Chapitre 12

IA génératives

Source : [1]

Bien que l'intelligence artificielle fasse parler d'elle depuis un certain temps maintenant, c'est bien l'IA générative qui a récemment éveillé un intérêt considérable dans le monde entier. Vous avez probablement entendu parler de *ChatGPT*, *DALL-E* ou peut-être *Midjourney*. Mais qu'est-ce que l'IA générative, et quel modèle se distingue comme le plus célèbre ?

Une contribution de Bernard Marr pour Forbes US – traduit par Lisa Deleforterie

La révolution de l'IA générative

L'IA générative fait référence à des systèmes d'intelligence artificielle capables de créer de nouveaux contenus, qu'il s'agisse de textes, d'images, de musiques ou même de codes. Ces modèles d'IA apprennent à partir de vastes quantités de données existantes pour générer quelque chose de nouveau et d'original. C'est comme apprendre à un ordinateur à être créatif ; et celui-ci apprend à une vitesse fulgurante.

Le champion incontesté : *ChatGPT* d'*OpenAI*

Quand il s'agit de notoriété dans le domaine de l'IA générative, un nom se démarque clairement : *ChatGPT* d'*OpenAI*. Mais pourquoi est-il si célèbre ?

1. **Innovation pionnière :** *ChatGPT* a été parmi les premiers à démontrer le véritable potentiel de l'IA générative d'une manière accessible au public. Il a transformé ce qui ressemblait à de la science-fiction en réalité.
2. **Polyvalence :** Qu'il s'agisse de rédiger des essais, de créer des images ou de déboguer des codes, la gamme de capacités de *ChatGPT* est impressionnante. C'est comme avoir un couteau suisse d'outils d'IA à portée de main.
3. **Adoption grand public :** En rendant *ChatGPT* librement accessible, *OpenAI* a démocratisé l'accès à l'IA avancée. Désormais, tout le monde, des étudiants aux PDG, peut expérimenter avec des contenus générés par l'intelligence artificielle.
4. **Évolution constante :** À chaque nouvelle version (GPT-3, GPT-4), le modèle gagne en puissance et en capacités, repoussant sans cesse les frontières du possible.

Les concurrents

Bien que *ChatGPT* soit le plus célèbre, il n'est pas le seul acteur en jeu. Le paysage de l'IA générative compte de nombreux concurrents impressionnants :

- ***Gemini* de *Google* :** Un rival solide de *ChatGPT*, offrant des capacités multimodales et des performances impressionnantes.
- ***Claude* d'*Anthropic* :** Connu pour son accent sur le développement éthique de l'IA et ses capacités conversationnelles robustes.
- ***LLaMA* de *Meta* :** La société mère de *Facebook* a également lancé son propre modèle de langage à grande échelle en open source.
- ***Stable Diffusion* de *Stability AI* :** Un choix populaire pour beaucoup dans le domaine de la génération d'images.
- ***Copilot* de *Microsoft* :** Intégrant une assistance IA dans tout l'écosystème *Microsoft*.

Chacun de ces modèles apporte quelque chose d'unique, mais l'avance et l'adoption généralisée de *ChatGPT* ont consolidé sa place au sommet de la renommée.

L'impact et les implications

L'essor de *ChatGPT* et de ses pairs d'IA générative a déclenché une véritable ruée vers l'or technologique. Les entreprises s'empressent d'intégrer ces technologies dans leurs processus, produits et services. Nous assistons à l'arrivée sur le marché d'assistants d'écriture alimentés par l'IA, de générateurs de code et même de thérapeutes virtuels.

Cependant, tout n'est pas sans défis. Les avancées rapides de l'IA générative ont également soulevé des questions importantes sur le droit d'auteur, la suppression d'emplois et la propagation de la désinformation. À mesure que ces modèles deviennent plus sophistiqués, distinguer le contenu généré par l'IA de celui créé par l'homme devient de plus en plus difficile.

Quel avenir pour l'IA générative ?

Le domaine de l'IA générative évolue à une vitesse vertigineuse. Nous verrons probablement émerger des modèles encore plus puissants et spécialisés dans les années à venir. L'intégration de l'IA générative dans notre vie quotidienne, de l'éducation personnalisée aux diagnostics de santé avancés, ne fait que commencer.

Quant à *ChatGPT*, sa renommée ne montre aucun signe de déclin. Avec *OpenAI* repoussant constamment les limites, nous pouvons anticiper que les futures versions seront encore plus performantes et, osons le dire, impressionnantes.

L'avenir alimenté par l'IA

Bien que *ChatGPT* soit actuellement en tête dans le monde de l'IA générative, la concurrence est féroce et le paysage est en constante évolution. Ce qui est clair, c'est que l'IA générative n'est pas une simple tendance passagère ; c'est une révolution technologique destinée à perdurer.

Alors que nous naviguons dans ce nouveau monde audacieux de contenu généré par l'IA, une chose est certaine : l'IA générative la plus célèbre d'aujourd'hui n'est peut-être que la partie émergée de l'iceberg. L'excitation réside dans l'imagination de ce que ces technologies pourront accomplir demain. Alors attachez vos ceintures, le voyage de l'IA ne fait que commencer !

Sources : [2, 3]

12.1. C'est quoi un *LLM* ? Comment fonctionnent les moteurs de *ChatGPT* et consorts ?

Qu'est-ce qui se cache derrière *ChatGPT* ? Comment fonctionnent les intelligences artificielles génératives textuelles ? Avec l'émergence des outils d'IA, nombreuses sont les questions autour de leur fonctionnement mystérieux.

En fait, derrière *ChatGPT*, il y a ce qu'on appelle un « *LLM* »… Un quoi ?

Qu'est-ce que ça veut dire « LLM » en IA ?

LLM est l'acronyme de l'expression anglaise « *Large Language Model* ». On pourrait la traduire en français par « *grand modèle de langage* ». Il s'agit de modèles de langage qui possèdent généralement au moins un milliard de paramètres. En français, on peut aussi les nommer « *modèles massifs de langage* » et les désigner avec l'acronyme « *MML* ».

Comment fonctionne un *Large Language Model*, le moteur des intelligences artificielles ?

Un *LLM*, c'est en réalité un réseau de neurones artificiels profonds[1], soit un logiciel dont la conception est inspirée du fonctionnement des neurones biologiques. Chaque neurone informatique (ou formel) possède des entrées (qui correspondent aux dendrites) ainsi qu'une sortie (correspondant à l'axone). À l'aide de règles précises qu'on lui indique, le neurone formel peut transformer une entrée en une sortie. Ces neurones artificiels sont associés en réseaux selon différents types de connexions (certaines auront plus de poids, ou exécuteront une tâche plus régulièrement).

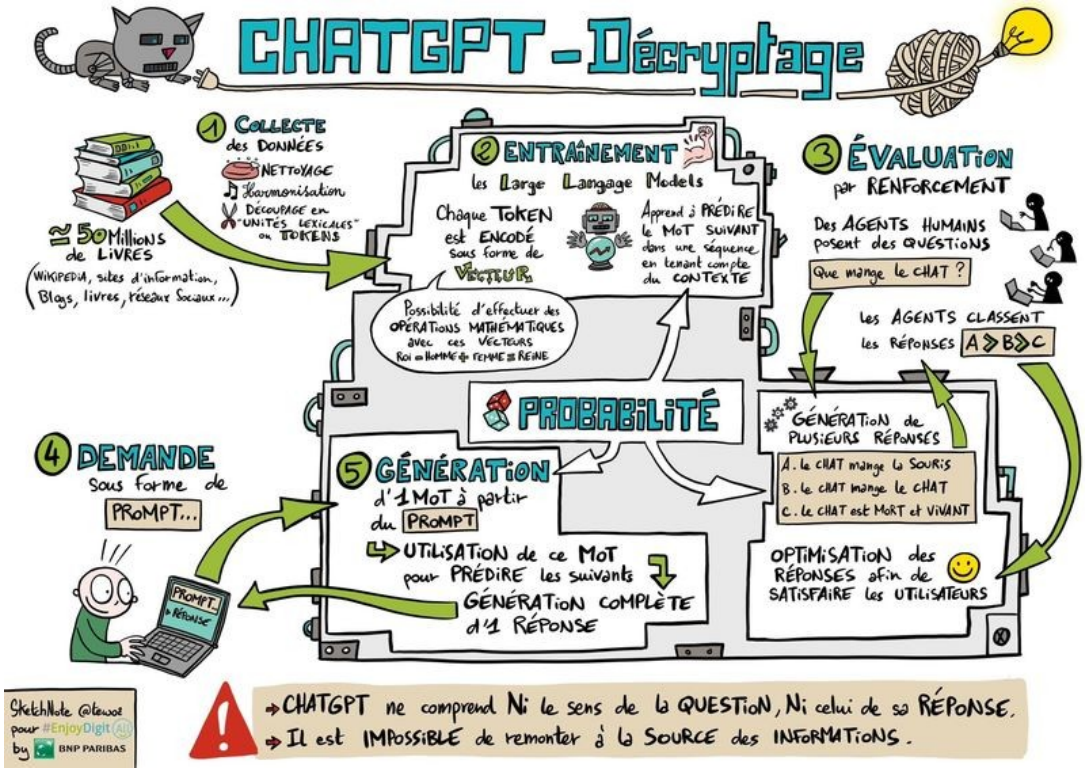
1 Pour en savoir plus, voir le paragraphe 4 du chapitre 11

La force de ce système de réseau de neurones, c'est que comme chez l'animal, il peut « *apprendre* » de lui-même : c'est le *machine learning*. Mais on peut aller plus loin avec l'apprentissage automatique (appelé *deep learning* en anglais), qui possède un avantage de taille : il ne nécessite pas qu'un être humain rentre « *à la main* » tout ce que la machine doit apprendre. De quoi décupler la puissance finale du système.

Pour faire apprendre à un *LLM*, il faut lui donner du texte, beaucoup de texte. Pour cela, on peut simplement prendre *Wikipédia* : selon la *Wikimedia Foundation*, l'encyclopédie en ligne compte plus de 58 millions d'articles en près de 300 langues. Il existe également des ensembles de données textuelles spécialisés dans l'entraînement de *LLM*, qui sont parfois *open source*.

La qualité de l'apprentissage dépend aussi de ce qu'on appelle l'étiquetage des données. Dans le domaine de l'intelligence artificielle, l'étiquetage est le fait de donner la réponse à une tâche demandée à partir de données déterminées. Pour du texte, l'étiquetage peut être par exemple de qualifier un texte de « *factuel* » dans son style, de « *familier* » dans son vocabulaire, ou bien « *d'injurieux* » dans ce qu'il dit.

