des outils personnalisés

Lien : https://mistral.ai

Perplexity

Concepteur : Perplexity.ai
Entreprise : Perplexity (entreprise américaine)

Infos de base :
Moteur de recherche conversationnel basé sur LLMs (GPT, Claude, etc.)
Modèle : Utilise des modèles tiers selon la version (Claude, GPT-4o, Mistral)
Version gratuite et Pro
Atout : réponses sourcées et actualisées

Lien : https://www.perplexity.ai/?login-source=oneTapHom

Table des matières