Lorsqu'on partage du texte en entrée à un chatbot, il est transformé en nombres par le *LLM*, puis analysé, et une sortie est formée en nombres également, avant d'être convertie en texte en sortie. Ces nombres, on les appelle en fait des vecteurs. Ce sont ces nombres qui permettent d'instaurer des scores de proximité entre eux. Plus le nombre possède de chiffres, plus le modèle est complexe, et donc performant. C'est une sorte de mathématisation du texte qui s'effectue et c'est ce qui permet à un algorithme d'imiter le langage humain.



Ce qu'a changé l'architecture *Transformer* au *deep learning*

C'est en 2017 qu'un changement technologique va bouleverser le monde de l'intelligence artificielle : la création de l'architecture *Transformer*. Elle résulte d'une longue combinaison de procédés techniques, avec des travaux datant de nombreuses années.

Un « transformeur », c'est un modèle d'apprentissage profond, principalement taillé pour ce qu'on appelle le traitement automatique des langues. Là où les réseaux neuronaux traditionnels comme les réseaux de neurones récurrents traitent une requête en entrée de manière séquentielle (du début d'une phrase à la fin), le transformeur peut paralléliser cette entrée, afin de considérablement réduire les temps d'entraînement. Réduire les temps d'entraînement, c'est avoir plus d'entraînements pour un coût de fonctionnement de serveurs égal et aller plus loin.

Un bon exemple de l'intérêt de cette architecture est raconté par le philosophe Daniel **Andler** dans son ouvrage *Intelligence artificielle, intelligence humaine : la double énigme*. Pour la phrase « *j'ai un frère, il est architecte* », « *frère* » et « *il* » désignent la même personne : la construction de la phrase est simple et les deux termes se suivent. Mais dans la phrase « *quand mon frère s'est fâché avec son associé, je lui ai avoué qu'il ne m'avait jamais plu* », « *mon frère* » et « *lui* » sont éloignés. C'est là que le *Transformer* utilise un mécanisme d'« *auto-attention* », qui prend en compte « *ces effets à distance du contexte* ». C'est ce mécanisme qui permet de prendre un contexte dans le traitement d'une entrée. Un mécanisme qui fonctionne sur deux principes : les « *masks* » et les « *tokens* ».

Pour le premier, il y a deux types de masques :

- Les « *filtres de causalité* » qui vont modifier le poids de certains vecteurs en fonction du contexte donné par la phrase ;
- Les « *filtres de padding* » qui font en sorte que toutes les phrases aient la même longueur mathématique (autant de nombres en elles), en ajoutant des mots inutiles et non pris en compte dans le traitement.

Ce sont les *tokens* qui permettent aux réseaux neuronaux de « *comprendre* » chaque mot en les traitant, pas seulement les uns à la suite des autres. Ils attribuent aussi des liens entre les mots.

Les premiers « *vrais* » modèles de langage : *GPT* et *BERT*

Deux *LLM*, qu'on peut considérer comme des pionniers, ont été publiés en 2018 à quelques semaines d'écart. Le premier, c'est *GPT*, pour *Generative Pre-Trained Transformer* d'*OpenAI.* Le second, c'est *BERT* de *DeepMind* (qui appartient à *Google*). Grâce à l'architecture *Transformer*, ils se sont révélés être des révolutions dans les *LLM*.

Ils sont très bons en compréhension du langage naturel ainsi qu'en génération de texte. Ils peuvent réaliser des tâches simplement en leur demandant de manière textuelle : « *résumer* », « *traduire* », « *rédiger* ». C'est aussi l'une des premières fois que des modèles de langage ne sont pas pré-entraînés pour une tâche particulière, mais pour tout un ensemble, dont on ne connaît même pas l'étendue.

Pourquoi parle-t-on de « *paramètres* » pour un modèle de langage ?

Lorsqu'on parle de *LLM*, on parle beaucoup de « *paramètres* » : plus il y en a, plus un modèle serait puissant, performant. C'est vrai, mais ce n'est pas une vérité générale. En fait, les réseaux neuronaux contiennent plusieurs nœuds, sur plusieurs couches. Comme l'explique *Amazon Web Services*, « *chaque nœud de chaque couche est connecté à tous les nœuds de la couche suivante* ». Chacun a un poids et un écart différent : ce sont ces poids et ces écarts qui sont en fait les paramètres d'un *LLM*. C'est pour cela qu'on peut « *facilement* » en avoir des dizaines de milliards. Ce qu'offrent les paramètres, c'est la capacité à davantage capturer de nuances et de complexités dans le langage. Cela permet de prendre en compte des données en entrée plus importantes et des sorties qui le sont aussi. Néanmoins, plus un *LLM* va loin dans la « *compréhension* », plus il lui faut de paramètres (de manière exponentielle) et de puissance (de serveurs). Durant la phase d'entraînement, ce sont les poids et les écarts qui sont ajustés de manière itérative.

À quoi servent les *large language models* ?

La grande force des *LLM*, c'est précisément qu'ils n'ont pas d'usage précis, puisqu'ils n'ont pas été entraînés sur une capacité en particulier. Leur fonctionnement neuronal fait qu'ils sont entraînés à la prédiction d'une suite probable en fonction d'une entrée donnée (une séquence de mots).

Si vous demandez à *ChatGPT* de vous raconter une histoire, un conte pour enfants par exemple, il va probablement démarrer par « *Il était une fois* », puisque c'est très classique. Ensuite, la probabilité de ce qui arrive après est « *dans un royaume* », ou « *une princesse* », quelque chose comme cela. En réalité, les *LLM* ne « *comprennent* » pas les textes sur lesquels ils ont été entraînés ni ce qu'on leur écrit. Les *LLM* sont simplement des systèmes statistiques, appliqués à la linguistique. Ils ne déterminent pas que des mots, mais également toute la syntaxe, la conjugaison et la ponctuation de ce qui fait les langues.

Ce qui fait qu'un *LLM* va être performant dépend de plusieurs facteurs. Tout d'abord, il y a le nombre des paramètres. Plus ils sont nombreux, plus le modèle de langage pourra prendre de facteurs en compte dans sa réponse, ce qui fera qu'elle sera plus précise. D'ailleurs, on découvre certaines capacités en agrandissant le modèle, en augmentant le nombre de paramètres. Comme l'écrit Daniel

Andler, « *une propriété émerge à partir d'une certaine taille, sans que l'on sache aujourd'hui pourquoi.* » La capacité de traduction, la simulation des émotions ou de l'humour en sont quelques exemples.

Cela dépend par ailleurs de la puissance de calcul consacrée au fonctionnement du *LLM*. Enfin, il y a la qualité des données qui lui ont été fournies en entrée par l'utilisateur. En clair, plus votre demande à *ChatGPT* est précise, plus le *LLM* derrière aura de contexte et d'informations pour vous apporter une réponse précise. La qualité des données comprend également la largeur de l'éventail de données qu'il a eu pour s'entraîner, ainsi que la qualité de leur étiquetage. Plus l'étiquetage a été poussé, plus le modèle peut « interpréter » les données d'entraînement et celles qu'on lui fournit lors de la requête.

Les *LLM* peuvent être entraînés pour un certain nombre de tâches. L'une des utilisations les plus connues est leur application en IA générative : ils sont capables de produire un texte de réponse à une invite ou une question. Par exemple, le *LLM ChatGPT*, disponible pour le grand public, peut générer des essais, des poèmes et d'autres formes textuelles en réponse aux entrées de l'utilisateur.

Tout ensemble de données complexes et de grande taille peut être utilisé pour entraîner des *LLM*, y compris les langages de programmation. Certains *LLM* peuvent aider les programmeurs à écrire du code. Ils peuvent écrire des fonctions à la demande ou terminer l'écriture d'un programme à partir d'un code qui leur a été donné comme point de départ. Les *LLM* peuvent également être utilisés pour les besoins suivants :

- Analyse des sentiments
- Résumé d'un article ou un livre
- Service à la clientèle
- Chatbots
- Recherche en ligne (*Perplexity*)
- Traduction automatique (*DeepL*)

Parmi les exemples de *LLM* réels, citons *ChatGPT* (*OpenAI*), *Gemini* (anciennement *Bard* de *Google*), *Llama* (*Meta*), *Claude* (*Anthropic*), *Perplexity* et *Bing Chat* (*Microsoft*). Le *Copilot* de *GitHub* est un autre exemple, mais pour il s'applique au codage et non au langage humain naturel.

Quels sont les avantages et les limites des *LLM* ?

L'une des principales caractéristiques des *LLM* est leur capacité à répondre à des requêtes imprévisibles. Un programme informatique traditionnel reçoit des commandes dans sa syntaxe acceptée, ou à partir d'un certain ensemble d'entrées de l'utilisateur. Un jeu vidéo dispose d'un ensemble fini de boutons, une application dispose d'un ensemble fini d'éléments sur lequel l'utilisateur peut cliquer ou qu'il peut saisir, et un langage de programmation est composé d'énoncés précis de type « si » et « alors ».

De son côté, un *LLM* peut répondre à un langage humain naturel et utiliser l'analyse de données pour répondre à une question non structurée ou à une invite d'une manière logique. Alors qu'un programme informatique classique ne reconnaîtrait pas une question telle que « Quels sont les quatre plus grands groupes de funk de l'histoire ? », un *LLM* peut répondre par une liste de quatre groupes de ce type et exposer un argument convaincant de ce qui en fait les meilleurs.

Cependant, concerne les informations qu'ils fournissent, les *LLM* ne seront fiables que si les données qu'ils ingèrent le sont. S'ils reçoivent des informations erronées, les réponses qu'ils donneront aux demandes de l'utilisateur le seront également. Il arrive que les *LLM* « hallucinent » un peu : ils créent de fausses informations lorsqu'ils ne sont pas en mesure de fournir une réponse précise. Par exemple, en 2022, le magazine *Fast Company* a interrogé *ChatGPT* sur le dernier trimestre financier de l'entreprise *Tesla* ; *ChatGPT* a certes produit un article cohérent en réponse, cependant une grande partie des informations contenues dans l'article ont été inventées.

En matière de sécurité, les applications destinées aux utilisateurs et basées sur les *LLM* sont aussi sujettes aux bugs que n'importe quelle autre application. Les *LLM* peuvent également être manipulés par des intrants malveillants afin de fournir certains types de réponses plutôt que d'autres, y compris des réponses dangereuses ou contraires à l'éthique. Enfin, les *LLM* posent un certain nombre de problèmes de sécurité liés notamment au fait que les utilisateurs peuvent y importer des données sécurisées et confidentielles afin d'accroître leur propre productivité. Toutefois, pour entraîner leurs modèles, les *LLM* utilisent les données qu'ils reçoivent, et ils ne sont pas conçus pour être des

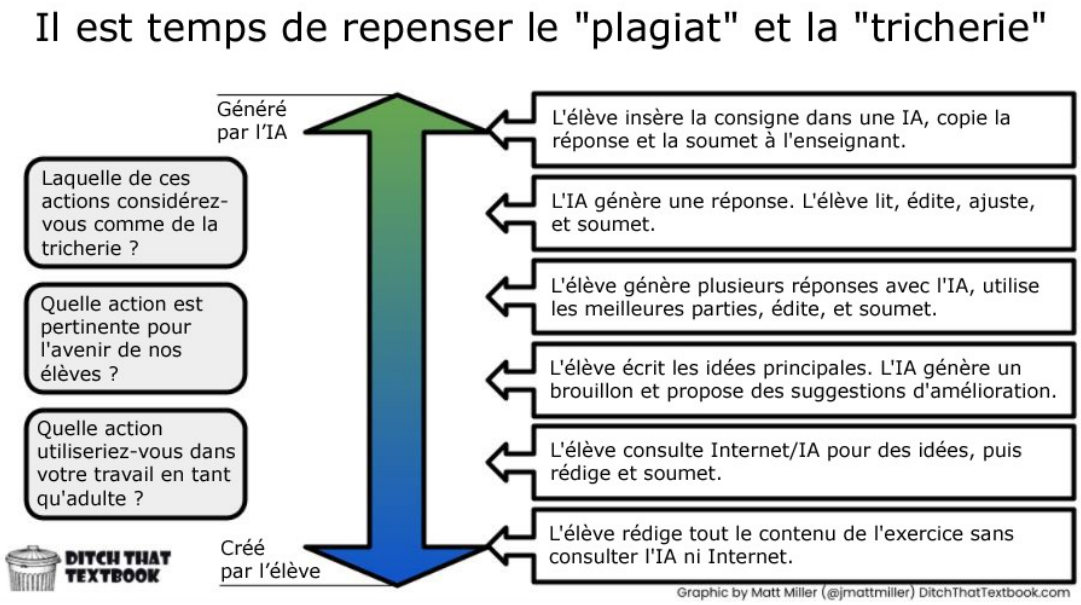
coffres-forts sécurisés ; il peut arriver qu'ils exposent des données confidentielles en réponse à des requêtes provenant d'autres utilisateurs.

12.2. IA et enseignement

Matrice SWOT de l'usage de l'IA générative dans l'enseignement

Source : [10]

Forces	**Faiblesses**
• Accessibilité • Gain de productivité • Créativité • Conversation pour générer une réponse plus étayée et personnalisée • Sources diverses et dans une grande variété de langues	• Biais et censure • Manque de transparence • Réponses dont la fiabilité et l'exactitude sont difficilement mesurables par un utilisateur non expert • Qualité des réponses dépendant du niveau d'expertise de l'utilisateur • Difficulté de dissocier le plagiat du travail original
Opportunités	**Menaces**
• Actualisation des pratiques de l'enseignement • Développement de nouvelles compétences • Développement de l'esprit critique • Personnalisation de l'apprentissage • Autonomisation de l'apprenant	• Satisfaction de réponses faciles et superficielles • Perte du goût de l'effort • Perte de l'esprit critique • Dépendance technologique • Réduction de l'interaction sociale • Utilisation des données personnelles sujette à caution



Chaque enseignant aura sa propre sensibilité. Pour ma part, je place la limite de l'acceptable au niveau 3 (depuis le bas) : « L'élève écrit les idées principales. L'IA génère un brouillon et propose des suggestions d'amélioration ». Au-dessus, c'est de la tricherie.

12.3. Prompt engineering

Un nouveau métier qui est apparu en 2021…

L'ingénierie de prompt (issu de l'anglais *prompt engineering*) est le processus de structuration d'un prompt, de sorte qu'il soit efficacement interprété et compris par un modèle d'intelligence artificielle générative. Un prompt (aussi appelé requête, ou invite de commande) est une description textuelle de la tâche qu'une IA doit effectuer.

12.3.1. L'importance de maîtriser la rédaction de prompts

Source : [4]

Dans le monde de l'IA, le « Prompt Engineering » est devenu une compétence incontournable. Pourquoi ? Parce que la qualité de votre interaction avec *ChatGPT* (ou un de ses concurrents) dépend en grande partie de la qualité de vos prompts.

Un prompt bien rédigé peut être la différence entre une réponse vague et une réponse précise. Il peut influencer la rapidité avec laquelle vous obtenez des informations, et même la quantité d'informations que vous recevez.

Les avantages d'un prompt bien rédigé

- **Des résultats précis** : Un prompt clair et concis donne des réponses claires et concises. Cela signifie moins de temps passé à interpréter les réponses ou à poser des questions de suivi.
- **Efficacité** : Avec un prompt bien rédigé, vous obtenez ce que vous voulez du premier coup. Cela signifie moins de requêtes, moins de temps passé à attendre et une meilleure expérience globale.
- **Réduction des coûts** : Moins de requêtes signifie moins de coûts. Si vous payez à la requête, cela peut rapidement s'additionner.

12.3.2. Les 3 ingrédients de base d'un prompt parfait

Cela est aussi valable avec les autres LLM.

Il y a trois éléments clés à considérer lors de la rédaction d'un prompt pour *ChatGPT* :

1. **Le contexte** : L'IA a besoin de comprendre le contexte pour fournir une réponse pertinente. Plus vous donnez de détails, mieux c'est.
2. **La tâche** : Soyez spécifique sur ce que vous voulez. Si vous voulez une liste, demandez une liste. Si vous voulez une explication détaillée, demandez-la.
3. **Le rôle** : Indiquez à l'IA comment vous voulez qu'elle réponde. Voulez-vous qu'elle soit un enseignant ? Un conseiller ? Un expert ? Le rôle peut influencer le ton et le niveau de détail de la réponse.

Des exemples basiques pour illustrer

- **Recherche d'une solution à un problème spécifique**
  - Mauvais prompt : « `Mon ordinateur ne démarre pas.` »
  - Bon prompt : « `En tant que technicien informatique, peux-tu me guider à travers les étapes pour résoudre le problème d'un ordinateur qui ne démarre pas ?` »

- **Demande d'opinion ou de conseil**
  - Mauvais prompt : « `Dois-je acheter une voiture électrique ?` »
  - Bon prompt : « `En tant que conseiller automobile, peux-tu me donner les avantages et les inconvénients d'acheter une voiture électrique par rapport à une voiture à essence ?` »

- **Exploration d'un sujet complexe**
  - Mauvais prompt : « `Parle-moi de Mars.` »
  - Bon prompt : « `En tant qu'astrophysicien, peux-tu m'expliquer les principaux défis de la colonisation de Mars ?` »

Chaque bon prompt ci-dessus intègre clairement le contexte, définit la tâche attendue et assigne un rôle spécifique à l'IA. Ces éléments combinés permettent d'obtenir des réponses plus précises et pertinentes de *ChatGPT*.

Sources : [5, 13]

12.3.3. Autres conseils pour encore améliorer les réponses

Voici à quoi ressemble un bon prompt :

Agis en tant que {Rôle}.
J'ai besoin de {Objectifs} auprès de {Cible}.
Tu vas donc {Verbe d'action} en suivant les {étapes}.
Le résultat doit être un {Format}. Adopte un {Ton}.

Rôle	Verbe d'action	Format	Ton
Prof de Lycée	Rédiger	Liste	Didactique
Astronome	Résumer	Tableau	Décontracté
Ingénieur	Analyser	Code HTML	Lyrique
Albert Einstein	Lister	Code Python	Professionnel
Avocat expérimenté	Simplifier	PDF	Humoristique
...	...	...	...

Exemples

Testez ces prompts !

```
Je veux que tu agisses en tant que formateur bureautique. Rédige un
tutoriel accessible pour les débutants sur Excel qui leur explique
simplement à quoi sert un tableau croisé dynamique dans Excel et
comment le mettre en œuvre. Écris avec un ton pédagogue et expert.

Rédige un texte de 1000 mots maximum sur l'histoire d'Internet, sous
forme chronologique, avec un sommaire et 2 niveaux de titres, en
précisant les dates et le nom des contributeurs et en concluant sur
l'avenir d'Internet.

Agis comme un spécialiste de la publicité sur les réseaux sociaux.
Tu réponds aux besoins d'entreprises souhaitant faire connaître
leurs produits. Ta tâche est de donner les meilleurs conseils pour
augmenter les ventes de ces entreprises grâce à la publicité sur les
réseaux sociaux. Tu as bien compris ?

J'essaie de mieux comprendre le concept d'équation différentielle
pour lequel je suis débutant et toi expert. Aide-moi en cela en
créant une analogie pratique et facile à comprendre.

Joue le rôle d'un examinateur pour le sujet que je te donnerai.
Interroge-moi alors sur ce sujet avec 10 questions fermées  où une
seule réponse est vraie. Va à la ligne avant de me faire 4
propositions de réponse. Évalue ma réponse après chaque question
avec une note de 0 si c'est faux et de 1 si c'est vrai. Donne-moi la
bonne réponse avec une brève explication seulement si ma note est de
0 parce que je me suis trompé. Écris mon score après les 10
questions.
```

Fournir un maximum de contexte à *ChatGPT*

Il peut sembler naturel de poser une simple question à *ChatGPT* et d'attendre la réponse. Mais une chose est sûre : plus le prompt est basique, plus la réponse du chatbot le sera également. Il convient donc d'apporter le maximum de contexte et de détails pour alimenter l'IA en informations. L'exemple du marathon est très parlant.

Nous avons demandé à *ChatGPT* (et vous ferez l'expérience vous-mêmes dans l'exercice 12.1)

« comment préparer un marathon ? ». Question à laquelle le chatbot a répondu en donnant une liste de huit conseils assez génériques sur la pratique sportive.

Dans un deuxième temps, nous avons précisé à *ChatGPT* être un débutant dans le domaine de la course à pied et vouloir courir un marathon dans six mois. La réponse de l'IA est alors beaucoup plus adaptée à notre prompt contextualisé. Elle nous donne un programme d'entraînement, des conseils sur la récupération, l'équipement, etc.

Enfin, nous avons demandé à *ChatGPT* comment préparer un marathon, lorsqu'on a l'expérience de la course, un bon rythme et trois mois devant soi. À nouveau, le chatbot donne une réponse beaucoup plus personnalisée, en prenant en compte les informations contextuelles fournies en amont.

Exercice 12.1

Dans les exercices de ce chapitre, vous pourrez tester des prompts sur *ChatGPT* et ses concurrents. À cet effet, vous devrez souvent créer un compte...

Avec *Copilot* de *Microsoft*, vous pourrez utiliser votre compte scolaire. Vous pourrez générer du texte et des images.

- *ChatGPT* : https://chatgpt.com/
- *Claude* : https://claude.ai/
- *Copilot* : https://copilot.microsoft.com/
- *Deepseek* : https://www.deepseek.com/
- *Gemini* : https://gemini.google.com/
- *Grok* : https://x.ai/
- *Llama* : https://llama.meta.com/
- *Mistral* : https://mistral.ai/fr/
- *Qwen* : https://chat.qwen.ai/

Testez ces trois prompts :

1. **« Comment préparer un marathon ? »**
2. **« Je suis un débutant en course à pied et je n'ai jamais couru de marathon. Je veux en courir un dans six mois. Comment préparer un marathon ? »**
3. **« Je fais régulièrement des courses de 10 km, une distance que je parcours en 43 minutes. J'ai plus de 10 ans d'expérience en course à pied. J'ai maintenant 45 ans et je voudrais courir un marathon dans 3 mois. Comment préparer un marathon ? »**

Donner un rôle, une identité, une profession à *ChatGPT*

Une des bonnes techniques pour tirer le meilleur de *ChatGPT* est de lui assigner un rôle. Poser une question via un prompt vous fournira toujours une réponse, mais elle ne sera peut-être pas adaptée à votre niveau de compréhension, de connaissance du sujet, de vulgarisation souhaitée.

Par exemple, *ChatGPT* peut rapidement vous expliquer le fonctionnement du Soleil si vous lui demandez. Mais peut-être que parler de fusion nucléaire n'est pas la chose la plus simple à comprendre. Pour simplifier la réponse de *ChatGPT*, nous lui avons demandé (et vous le ferez aussi dans l'exercice 12.2), dans notre nouveau prompt, d'incarner un enseignant dans une classe de CP-CE1. Et, de nouveau, de nous expliquer comment fonctionne le Soleil. L'IA joue alors de comparaisons imagées et adaptées à l'âge de notre audience pour expliquer son propos.

De la même manière, mais dans un autre contexte, *ChatGPT* peut se muer en expert s'il comprend qu'il est un émetteur qualifié et qu'il s'adresse à un auditoire composé de ses pairs. L'explication du fonctionnement du Soleil par un scientifique devant un parterre de confrères se veut donc plus complexe, sans détailler des termes dont on peut estimer qu'ils sont connus par l'audience.

Enfin, donner une identité à *ChatGPT* ne concerne pas forcément que la profession. Il est tout à fait possible de le faire incarner des personnages, fictifs ou non, pour générer un type de réponse particulier. Le Pr. Emmett Brown du film *Retour vers le futur* peut expliquer le fonctionnement du Soleil.

L'exercice 12.2 vous montrera la façon dont différentes identités permettent d'obtenir d'autres perspectives et d'autres discours sur un même sujet.

Exercice 12.2

Testez ces quatre prompts :

1. **« Comment fonctionne le Soleil ? »**
2. **« Je veux que tu prennes le rôle d'un enseignant d'école primaire. Tu donnes des cours à des enfants de 7 ans. Comment fonctionne le Soleil ? »**
3. **« Je veux que tu prennes le rôle d'un scientifique particulièrement qualifié sur les questions solaires, qui fait une conférence devant des confrères. Comment fonctionne le Soleil ? »**
4. **« Je veux que tu me répondes comme le ferait le professeur Emmett Brown dans le film *Retour vers le futur*. Comment fonctionne le Soleil ? »**

Le professeur Emmett **Brown**

Définir sa cible et son canal de communication

Quand on communique avec *ChatGPT*, il est primordial de savoir à qui sera destinée la réponse générée, sur quel support, et surtout de communiquer ces informations à l'IA.

Par exemple, si sur votre chaîne *YouTube* consacrée au jardinage, vous voulez réaliser une vidéo sur le matériel nécessaire pour entretenir son potager, ne vous contentez pas de simplement poser la question « de quel matériel ai-je besoin pour faire un potager ? » à *ChatGPT*. Certes, le chatbot vous fournira une liste dudit matériel, mais il n'adaptera sa réponse à aucune cible.

Si vous précisez en revanche que la réponse de *ChatGPT* doit être une ébauche de script pour une vidéo YouTube, destinée à un public féminin, la réponse va être bien plus adéquate.

De même, il est tout à fait possible d'adapter la réponse de *ChatGPT* à ses besoins. Notamment sur le format généré, qu'il s'agisse d'un script pour une vidéo *YouTube*, ou comme ci-dessous d'une chanson destinée à un public adolescent.

Exercice 12.3

Testez ces trois prompts :

1. **« De quel matériel ai-je besoin pour faire un potager ? »**
2. **« Je tiens une chaîne Youtube sur le jardinage, destinée à un public féminin et souvent parent. J'ai besoin du script pour une vidéo qui décrit le matériel dont on a besoin pour faire un potager. »**
3. **« Pour un live musical sur Twitch, j'ai besoin de chanter une chanson sur le matériel dont on a besoin pour faire un potager. Ce sont surtout des adolescents amateurs de rap qui m'écoutent. »**

Définir un format de sortie

Au-delà du type de réponse que vous voulez générer par rapport à votre audience – un script, une chanson -, il est aussi possible de demander à *ChatGPT* un format de sortie particulier pour, notamment, créer du contenu à destination du web. Le langage HTML offre par exemple un balisage léger et est connu de *ChatGPT*, qui peut donc générer un texte prêt à être intégré à votre page web.

Nous avons demandé à l'IA (et vous le ferez dans l'exercice 12.4) un court article sur les cinq plus importants séismes de l'histoire.

Notre prompt contient donc la demande d'un article sur les cinq plus grands séismes, avec des titres et sous-titres ainsi que des tableaux récapitulatifs, en langage HTML. *ChatGPT* va générer ce texte avec des balises, qu'il suffit ensuite de copier et coller sur notre page web.

Exercice 12.4

Testez ces trois prompts :

1. **« Rédige un article de 250 mots sur les 5 séismes les plus importants de l'histoire. »**

2. **« Rédige un article de 250 mots sur les 5 séismes les plus importants de l'histoire, sous forme d'un tableau. »**
3. **« Rédige un article de 250 mots sur les 5 séismes les plus importants de l'histoire en langage HTML pour ma page web, avec un titre et un sous-titre pour chaque séisme. Ajoute à la fin un tableau récapitulatif pour chaque séisme, sa puissance et les décès provoqués. »**

Effectuer des requêtes à la chaîne (chained prompting)

Si vous voulez obtenir plusieurs contenus rattachés à une même requête, il est pertinent d'utiliser les *chaînes de prompts* – ou *chained prompting* en anglais – qui vont vous permettre d'effectuer plusieurs demandes à la suite ou d'en préciser la teneur.

Nous voulons par exemple un article de 200 mots sur les séismes (voir exercice 12.5). Mais, dans la foulée, nous souhaitons qu'il comporte des titres et des intertitres.

Puis nous demandons à *ChatGPT*, sans préciser de nouveau que la base est le premier article de 200 mots, un court extrait pouvant faire office de meta description. Et nous voulons, en une phrase, un résumé des trois points forts de l'article.

Enfin, nous demandons à *ChatGPT* d'en détailler un en une centaine de mots. *ChatGPT* « se souvient » donc, au sein d'une même conversation, de vos précédents prompts et est capable de se baser sur le premier d'entre eux pour répondre aux suivants. Avec cette chaîne de requêtes, l'IA peut vous générer tout le contenu dont vous avez besoin au fur et à mesure.

Exercice 12.5

Testez **consécutivement** ces cinq prompts :

1. **« Je voudrais un article de 200 mots sur les séismes. »**
2. **« Je voudrais cet article mais avec un titre, au moins deux intertitres et une conclusion. »**
3. **« Rédige un extrait de 50 mots qui pourra faire office de meta description. Il faut donc qu'elle contienne les mots-clés en rapport aux séismes. »**
4. **« Rédige un court paragraphe de 20 mots avec les trois points forts de l'article. »**
5. **« Développe un des points forts en au moins 100 mots. »**

Donner un exemple de la réponse attendue à *ChatGPT*

Outre le « chained prompting », une autre technique nommée « *one-shot prompting* » est souvent utilisée pour uniformiser les réponses et leur donner l'aspect voulu dès l'énoncé de la requête.

Dans l'exercice 12.6, nous souhaitons que *ChatGPT* nous génère une liste de clubs de football comportant plusieurs détails, comme leur date de fondation, les couleurs du maillot, le palmarès, les rivaux, etc. Nous donnons directement le format à l'IA en lui transmettant un exemple de ce que nous voulons, puis en lui fournissant la liste des clubs sur lesquels rédiger.

Exercice 12.6

Testez ce prompt :

```
« Décris ces clubs de football selon l'exemple donné ci-dessous :
Nom : Real Madrid (Espagne)
Date de fondation : 6 mars 1902
Couleur(s) : blanc
Palmarès : 35 championnats d'Espagne, 19 coupes d'Espagne, 12
supercoupes d'Espagne, 14 ligues des champions, 2 coupes UEFA, 5
coupes du monde des clubs.
Rivaux : FC Barcelone, Atlético de Madrid.

FC Barcelone
Liverpool FC
Bayern Munich
Juventus de Turin »
```

Des conseils supplémentaires pour donner le bon prompt à *ChatGPT*

1. **Utiliser des verbes d'action et des mots forts :** avec *ChatGPT*, il faut être le plus clair et précis possible et cela passe par l'utilisation des bons termes. Les verbes sont importants. Plutôt que d'écrire en prompt « réécris ce texte », essayez par exemple « clarifie ce texte ». Préférez aussi la forme positive, en ne demandant pas « n'écris pas de manière familière » ou « n'écris pas à la forme passive », mais « écris de manière soutenue » ou « écris à la forme active ».

2. **Dire ce qui ne va pas à *ChatGPT* :** si l'IA vous fournit une réponse qui ne vous convient pas, c'est qu'elle ne vous a pas compris. Et il faut lui dire ce qui a engendré cette erreur d'interprétation, ce qu'elle a manqué, afin qu'elle puisse corriger facilement son erreur sans que vous ayez à tout recommencer.

3. **Limiter la taille des réponses :** *ChatGPT* n'est pas infaillible, surtout avec les tâches longues. Au-delà de 500 mots dans la réponse, le chatbot a parfois tendance à s'arrêter net. Et il perd parfois le fil quand on lui demande de continuer sa rédaction.

4. **Reformuler et reposer les questions :** si les réponses ne vous conviennent pas, il faut les reposer ou les reformuler, car bien souvent *ChatGPT* change de réponse à chaque question. En outre, en incarnant des personnages, en précisant la cible ou l'émetteur, l'IA pourra donner des éléments de réponses qu'elle a pu évincer auparavant.

5. **Demandez à *ChatGPT* s'il a bien compris** la question : « As-tu bien compris ? » Si ce n'est pas le cas, il vous demandera des précisions.

12.4. Générateurs d'images

Au paragraphe 12.2, nous avons utilisé des générateurs de textes. Il existe aussi des générateurs d'images dont le principe est identique : on décrit le mieux possible l'image que l'on veut dans un prompt et l'IA nous renvoie une ou plusieurs images.

Voici les principaux (août 2024) :

- ***DALL·E*** est une IA génératrice d'images lancée en janvier 2021 qui permet de créer des images à partir de consignes rédigées sous forme de texte. Cet outil est accessible gratuitement via le site *Bing* de *Microsoft* (*Copilot Designer*) ou sur *ChatGPT* en payant un abonnement premium.
- ***Midjourney*** est un modèle d'IA génératrice d'images dont la première version date de 2022. Depuis, de grands progrès ont été faits sur le réalisme des émotions et des traits de visage comme sur le traitement de l'éclairage et de la profondeur de champ. La V6 de *Midjourney*, sortie en décembre 2023, en a fait l'une des IA de génération d'image les plus créatives du marché mais également avec les rendus les plus photoréalistes.
- ***Stable Diffusion*** est un modèle d'IA générative conçu pour générer des images ultra réalistes. Développé depuis 2022 par l'université de Munich, *Runway* et *Stability AI*, cet outil complet vous permet de créer des images à partir de texte ou d'images préexistantes en suivant une consigne donnée.
- ***Leonardo*** est un générateur d'image par intelligence artificielle très utilisé pour la conception de jeux vidéo. Si cet outil permet comme d'autres modèles d'IA de produire ou de retoucher n'importe quel type d'images, ses fonctionnalités avancées sont avant tout utilisées par les gamers pour générer des rendus créatifs de paysages, de textures, de personnages ou d'objets.
- Il en existe beaucoup d'autres, gratuits ou payants (voir [6] pour plus de détails) : *MyEdit*, *Adobe Firefly*, *Craiyon*, *NightCafe Creator*, *Canva Text-to-image*, *Playground AI*, *OpenArt*, *Civitai*, *StarryAI*, *Bluewillow*, *Ideogram*, etc.

12.4.1. Copilot Designer

Le processus pour créer une image est essentiellement le même quel que soit l'outil utilisé. Pour nous entraîner, nous allons utiliser *Copilot Designer de Microsoft* (anciennement *Bing Image Creator*).

Quatre raisons à cela :

- il est entièrement gratuit (il y a cependant une limite quotidienne) ;
- pour pouvoir l'utiliser, il suffit d'avoir un compte *Microsoft* (vous pourrez utiliser votre compte scolaire) ;
- il est simple d'utilisation : tout se passe sur une page web et on peut faire ses requêtes en français (mais les résultats seront peut-être meilleurs en anglais) ;
- il génère des images avec DALL·E 3.

Comment utiliser Copilot Designer ?

Source : [7]

1. Pour commencer, rendez-vous à l'adresse https://www.bing.com/images/create.
2. Connectez-vous avec un compte *Microsoft* et vous accéderez à l'interface navigateur de l'IA.
3. *Copilot Designer* mise sur la simplicité pour générer des images. Contrairement à *Midjourney*, qui passe par un serveur *Discord*, et dispose de nombreuses commandes et options, *Copilot Designer* se limite à une boîte de texte où vous pouvez rédiger vos prompts. Une fois votre requête prête, il vous suffit d'appuyer sur le bouton *Créer* et l'IA vous générera 4 images.
4. Le modèle de base de prompt est le suivant : « *Adjectif* + *Nom* + *Verbe* + *Style* ». Mais il est tout à fait possible d'inverser le nom (ou le sujet) et l'adjectif.
5. Vos dernières générations (contenant chacune 4 images) seront conservées dans « Mes créations ». Vous pouvez aussi les télécharger sur votre ordinateur.
6. Notez que si vous êtes en manque d'inspiration, une option *Surprenez-moi* est disponible : celle-ci vous propose un prompt aléatoire et vous permet de comprendre comment sont rédigées les requêtes sur *Copilot Designer*.

Quelques conseils

1. Par défaut, des générations photo-réalistes

En rédigeant un prompt mentionnant uniquement les éléments constitutifs de votre image, *Copilot Designer* va, par défaut, générer une image dans un style photo-réaliste. Par exemple, si vous entrez le prompt **« `Un train rouge futuriste dans un paysage de montagnes sous la neige` »**, voici les résultats que vous pourrez obtenir (cela varie d'une fois à l'autre) :

2. L'importance de préciser le style graphique

Comme pour toute IA génératrice d'images, il est important de préciser le style graphique que vous souhaitez obtenir. Pour avoir un autre rendu, il va falloir le préciser.

Voici un premier exemple avec la requête suivante : « **`Un train rouge futuriste dans un paysage de montagnes sous la neige, dans le style impressionniste`** ».

Les amoureux des trains suisses reconnaîtront une locomotive Re 4/4 en haut à gauche…

Voici quelques exemples de styles que vous pouvez utiliser :

- Pixel art
- Croquis ou sketch
- Aquarelle
- Origami
- Cartoon ou bande dessinée
- Réaliste ou photoréaliste
- Peinture à l'huile
- Rétro ou vintage
- Minimaliste
- Abstrait
- Pop art
- Art nouveau
- Surréaliste
- Impressionniste
- Modèle ou maquette
- Cubisme
- Comics
- Mangas
- Art gothique
- Pointillisme
- Street art
- Art numérique
- Art fantastique
- Science-fiction
- Steampunk

Pour appliquer un style, il suffit de l'inclure dans votre description lors de la génération d'image. Par exemple : « **`Crée une image d'un chat dans le style pixel art`** » ou « **`Génère un paysage urbain dans un style aquarelle`** ».

Il est aussi possible de demander d'adopter le style propre à un artiste, à condition que celui ou celle-ci fasse preuve d'une certaine renommée. Nous avons par exemple demandé à l'IA de s'inspirer des travaux de Picasso. Voici notre invite : « **`Un train rouge futuriste dans un paysage de montagnes sous la neige, façon Picasso`** ». Et voici les résultats :

3. Précisions et exclusions

Il ne faut pas hésiter à préciser certains détails que l'on veut voir, ou non, apparaître. Mais ne surchargez pas trop votre requête, l'outil à tendance à rapidement omettre des consignes… Voici notre invite : « **`Un train rouge futuriste dans un paysage de montagnes sous la neige, avec la neige qui tombe`** ».

Et voici les résultats :

4. Soignez la composition

Expliquez à DALL·E comment vous souhaitez disposer les éléments sur l'image. Cela vous permettra d'obtenir un rendu plus conforme à vos désirs. Voici quelques aspects à ne pas négliger lorsque vous rédigez votre requête :

- Position des éléments sur l'image

- Taille relative des éléments
- Perspective et points de vue (plongée, contre-plongée, plan rapproché, etc)
- Éclairage et heure de la journée
- Couleurs (dominantes, nuances)
- Textures et détails
- Mouvements (un chat sautant par-dessus une clôture)

Voici un exemple de prompt qui insiste sur la composition : **« Art numérique. Vue frontale. Deux robots humanoïdes échangent un baiser sur une plage face à la mer. La scène se déroule au coucher du soleil, donnant une lumière rougeoyante. Des mouettes volent dans le ciel ».**

Source : [8]

12.5. Demander à *ChatGPT* de jouer le rôle d'un expert

On a vu dans la § 12.2 qu'il était efficace de demander à *ChatGPT* d'assumer une personnalité ou mieux encore un rôle d'expert dans un domaine donné. Une fois que ce rôle d'expert a été posé, nous voulons maintenant :

- amener *ChatGPT* à nous poser des questions afin de mieux cerner quel est notre besoin exact ;
- nous donner sa réponse que lorsqu'il a ainsi accumulé suffisamment d'informations.

Nous allons demander à *ChatGPT* de jouer le rôle d'expert en création de prompt pour une IA générative telle que *Dall·E*. C'est la technique du métaprompt.

Le prompt magique...

Voici le dialogue que nous avons eu avec *ChatGPT* pour définir en premier lieu le rôle attendu. Vous pouvez bien évidemment modifier ces requêtes à votre gré.

Copiez-collez ce prompt dans *ChatGPT*.

Moi : **« Je vais te confier une mission. Je veux que tu agisses comme un expert en création de prompt pour ChatGPT. Je vais te donner mes instructions plus bas. Est-ce que la mission est claire ? »**

ChatGPT : **« Oui, la mission est claire. Je vais agir comme un expert en création de prompts pour ChatGPT. Quelles sont tes instructions ? »**

Copiez-collez ce prompt dans *ChatGPT*.

Moi : **« En premier lieu, tu vas me demander quel est le thème du prompt. Tu peux me poser des questions afin de comprendre au mieux ce que je recherche. En deuxième lieu, tu vas me poser toutes les questions qui te semblent nécessaires pour améliorer le prompt. Lorsque tu estimes que tu as suffisamment d'information, tu vas me proposer un prompt. Si je suis satisfait, je te le dis. Sinon, nous réitérons les étapes 2 et 3 jusqu'à ce que je sois satisfait.**

OK ? »

ChatGPT : « OK, c'est compris. Commençons alors. Quel est le thème du prompt que tu souhaites créer ? »

Ici, nous donnons sa mission à *ChatGPT* : écrire un article sur un sujet donné, un scénario pour un film, etc. Nous allons lui demander un prompt pour une image.

Copiez-collez ce prompt dans *ChatGPT*.

Moi : « **Je souhaite obtenir un prompt pour Dall-E 3. L'objectif est d'obtenir une image à la fois originale et très belle.** »

À partir d'ici, *ChatGPT* va entamer un dialogue pour connaître nos besoins. Reprenons notre exemple du train rouge dans la montagne.

ChatGPT : « D'accord, merci pour cette information. Pour mieux comprendre ce que tu recherches, voici quelques questions :
1. Style artistique : Y a-t-il un style artistique spécifique que tu préfères ? (par exemple, réaliste, impressionniste, surréaliste, etc.)
2. Sujet principal : Quel est le sujet principal de l'image ? (par exemple, paysage, portrait, scène urbaine, scène de la nature, etc.)
3. Éléments spécifiques : Y a-t-il des éléments ou des détails spécifiques que tu souhaites inclure dans l'image ? (par exemple, animaux, objets, couleurs spécifiques, etc.)
4. Ambiance : Quelle ambiance ou atmosphère souhaites-tu pour l'image ? (par exemple, joyeuse, mystérieuse, paisible, dramatique, etc.)
5. Inspiration ou références : As-tu des inspirations ou des images de référence qui pourraient aider à guider la création de l'image ?
Une fois que j'aurai ces informations, je pourrai affiner le prompt pour qu'il corresponde au mieux à tes attentes. »

Moi : « **Un train rouge futuriste roule dans un paysage de montagnes, dans le style photoréaliste. Il neige. L'ambiance est froide. Il n'y a pas d'habitation, juste la nature sauvage et le train.** »

ChatGPT : « Merci pour ces précisions. Voici quelques questions supplémentaires pour affiner davantage le prompt :
1. Détails du train : Souhaites-tu des détails spécifiques sur le train rouge futuriste (par exemple, forme, design, technologie visible) ?
2. Montagnes : As-tu une préférence pour le type de montagnes (par exemple, rocheuses, avec des forêts, enneigées) ?
3. Éléments naturels : Y a-t-il d'autres éléments naturels que tu souhaites inclure (par exemple, rivières, arbres spécifiques, animaux sauvages) ?
4. Ciel et météo : Comment veux-tu que le ciel soit représenté (par exemple, nuageux, clair, coucher de soleil) ? Souhaites-tu que la neige tombe visiblement dans l'image ?
5. Perspective : As-tu une préférence pour la perspective ou l'angle de vue (par exemple, vue aérienne, vue latérale, vue de face du train) ?
Ces détails m'aideront à créer un prompt plus précis et correspondant à tes attentes. »

Moi : « **Je veux que le train ressemble à un TGV. Pour les montagnes, les Alpes. Je veux voir la neige tomber. J'aimerais voir le train arriver vers moi.** »

ChatGPT : « Merci pour ces précisions supplémentaires. Voici le

```
prompt que je propose pour obtenir une image à la fois originale et
très belle, en suivant tes instructions :

---

"Un train rouge futuriste ressemblant à un TGV roule à travers un
paysage de montagnes alpines. Il neige et les flocons tombent
visiblement. L'image est dans un style photoréaliste, avec une
ambiance froide et aucune habitation en vue, seulement la nature
sauvage. Le train arrive vers le spectateur, capturant un moment
dynamique dans ce cadre naturel."

---

Es-tu satisfait de ce prompt ou y a-t-il des ajustements que tu
souhaiterais ? »
```

On voit que malgré nos instructions, *Dall-E* ne peut pas s'empêcher de rajouter des chalets...

En résumé

Vous avez compris quelle est l'approche :

1. On demande à *ChatGPT* de jouer le rôle d'expert en création de prompt.
2. On lui demande de nous poser des questions jusqu'à ce qu'il estime qu'il détient suffisamment d'informations.
3. Il nous soumet alors un prompt détaillé issu de ce dialogue.
4. Il ne nous reste plus qu'à lui soumettre ce prompt (à lui ou à un générateur d'images).

En appliquant une telle méthode, vous allez voir vos requêtes devenir bien plus approfondies, détaillées et efficaces, et pas seulement pour la génération d'images !

À noter que cette méthode ne fonctionne pas avec toutes les IA génératives. C'est une des forces de *ChatGPT* de pouvoir poser des questions à l'utilisateur, mais toutes les IA n'en sont pas capables.

Source : [9]

12.6. *Perplexity AI* : votre moteur de recherche est une IA

Perplexity AI s'est rapidement imposé comme un concurrent sérieux dans le domaine des moteurs de recherche alimentés par l'IA. Lancé en 2022, il a séduit plus de 10 millions d'utilisateurs mensuels actifs en à peine deux ans. Sa force ? Une approche novatrice qui combine la puissance des modèles de langage avancés avec une indexation en temps réel du web. Au-delà de la recherche web, Perplexity est aussi très compétent en tant que chatbot plus classique grâce à un mode dédié. Sur la curation des modèles, le service va droit au but avec le meilleur des modèles *Claude*, *GPT*, *Llama* ainsi qu'un modèle *Sonar* maison optimisé pour la navigation web.

L'un des atouts majeurs de Perplexity réside dans sa transparence. Contrairement à d'autres chatbots, il cite systématiquement ses sources, permettant aux utilisateurs de vérifier la fiabilité des informations fournies. Cependant, son jeune âge implique qu'il n'a pas encore la même profondeur de connaissances que certains concurrents plus établis. Malgré cela, sa croissance fulgurante et sa valorisation dépassant les 500 millions de dollars témoignent de son potentiel disruptif dans l'industrie de la recherche en ligne.

Le moteur de recherche repose sur les algorithmes de *ChatGPT 3.5* ainsi que ceux développés en

interne. Toutefois, une version premium du service permet d'avoir accès aux modèles *GPT-4*, *Claude* 3, *Mistral Large*, *Meta Llama* 3 ainsi que le modèle *Experimental* de *Perplexity*. Et cette diversité, c'est justement la force du service.

Si vous avez déjà un peu joué avec *ChatGPT* pour générer quelques textes ou obtenir la réponse à une question, vous pouvez sans doute vous demander de prime abord en quoi Perplexity se distingue finalement du service d'*Open AI*. Mais puisqu'il s'agit d'un véritable moteur de recherche, il pourrait aussi dans certains cas supplanter *Google*.

La fraîcheur et la clarté des résultats

Dans sa version gratuite, *ChatGPT* se base uniquement sur les données avec lesquelles il a été initialement entraîné. Celles-ci ne sont pas rafraîchies avec des informations mises à jour. De son côté, *Perplexity* utilise non seulement ces données mais peut également analyser le Web pour les affiner sans obtenir des résultats obsolètes et surtout, en spécifiant les sources.

Pour illustrer cette différence, prenons comme exemple la simple question : « quels sont les meilleurs smartphones pour la photo ». À cette dernière, *ChatGPT* est resté à l'iPhone 14 Pro, le Google Pixel 7 Pro et le Samsung Galaxy S23 Ultra. De son côté, *Perplexity* est capable d'obtenir des sources bien plus fraîches et nous propose l'iPhone 15 Pro, le Google Pixel 8 ou le Samsung Galaxy S24 Ultra.

Par rapport à *Google*, *Perplexity* ne propose pas de publicité ni ne met en avant ses partenaires. Son modèle économique par abonnement est clair, et disons-le, sain. Exit les résultats sponsorisés ou les encarts Shopping. En lieu et place d'une liste de liens dont l'agencement est régulé par *Google* et ses mises à jour d'algorithmes aléatoires, *Perplexity* souhaite véritablement donner une réponse efficace et *sourcée*.

Des sources mises en évidence et maîtrisées

La très grande médiatisation de *ChatGPT* a permis d'illustrer les travers des algorithmes avec parfois des sources plus que douteuses. C'est aussi ce qui a récemment terni l'image de *Gemini* invitant les utilisateurs à mettre de la colle sur une pizza pour éviter que le fromage ne coule partout...

Sur *Perplexity*, l'idée est de mettre en évidence les sources d'informations, lesquelles sont positionnées au-dessus des résultats de recherche. Cela apporte évidemment un côté très rassurant pour l'internaute, il est alors possible d'aller plus loin que le résumé généré et de lire l'article en détails.

Autre point intéressant, nous sommes en mesure de gérer plus finement les sources d'informations. D'un côté, il est possible de retirer une source pour ne laisser que celles que nous jugeons qualitatives, d'un autre, nous pouvons ajouter nous-mêmes des informations en chargeant des images, du texte ou des fichiers PDF. Dans sa version gratuite, *Perplexity* autorise trois fichiers, lesquels seront exploités pour répondre à vos requêtes ultérieures.

Quid du respect de la vie privée ?

Bien évidemment, dès qu'on parle de chargement de fichiers... on ne peut s'empêcher de penser à la notion de vie privée. Nous ne reviendrons pas en détails sur les travers de *Google* en la matière, ni sur son modèle économique vieillissant lié à la publicité ciblée.

Pour sa part, *Perplexity* explique collecter les informations relatives à votre appareil ainsi que vos interactions avec le service. Cela signifie que les requêtes précédentes et les résultats retournés sont également exploités. L'entreprise ajoute « nous utilisons ces données pour améliorer l'expérience de recherche pour nos utilisateurs. Si vous souhaitez vous désinscrire, allez dans votre profil, accédez au paramètre "Utilisation des données IA" sous "Paramètres", et désactivez-le pour empêcher Perplexity d'utiliser vos données de recherche pour améliorer nos modèles d'IA. »

En ce qui concerne le chargement de vos fichiers, *Perplexity* assure que ces derniers ne sont exploités que pour effectuer une tâche (comme une traduction ou un résumer) ou pour ne répondre qu'aux questions de cet utilisateur. Ces documents sont conservés pour une période de 30 jours. Une fois le fichier effacé, il sera toutefois possible de rebondir sur des résultats de recherche précédents basés sur ce fichier. En revanche, si vous supprimez cette conversation, il sera nécessaire de recharger le fichier en question pour alimenter de nouveau le moteur avec ces informations.

Source : [13] (extraits)

12.7. Les générateurs de code

« L'IA va écrire 80 % du code d'ici 5 ans » affirmait en 2022 de manière prophétique et un brin provocatrice Thomas **Dohmke**, le patron de *GitHub*, la plateforme utilisée par plus de 100 millions de développeurs.

Les IA génératives comme *ChatGPT*, *Copilot* ou *Gemini* ont marqué la démocratisation de l'intelligence artificielle. Basées sur de grands modèles de langage (*LLM*), ces IA ne servent pas qu'à créer du texte ou des images. Elles peuvent aussi générer du code, le commenter, suggérer des optimisations ou faire la chasse aux bugs, faisant gagner un temps précieux aux développeurs.

Lancé en 2021, *GitHub Copilot*, l'IA de génération de code la plus populaire, compte déjà en octobre 2023 plus d'un million d'utilisateurs payants dans plus de 37 000 organisations !

Il n'en faut pas plus pour consacrer l'année 2024 comme celle des IA spécialisées en code. De quoi raviver le débat engendré par la tendance low-code/no-code sur la disparition du métier de développeur, ou du moins son inéluctable évolution.

Le code, une donnée idéale pour alimenter l'IA

Le code fait partie des domaines de prédilection de l'IA. Et ce, bien avant l'arrivée de *ChatGPT* en novembre 2022. Citons l'assistant *IBM Watson Code* en 2015, *DeepCode* en 2016 pour trouver les failles de sécurité d'un programme ou *Microsoft IntelliCode* en 2017 pour améliorer la saisie semi-automatique de code d'*IntelliSense* apparue dix ans plus tôt dans *Visual Studio*.

Rien d'étonnant à cela, ces IA sont basées sur de grands modèles de langage (LLM) entraînés sur de grands ensembles de données à base de texte. Et pour alimenter les algorithmes d'apprentissage automatique, ces IA ont besoin de données structurées. Le code fait alors figure de candidat idéal : sous format texte, structuré, disponible en grande quantité, en open source, dans d'immenses dépôts comme *GitHub* ou *SourceForge*.

Ainsi, *GitHub Copilot* a été entraîné sur les 54 millions de dépôts de *GitHub*, notamment un jeu de données de 159 Go de code Python.

À quoi servent les IA de code ?

Utiliser l'intelligence artificielle est donc devenu une pratique courante de nombreux développeurs. On les retrouve sous la forme d'extensions dans les environnements de développement (IDE) les plus courants, comme *Visual Studio Code*, *Eclipse* ou les logiciels spécialisés de *JetBrains* (*IntelliJ*, *PhpStorm*, *PyCharm*, *WebStorm*…).

À ce jour, on compte plusieurs centaines d'outils d'aide au développement, couvrant une large variété d'applications. Leur nombre augmente sans cesse.

Les usages des IA pour le développement sont très nombreux. Voici quelques exemples.

Génération automatique de code

C'est de loin la fonction la plus utilisée par les développeurs. L'IA crée du code à partir d'une requête (un prompt), une description en langage naturel de ce que l'on souhaite, ou à partir du code déjà saisi en le complétant automatiquement. Cette dernière fonction d'autocomplétion est une amélioration des systèmes de saisie semi-automatique qui existent depuis longtemps dans les IDE. Mais au lieu de deviner seulement le prochain mot, l'IA va générer une ou plusieurs lignes de code selon des statistiques de probabilité.

L'IA ne fait que des suggestions. Le développeur peut affiner sa demande de manière itérative afin d'obtenir le résultat désiré.

Le gain de temps est énorme. Les développeurs peuvent coder plus vite (55 % plus vite selon les statistiques de *GitHub*), avec plus de précision et avec une meilleure compréhension du code.

Détection et correction des bugs

Une IA de code sert également à débugger un code, à le valider par rapport à la syntaxe d'un langage donné, mais aussi détecter des erreurs difficiles à trouver dans un code complexe. Ainsi, l'IA analyse le code qu'on lui soumet et détecte les éventuelles erreurs, comme :

- les erreurs de syntaxe : un point-virgule qui manque à la fin d'une instruction en Java ou en

JavaScript, une balise de fermeture manquante en PHP, une accolade en trop dans une CSS…
- les erreurs d'exécution : utilisation d'une variable non initialisée ou supprimée (NullPointer), tentative d'accès à un objet qui n'est plus accessible en mémoire (PointerException), pile d'appels trop grande (StackOverflow)…
- les erreurs de logique : division par zéro, débordement arithmétique, appel de fonction inexistante…
- les erreurs d'entrées-sorties : erreurs de lecture ou d'écriture sur un support de stockage, tentative d'ouverture d'un fichier qui n'existe pas…
- les erreurs de formatage : format inattendu par une chaîne de caractères, type de donnée incorrect…

Analyse et explication de code

Pour mieux comprendre un code, un développeur peut questionner une IA en langage naturel à travers une interface de chat. Certaines IA sont également capables de générer des visualisations du code afin d'aider les développeurs à mieux appréhender la structure du code, sa logique et les relations entre les différentes parties le composant.

L'utilisation massive de l'IA pour expliquer du code a entraîné la chute de l'audience de *Stack Overflow*, le célèbre forum de questions-réponses consulté par des millions de développeurs, provoquant la suppression de 28 % de son personnel en octobre 2023.

Documentation du code

Terminer le code nécessite de rédiger sa documentation, une étape fastidieuse et souvent négligée. Trop souvent, les développeurs sont poussés à fournir un code fonctionnel de plus en plus rapidement, laissant la documentation au second plan.

L'intelligence artificielle peut changer la donne. En analysant le code, elle peut produire immédiatement une documentation complète et mise en forme, garantissant sa clarté et sa lisibilité. Ce processus révolutionnaire réduit considérablement la dette technique et permet aux développeurs de se concentrer sur leur cœur de métier : la création de code.

Conversion d'un langage à un autre

Maîtrisant plusieurs langages de programmation, la plupart des IA de code sont capables de traduire du code d'un langage à un autre. L'opération n'est pas aisée, car passer d'un langage à un autre nécessite de convertir la syntaxe (changement de structure par blocs vers indentation, par exemple), la logique (changement des paradigmes et des structures de contrôle), trouver des bibliothèques équivalentes dans le langage cible, convertir les commentaires et la documentation, et optimiser le code pour le langage cible.

On pourra ainsi passer du C++ vers Python et vice versa ou du Java vers Python en quelques secondes. La conversion automatique de code par IA accélère le processus de développement logiciel sur différentes plateformes et évite les erreurs de traduction manuelles. Outre les IA de code ci-dessous, citons des IA spécialisées comme *DeepCode*, *Polyglot*, *Code Trans* et *AI Code Translator*.

Quelques générateurs de code

Ne pas confondre avec *Microsoft Copilot*...

- *GitHub Copilot :* https://github.com/features/copilot
- *Codeium :* https://codeium.com/
- *Amazon Q Developer :* https://aws.amazon.com/fr/q/developer/
- *Replit AI :* https://replit.com/ai
- …

12.8. Coder avec *ChatGPT*

ChatGPT ne parle pas seulement des langues humaines. Il parle auss en langage informatique.

Le métier de développeur est en pleine mutation. Utiliser au quotidien les IA dans son travail peut vous faire gagner un temps très précieux. Voyons comment.

12.8.1. Faire écrire à ChatGPT un code d'exemple

Imaginez, par exemple, que vous devez écrire un programme qui trie une liste de nombres. Vous avez un problème concret, mais vous ne savez pas comment vous y prendre.

Vous pouvez demander à *ChatGPT* de vous aider avec un prompt simple comme "**`Ecris un programme Python qui trie une liste de nombres`**" :

Voici un exemple de programme Python qui trie une liste de nombres en utilisant la méthode de tri intégrée `sort()` :

python — Copier le code

```python
# Fonction pour trier une liste de nombres
def trier_liste(liste):
    liste.sort()  # Trie la liste dans l'ordre croissant
    return liste

# Exemple d'utilisation
nombres = [12, 5, 9, 3, 15, 8, 2]
print("Liste avant le tri :", nombres)

liste_triee = trier_liste(nombres)
print("Liste après le tri :", liste_triee)
```

ChatGPT ne fournit pas seulement un code Python, il explique en plus comment il fonctionne :

Explication :

- La fonction `trier_liste` prend une liste de nombres en entrée et utilise la méthode `sort()` pour trier la liste en place (c'est-à-dire qu'elle modifie directement la liste d'origine).
- L'exemple montre comment appeler cette fonction avec une liste donnée et afficher la liste avant et après le tri.

Si tu veux trier dans l'ordre décroissant, tu peux ajouter un argument à `sort()` comme ceci :

python — Copier le code

```python
liste.sort(reverse=True)
```

Cela triera la liste en ordre décroissant.

Auparavant, on aurait fait une recherche *Google* et on serait tombé sur une page du site *StackOverflow* pour obtenir la même réponse.

Cette interaction simple montre comment *ChatGPT* peut aider à produire du code qui résout un problème spécifique. De cette manière, vous pouvez vous concentrer sur des tâches de plus haut niveau et laisser *ChatGPT* s'occuper du reste.

Attention ! Les codes fournis par *ChatGPT* ne sont pas toujours corrects. Parfois, il se trompe. Cela dit, il fonctionne globalement très bien.

Si cela arrive, il est possible de signaler précisément le problème à *ChatGPT* pour qu'il corrige l'erreur. Parfois il y parviendra, parfois pas…

Si le résultat est faux, demandez à *ChatGPT* de corriger son code !

Exercice 12.7

Écrire un programme Python qui place dans une grille carrée de 3 lignes et 3 colonnes les nombres 1, 2, 3, 4, 6, 9, 12, 18, 36 de sorte que le produit de chaque ligne, chaque colonne et chaque diagonale fasse 216. Demandez à *ChatGPT* de faire le boulot !

12.8.2. Demander à *ChatGPT* des explications

Ecris un programme Python qui donne toutes les façons d'obtenir 216 en multipliant 3 nombres entiers choisis parmi 1, 2, 3, 4, 6, 9, 12, 18, 36. On ne peut utiliser chaque nombre qu'une seule fois.

Voici un programme Python qui trouve toutes les façons d'obtenir 216 en multipliant 3 nombres entiers choisis parmi la liste donnée `[1, 2, 3, 4, 6, 9, 12, 18, 36]`, en s'assurant que chaque nombre est utilisé une seule fois dans chaque combinaison :

python Copier le code

```python
import itertools

# Liste des nombres disponibles
nombres = [1, 2, 3, 4, 6, 9, 12, 18, 36]
produit_cible = 216

# Trouver toutes les combinaisons de trois nombres
combinations = list(itertools.combinations(nombres, 3))

# Filtrer les combinaisons dont le produit est égal à 216
resultats = [comb for comb in combinations if comb[0] * comb[1] * comb[2] == produit_cible]

# Afficher les résultats
print("Les combinaisons de trois nombres qui donnent 216 sont :")
for res in resultats:
    print(res)
```

On peut maintenant entamer un dialogue avec *ChatGPT* pour qu'il explique certaines parties du code.

Je ne comprends pas la ligne:
resultats = [comb for comb in combinations if comb[0] * comb[1] * comb[2] == produit_cible]

Pas de souci ! Je vais t'expliquer cette ligne en détail.

La ligne en question :

python Copier le code

```python
resultats = [comb for comb in combinations if comb[0] * comb[1] * comb[2] == produit_cible
```

C'est ce qu'on appelle une **compréhension de liste** en Python, un moyen compact de créer une nouvelle liste en appliquant une condition à chaque élément d'une autre liste. Décomposons-la.

Suit une longue explication détaillée que l'on n'a pas reproduite ici…

12.8.3. Faire corriger ses erreurs de code par *ChatGPT*

Parfois, notre code ne se comporte pas comme prévu. Dans ces moments-là, *ChatGPT* peut être un outil précieux pour déceler et corriger les erreurs.

Supposons que vous ayez un code qui est censé imprimer les nombres de 1 à 10, mais qu'il imprime seulement jusqu'à 9.

Corrige ce code Python. Il est censé imprimer les nombres de 1 à 10, mais il imprime seulement jusqu'à 9.

```
for i in range(1, 10):
    print(i)
```

Le problème dans ton code vient du fait que la fonction `range(1, 10)` génère des nombres allant de 1 à 9, mais sans inclure 10. En Python, la fonction `range` exclut toujours la borne supérieure.

Pour corriger cela et imprimer les nombres de 1 à 10, il suffit de changer la borne supérieure à 11 :

python — Copier le code

```python
for i in range(1, 11):  # 11 est exclu, donc la boucle s'arrête à 10
    print(i)
```

Explication :

- `range(1, 11)` génère les nombres de 1 à 10, car le deuxième argument (11) est exclu.

Exercice 12.8

Demandez à *ChatGPT* de corriger ce programme Python :

```
a = 1.4142
r = sqrt(2)
print('racine de 2 =', r)
if r = a :
print('a = r')
```

Exercice 12.9

Demandez à *ChatGPT* de corriger ce programme Python :

```
réponse = 0
limite = 5*9**5+1
for n in range(10,limite)
    s = 0
    l = list(str(n))
    for i in l:
        s += int(i)**5
    if n = s:
        reponse += n
        print(n)
        print()

print(reponse)
```

12.8.4. Demander à ChatGPT de nous expliquer un message d'erreur

Les messages d'erreur peuvent parfois être déroutants, surtout pour les débutants. *ChatGPT* peut nous aider à comprendre ces messages et même corriger notre code sur la base de l'erreur.

Que signifie cette erreur Python "TypeError: can only concatenate str (not "int") to str" et comment la corriger?

L'erreur "**TypeError: can only concatenate str (not 'int') to str**" survient lorsque tu essaies de **concaténer (ou joindre)** une chaîne de caractères (`str`) avec un entier (`int`) sans d'abord convertir l'entier en chaîne.

Exemple de code causant l'erreur :

python Copier le code

```python
age = 25
print("J'ai " + age + " ans.")
```

Dans cet exemple, Python ne peut pas concaténer la chaîne `"J'ai "` et l'entier `25` directement, car la concaténation ne peut se faire qu'entre des chaînes de caractères.

Comment corriger cela ?

Tu dois convertir l'entier en chaîne de caractères avant de le concaténer avec d'autres chaînes. Cela se fait avec la fonction `str()` :

Correction avec `str()` :

python Copier le code

```python
age = 25
print("J'ai " + str(age) + " ans.")
```

Ici, `str(age)` convertit l'entier `age` en une chaîne de caractères avant la concaténation, ce qui résout l'erreur.

Source : [14]

12.8.5. Les limites de *ChatGPT*

Bien que *ChatGPT* présente de nombreuses capacités avancées, il existe plusieurs limitations importantes à considérer.

Manque de créativité et d'analyse complexe

Malgré ses capacités impressionnantes, *ChatGPT* présente des limitations notables en termes de créativité et d'analyse complexe. Bien qu'il puisse générer du texte cohérent et bien structuré, *ChatGPT* manque de la capacité à effectuer des analyses critiques ou à proposer des idées originales.

Par exemple, *ChatGPT* est efficace pour automatiser des tâches répétitives et fournir des explications basées sur des données existantes, mais il est incapable de remplacer la pensée créative et la profondeur d'analyse qu'un développeur humain apporte à la résolution de problèmes complexes et à l'innovation technologique.

Des problèmes de sécurité identifiés et signalés par OpenAI

L'utilisation de *ChatGPT* pour générer du code informatique comporte des risques de sécurité importants. Le code produit par l'IA peut contenir des bugs ou des vulnérabilités difficiles à détecter immédiatement. Par exemple, un script généré pourrait manquer de vérifications de sécurité essentielles, rendant une application vulnérable à des attaques telles que l'injection SQL ou le cross-site scripting.

Ainsi, bien que *ChatGPT* puisse assister les développeurs en automatisant certaines tâches, une révision humaine minutieuse est recommandée par *OpenAI* pour garantir la sécurité et la fiabilité du code généré.

12.8.6. L'impact de *ChatGPT* sur le métier de développeur

ChatGPT a un impact significatif sur le métier de développeur, transformant les méthodes de travail et les compétences requises.

L'automatisation des tâches répétitives : un gain de productivité

L'une des principales contributions de *ChatGPT* au métier de développeur est l'automatisation des tâches répétitives. Grâce à ses capacités avancées, *ChatGPT* peut prendre en charge des tâches chronophages telles que la génération de scripts de test, la configuration d'environnements de développement, et la rédaction de documentation technique.

En déléguant ces tâches à l'IA, les développeurs peuvent se concentrer sur des aspects plus stratégiques et créatifs de leur travail, comme la conception architecturale et l'innovation logicielle.

Une évolution des compétences nécessaires : comprendre la GenAI

Avec l'intégration croissante de *ChatGPT* dans les flux de travail de développement, les compétences nécessaires pour les développeurs évoluent. Il devient essentiel pour les professionnels de la tech de se familiariser avec les outils d'IA et de comprendre comment les utiliser efficacement.

La mise à jour continue des compétences est cruciale, notamment l'apprentissage des nouvelles technologies et des meilleures pratiques pour collaborer avec l'IA. Les développeurs doivent également développer des compétences en supervision et en évaluation critique du travail généré par l'IA pour garantir la qualité et la sécurité du code produit.

Conclusion : pourquoi ChatGPT ne remplacera pas les développeurs

Bien que *ChatGPT* et les autres technologies d'IA puissent transformer le développement logiciel en automatisant des tâches répétitives et en fournissant une assistance précieuse, elles ne remplaceront pas les développeurs. L'intervention humaine reste cruciale pour l'innovation, la résolution de problèmes complexes et la supervision des systèmes d'IA eux-mêmes. Les développeurs continuent de jouer un rôle central dans la création, la maintenance et l'amélioration des technologies en assurant que ces outils restent sûrs, fiables et adaptés aux besoins spécifiques des projets.

Exercice 12.10

D'autres IA génératives peuvent aussi générer ou corriger du code !

Refaites les exercices 7 à 9 avec *Claude* et *Gemini* (et d'autres si vous le souhaitez…).

Sources

[1] Marr, Bernard, « IA générative : quelle IA se distingue dans la course effrénée ? », <https://www.forbes.fr/technologie/ia-generative-quelle-ia-se-distingue-dans-la-course-effrenee/>, Forbes, 2024

[2] Bernard Hugo, « C'est quoi un LLM ? Comment fonctionnent les moteurs de ChatGPT, Google Bard et autres ? », <https://www.frandroid.com/culture-tech/intelligence-artificielle/1852573_cest-quoi-un-llm-comment-fonctionnent-les-moteurs-de-chatgpt-google-bard-et-autres>, Frandroid, 2024

[3] Cloudfare, « Qu'est-ce qu'un grand modèle de langage (LLM) ? », <https://www.cloudflare.com/fr-fr/learning/ai/what-is-large-language-model/>, 2024

[4] Salenne Ludo, « Rédiger un Prompt ChatGPT : Le Guide Ultime pour un Prompt Parfait », <https://www.ludosln.net/rediger-prompt-chatgpt-parfait>, 2023

[5] Eugène Matthieu, « ChatGPT : notre guide pour créer les meilleurs prompts »,

<https://www.blogdumoderateur.com/chatgpt-guide-creer-meilleurs-prompts>, 2023

[6] PrintOBlog, « Les 15 meilleures alternatives à Midjourney », <https://www.printoclock.com/blog/midjourney-alternative/>, 28 juin 2024

[7] Eugène Matthieu, « Comment utiliser Bing Image Creator : le guide pour générer des images par IA », <https://www.blogdumoderateur.com/comment-utiliser-bing-image-creator-guide/>, 29 juin 2023

[8] Ichbiah Daniel, « ChatGPT : découvrez l'astuce pour créer le prompt ultime », <https://www.futura-sciences.com/tech/questions-reponses/intelligence-artificielle-chatgpt-decouvrez-astuce-creer-prompt-ultime-19389/>, 27 janvier 2024

[9] Belfiore Guillaume, « Perplexity.ai : que vaut le moteur de recherche face à Google et ChatGPT ? », <https://www.clubic.com/avis-529920-perplexity-ai-que-vaut-le-moteur-de-recherche-face-a-google-et-chatgpt.html>, 17 juin 2024

[10] Laurent le Feuvre, *L'apprentissage et l'enseignement à l'ère des IA génératives*, Leading change et leadership, 2025

[11] Mickaël Bertrand, *J'enseigne avec l'IA ! Guide pratique de l'IA au service de l'enseignant et de l'élève*, Vuibert, 2025

[12] https://bonprompt.com/cest-quoi-un-bon-prompt/

[13] « IA : les générateurs de code, une révolution pour les développeurs », ORSYS le mag, <https://orsys-lemag.com/ia-les-generateurs-de-code-une-revolution-pour-les-developpeurs/>, 7 février 2024

[14] Durand Amandine, « Est-ce que ChatGPT va remplacer les développeurs ? », Wild Code School, <https://www.wildcodeschool.com/fr-fr/blog/est-ce-que-chatgpt-va-remplacer-les-développeurs>, 22 juin 2024